Application of the Multiblock Method in Computational Aerodynamics

Brian J. Gribben, M.Eng

Thesis submitted to the Faculty of Engineering,
University of Glasgow, for the Degree of Doctor of Philosophy

University of Glasgow
Department of Aerospace Engineering

November 1998

Abstract

The main challenge in computational aerodynamics is to provide practical, credible, cost and schedule effective methods for routine design application and for full integration of these methods into the design cycle. Although advances in physical modelling and solution algorithms are continuing requirements of the aerospace industry, other more practical difficulties also impede the full realisation of the potential of existing methods. The contribution of this thesis is to examine and tackle several of these issues and to evaluate computational aerodynamics as a tool for engineering design and scientific enquiry.

An advanced computational aerodynamics method is evaluated as an engineering tool for axisymmetric forebody and base flow problems. First the adaption of an existing two-dimensional flow solver to axisymmetric flow is described, then specific test cases are considered. The motivation for creating an axisymmetric flow solver is the considerable performance improvements compared to a fully three-dimensional method. The accuracy and robustness of the method are very good for forebody problems. For base flow problems accuracy and robustness are less satisfactory, although the performance of other prediction methods is also poorer for this more demanding problem. For both problem types the speed of the flow solver, the required computing resource and the time and effort necessary for pre- and post-processing are all satisfactory for routine calculation in an engineering environment.

Shock reflection hysteresis and plume structure in a low density, axisymmetric highly underexpanded air jet is examined using a Navier-Stokes flow solver. This type of jet is found in a number of applications e.g. rocket exhausts and fuel injectors. The plume structure is complex, involving the interaction of several flow features, making this a demanding problem. Two types of shock reflection appear to occur in the plume, regular and Mach, depending on the jet pressure ratio. The

existence of a dual solution domain where either type may occur has been predicted, in agreement with experiment where the same phenomenon has been observed for a nitrogen jet. There is a hysteresis in the shock reflection type; the reflection type observed in the dual solution domain depends on the time history of the plume development. A quasi-steady approach is employed in order to calculate the entire hysteresis loop. The results of the computational study are used to examine the structure of the plume, and are compared with experimental data where possible. Some flow features not initially recognised from experiment have been identified, notably curvature of the Mach disc, recirculation behind the Mach disc and the 'regular' reflection having Mach reflection characteristics. Included in the study is a review of the two dimensional shock reflection hysteresis problem to establish a theoretical background. The value of CFD as a tool for scientific investigation is clearly demonstrated by this study.

The need for automation of the multiblock grid generation process is discussed. A new approach to automatically process a multiblock topology in order to prepare it for the grid generation process is described. The method is based on a cost function which attempts to model the objectives of the skilled grid generation software user who at present performs the task of block positioning and shaping in an interactive manner. A number of test cases are examined. It is also suggested that an existing unstructured mesh generation method could be adopted as an initial topology generation tool. Further work towards creating a fully automatic grid generation tool and extension into three dimensions are discussed.

The parallel execution of an aerodynamic simulation code on a non-dedicated, heterogeneous cluster of workstations is examined. This type of facility is commonly available to CFD developers and users in academia, industry and government laboratories and is attractive in terms of cost for CFD simulations. However, practical considerations appear at present to discourage widespread adoption of this technology. The main obstacles to achieving an efficient, robust parallel CFD capability in a demanding multi-user environment are investigated. A static load-balancing method is described which takes account of varying processor speeds. A dynamic re-allocation method to account for varying processor loads has been developed. Use of proprietary software has facilitated the implementation of the method.

Acknowledgements

I would like to express my sincere gratitude to my supervisors Dr. Ken Badcock and Prof. Bryan Richards. Their continual assistance and encouragement and relentess enthusiasm have been greatly appreciated. Thanks also to the other members of the CFD group, especially Franck Cantariti, Mark Woodgate, Laurent Dubuc and Bill McMillan for all their help over the past three years. Special thanks is extended to Trevor Birch at the Defence Evaluation and Research Agency, Bedford for his support, advice and practical assistance. I would also like to express my gratitude to Paul Welsh, Terry Cain and Martin Gilmore from DERA Farnborough for generously providing experimental data and fruitful discussions during the course of the study, and to Mairi Macdonald at the Queen's University of Belfast for assistance with mesh generation.

This work is supported by a University of Glasgow scholarship and sponsorship from DERA Bedford.

Contents

List of Figures

Chapter 1

Introduction

In the last forty years the discipline of computational fluid dynamics (CFD) has undergone considerable growth. CFD methods are now employed in a number of diverse fields including aircraft, ship and road vehicle design, meteorology, oceanography, haematology, astrophysics, mineral recovery, polymer manufacture and machinery lubrication. This work is primarily concerned with computational aerodynamics, an important subset of CFD in which many of the elements of CFD technology were first developed. However, much of the discussion applies to other application areas.

The primary impetus for the development of computational aerodynamics has always been the requirements of aircraft designers, who need reliable aerodynamic predictions to produce better aircraft. Before the advent of computational techniques the primary tool for aerodynamic investigation was the wind tunnel. This was complemented by an advanced theoretical understanding of fluid mechanics which aided interpretation of experimental results. Aerodynamic theory itself provided analysis tools, notably Kutta-Joukowski aerofoil theory, Prandtl's wing and boundary layer theories, Jones' slender wing theory[1] and Hayes' linearized supersonic flow theory[2]. Notwithstanding the ingenuity and continuing relevance of these methods, they all require simplifying assumptions which limit their applicability; none of the methods are suitable for complex flows with strong nonlinear effects. The advent of electronic computers made possible the use of numerical methods for calculating aerodynamic values. An early example of a numerical approach to an aerodynamic

problem is the manual calculation of the Theodorsen method for conformal mapping to develop the NACA 6 aerofoil series in the 1940's[3]. In 1947 tables for supersonic flow around cones were compiled by solving the Taylor-MacColl equation using a primitive computer[4]. Calculating machines created the potential to greatly extend the practicality of a numerical approach. By the 1960's the possibility of using high speed digital computers arrived. A major breakthrough was the development of panel methods for the solution of linearized potential flow. Despite being restricted to inviscid, incompressible, irrotational flow this approach proved very useful in calculating pressure forces for commercial airliner configurations where the flow is largely attached. The method was first developed in 1962[5] and was subsequently applied to lifting flows[6] and linearized supersonic flow[7].

The 1970's saw considerable effort devoted to nonlinear flow models, focussing on transonic flow with shock waves. A major advance was Murman and Cole's scheme for solving the transonic small-disturbance equation[8]. Full potential flow methods followed quickly[9],[10],[11],[12]. Algorithm capability for the Euler equations[13],[14],[15] was greatly enhanced by the introduction of flux-splitting[16] for better shock capturing. Subsequent algorithm development for the Euler and Reynolds-Averaged Navier-Stokes (RANS) equations has continued apace; a very wide variety of discretisations and solution procedures are now available, see for example[17],[18].

Inevitably the development of CFD is strongly influenced by advances in computer hardware. Calculations using more complex mathematical models on a progressively larger scale have been made possible by improvements in computer speed and memory size. It has been estimated that better algorithms and better hardware have contributed roughly equally to CFD progress in the last two decades[17]. In the 1950's the state of the art was represented by the solution of a linear flow model with a few hundred unknowns. Before 1965 computational methods were scarcely used in aerodynamics, but within ten years linear potential methods applied to entire aircraft configurations were well established. In 1983 an Euler solution for an entire aircraft configuration was regarded as an attainable objective[19]. By 1989 three-dimensional steady and two-dimensional unsteady RANS solutions were being obtained on supercomputers[20]. Now in 1998, three-dimensional unsteady RANS

solutions can be obtained using a parallel machine, based on commodity processing units, which is two orders of magnitude less expensive than the previous generation of supercomputers.

CFD is now sufficiently developed to be widely accepted as a key tool in aerodynamic design. Due to impressive algorithm development, especially in the past two decades, the underlying principles for the design and implementation of robust, accurate schemes are now well established[17]. CFD complements the other two approaches of pure theory and pure experiment in aircraft design. However, the aerodynamics community unanimously recognises that CFD still has far to go before fully realising its potential[17],[21],[22],[23]. The main challenge of today is to provide practical, credible, cost and schedule effective codes for routine design application and for full integration of these codes into the design cycle, i.e. to obtain a mature CFD capability. Obstacles to this aim are the well-known modelling difficulties which limit the fidelity of computational aerodynamic simulation, principally regarding turbulence but also in other areas such as finite reaction rate chemistry and nonequilibrium thermodynamics. Industry would naturally welcome any advances in modelling as well as in solution algorithms to improve the robustness and reduce the turn-around time of simulations. However to fully exploit today's CFD technology, the best way forward for engineers is to learn to live with the modelling and algorithm limitations, to anticipate and quantify them, in the same way as inherent limitations of wind tunnel testing can often be tolerated and accounted for by drawing on the large body of experience accumulated over time. The well established experimental and semi-empirical methods for aerodynamic analysis are useful tools because they are applied with sound engineering judgement. Credibility of CFD simulations is often doubted in industry; the best way to reduce the credibility gap is through gaining experience and understanding of the strengths and limitations of CFD methods to promote the application of engineering judgment. To encourage this, and to make CFD more attractive to the wider engineering community, the practical difficulties which impede use of CFD must be overcome; at present only large enterprises have the resources and expertise to purposefully exploit CFD. Effective use of computational aerodynamics in the design process is hindered by long lead times (especially for grid generation) and very high computational and human

costs. These practical difficulties will remain despite any advances in modelling or algorithms. The contribution of this thesis is to examine and tackle several of these issues and to evaluate CFD as a tool for engineering design and scientific enquiry.

The work presented here is based around an existing two-dimensional multiblock flow solver for the Euler and Navier-Stokes equations, developed by the CFD group in the Department of Aerospace Engineering at the University of Glasgow. Details of the flow solution method and implementation will be described where appropriate. However, it is useful to explain the term "multiblock" now since the concept is important to each of the following chapters. Figure 1.1 shows a structured grid for an aerofoil calculation. Note that for this case the grid can be mapped onto a rectangle in parametric space. This facilitates the implementation of a flow solution method; the grid points and flow quantities associated with grid cells are considered as elements of two-dimensional arrays. However, this feature does not extend to arbitrary geometries. Even for relatively simple configurations it becomes difficult or impossible to create a structured grid. There are two main approaches to this problem. An example of an *unstructured grid* for a two-element aerofoil problem is shown in Figure 1.2. It is possible to construct an unstructured grid for any geometry since there are no associated geometric restrictions. However, flow solution methods are less efficient due to the necessity of a more laborious data structure. A detail of a *multiblock structured* grid around the leading edge slat of an aerofoil is shown in Figure 1.3. The premise of the multiblock method is to employ a number of structured grids, or *blocks*, in order to achieve geometric flexibility. The outlines of the grid blocks are shown in Figure 1.4. The advantages of the structured grid flow solver are retained, but at the expense of considerable grid generation complexity. Thus the choice between an unstructured or multiblock approach is primarily a trade-off between flow solver and grid generation complexity. Note however that there are also other issues; for example many researchers assert that shear layer resolution on unstructured grids is unsatisfactory. The multiblock method used here and unstructured grids are the two most common approaches, although other inventive approaches also appear in the literature[24].

The main body of this work is split into two parts. Part I (Chapters 2,3,4) concerns the implementation of an axisymmetric flow solver and its application to

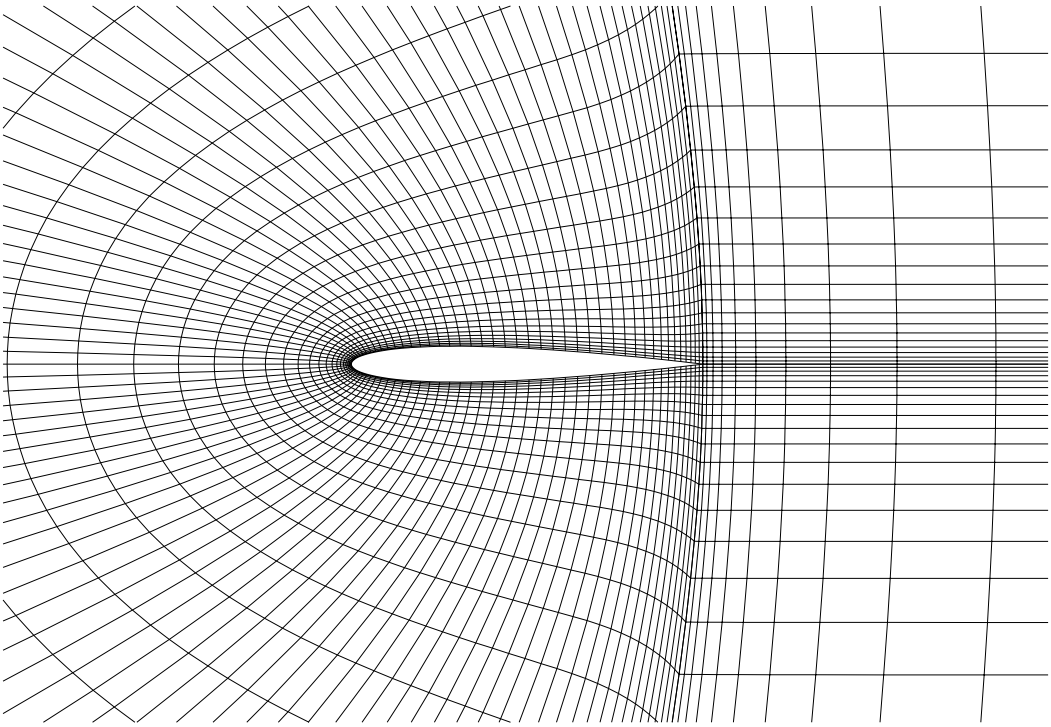


Figure 1.1: *Structured grid*

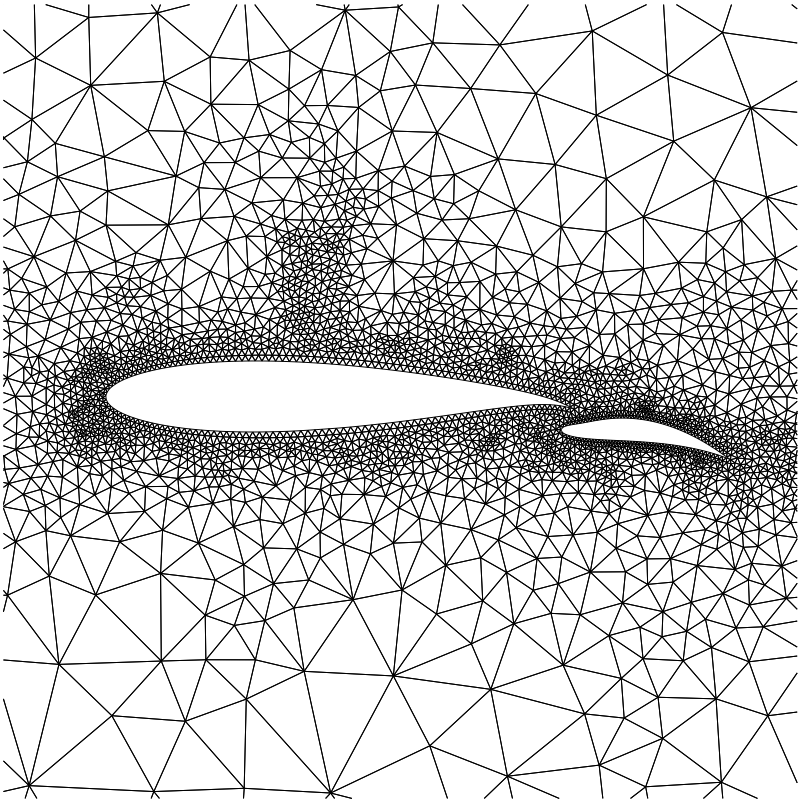


Figure 1.2: *Unstructured grid*

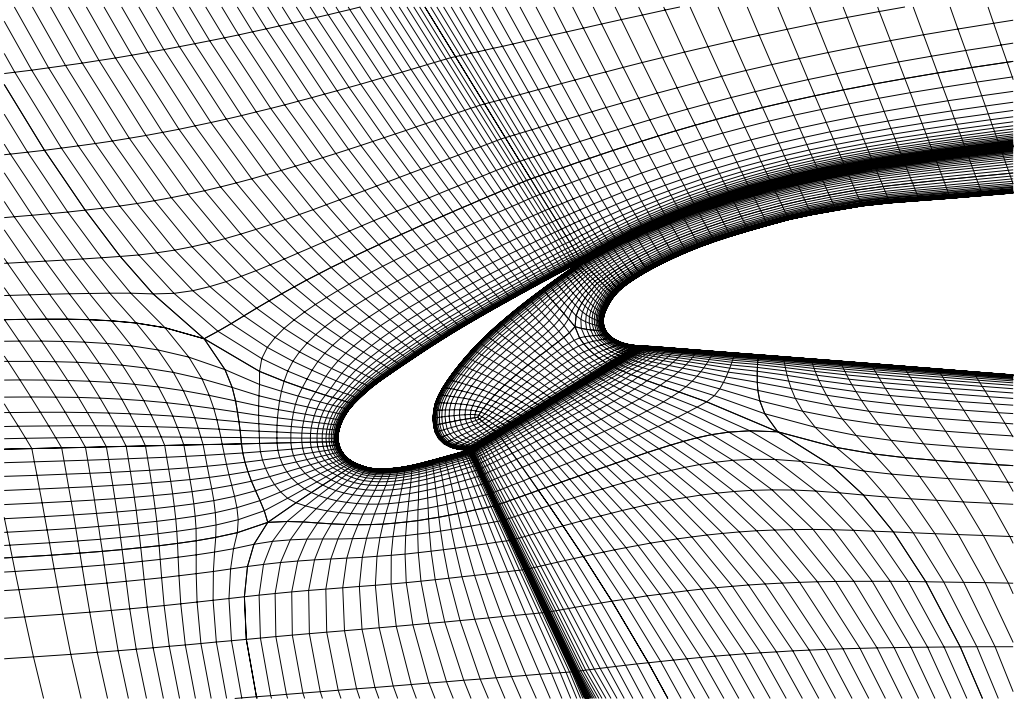


Figure 1.3: *Detail of a multiblock structured grid*

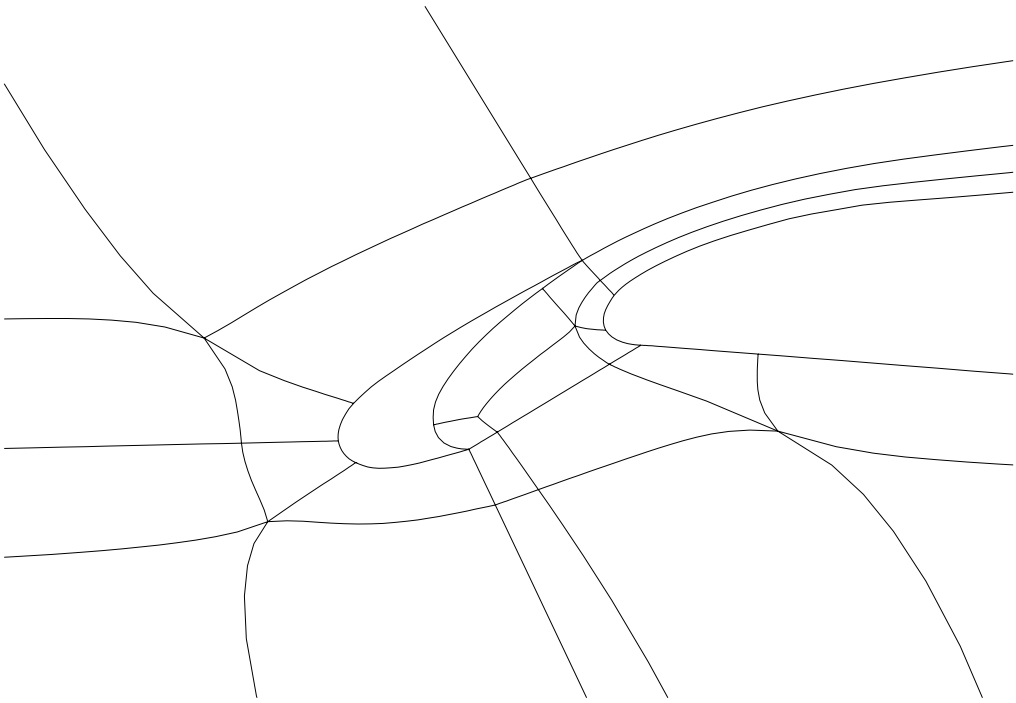


Figure 1.4: *Block outlines of a multiblock structured grid*

engineering and scientific problems. Part II (Chapters 5 and 6) focusses on the more practical issues of multiblock grid generation and parallel computation.

In Chapter 2 the adaption of an existing two-dimensional flow solver to axisymmetric flow is described in detail. A fully three-dimensional flow solver is required to simulate large, complex aerodynamic problems. However, for the special case of axisymmetric flows, an essentially two-dimensional method can be employed, requiring substantially less memory and two orders of magnitude less computation time. The surprisingly wide application range for an axisymmetric flow solver and the clear performance advantage over three-dimensional methods provides the motivation for this work.

In Chapter 3 the ability of an advanced CFD method to model two classes of flow, axisymmetric forebody and base flow, is examined. The capability of the CFD method as an engineering tool for these problems is assessed. First the adaption of an existing two-dimensional flow solver to axisymmetric flow is described, then specific test cases are considered. The accuracy of the results, the robustness and speed of the flow solver and the time and effort required for pre- and post-processing are considered. Particular emphasis is given to the prospect of routine calculation in an engineering environment for these problem types.

Computational aerodynamics has principally been viewed as a design tool which complements experimentation and theory. There is also the possibility of its use as a tool for scientific investigation[28]. The direct numerical simulation (DNS) of turbulence is a good example of this. In engineering codes turbulent flow is usually modelled using the Reynolds-averaged Navier-Stokes equations with a single-point closure turbulence model to account for the effects of turbulent motion, which is impractical to compute directly. The complexity of turbulence models varies from a modified laminar viscosity coefficient to an additional series of partial differential equations to model transport of turbulent stresses. Despite the plethora of models available, none have been accepted as generally accurate and applicable. The lack of an appropriate statistical description of the effects of turbulence is frequently quoted as the pacing item for CFD simulation. Each model contains adjustable coefficients which are determined empirically. Part of the difficulty in turbulence modelling is that experimental measurement of the modelled quantities is difficult, rendering the

empiricism unreliable. However, DNS results for simple turbulent shear flows on a small scale are becoming available, affording direct testing of turbulence models and investigation of turbulence phenomena in general.

Another area where CFD is used to promote understanding of physical phenomena is in shock reflection studies. Freestream perturbations in wind tunnel experiments on the stability of shock patterns have caused real uncertainty in interpreting results, so researchers are now relying heavily on the results of CFD simulations. In addition to improving the practicality and accuracy of simulations for design, how best to employ numerical techniques in scientific investigations such as the examples given is another challenge to the CFD community. A detailed numerical study of shock reflection hysteresis in an underexpanded jet is described in Chapter 4. This can be viewed as a model situation where experimental studies are limited and CFD can potentially play a very important role. The theoretical background and understanding of this complex phenomenon is reviewed. Recent experimental and numerical contributions in this area are discussed. The CFD method described in Chapter 3 is applied to an underexpanded jet flow. Comparison with experimental data is made where possible. The detail obtained from the CFD simulation enables identification of several flow features not initially recognised from the experiments. Understanding of the plume structures and hysteresis phenomenon have been greatly enhanced by the CFD study.

Part II concerns the more practical issues of pre-processing and parallel computing, each of which are in their own right important sub-topics in CFD. A major bottleneck in CFD analyses of complex configurations occurs at the pre-processing stage, consisting of geometry definition using computer aided design (CAD) software, interfacing the CAD model with grid generation software, and grid generation itself. Pre-processing remains a labour intensive task, especially at the grid generation stage. Several man-months of skilled effort may be required to generate a structured grid around entire aircraft configurations. Generating unstructured grids requires less effort in general, but is still very time consuming. The large amount of time and effort taken in generating grids is recognised as the major difficulty in the routine use of CFD[17],[25]. In Chapter 5 the need for automation of the multiblock grid generation process is discussed. A new approach to automatically

process a multiblock topology in order to prepare it for subsequent grid generation is described. The method is based on a cost function which attempts to model the objectives of the skilled grid generation software user who at present performs the task of block positioning and shaping in an interactive manner. A number of test cases are examined. It is also suggested that an existing unstructured mesh generation method could be adopted as an initial topology generation tool. Further work towards creating a fully automatic grid generation tool and extension into three dimensions are discussed briefly.

The exploitation of emerging technology as computing power continues to increase raises research issues in its own right. However, the large amount of research into parallel CFD is not matched by the amount of research conducted using parallel CFD as a tool[26]. The potential of using low-cost commodity processors or no-cost spare capacity on existing workstations for parallel aerodynamic simulations has recently added further to the large number of research papers devoted to parallel CFD, see for example[27]. In order for parallel CFD technology to have a greater impact on the productivity of CFD simulation, a number of practical difficulties must be fully addressed, most importantly effective parallelisation and robust, reliable execution on non-dedicated parallel machines. This is the type of resource available to small and medium sized enterprises, rather than the large organisations with access to powerful dedicated computers on which the majority of parallel CFD research has been focussed and for which parallelisation is largely a solved problem. The parallel execution of an aerodynamic simulation code on a non-dedicated, "open" cluster of workstations is examined in Chapter 6. This type of facility is commonly available to CFD developers and users in academia, industry and government laboratories and is a very attractive option to achieve an upgrade in computing resource for CFD simulations without large expenditure. However, practical considerations appear at present to be discouraging widespread adoption of this technology. The main obstacles to achieving an efficient, robust parallel CFD capability in a demanding multi-user environment are investigated. A parallelism strategy for a structured multiblock flow solver which takes account of heterogeneity of the parallel machine and of load variation due to the presence of other users is described. The emphasis is on robustness and ease of implementation, distinct from other published work in

this area.

In the concluding chapter progress towards the stated aims of this work is reviewed. General experiences gained in the course of this study are discussed, and recommendations are made for future work.

Part I

Application of an Axisymmetric Flow Solver

Chapter 2

Axisymmetric Formulation

2.1 Introduction

In order to simulate large, complex aerodynamic problems using CFD a fully three-dimensional flow solver is necessary, using a multiblock approach or otherwise to cope with geometric complexity. However, there is a class of aerodynamic flows for which it is not necessary to resort to the complexity and expense of a full 3-D method. It is possible to make use of an existing two-dimensional flow solver to develop an axisymmetric flow solver, thus achieving a level of three-dimensional capability, albeit limited to axisymmetric geometries at zero incidence and sideslip. For this type of problem an axisymmetric rather than a full 3-D solver is a more efficient tool, considering the comparatively large amount of memory and CPU time required for 3-D calculations. The equations for axisymmetric flow can be cast in a form very similar to those for planar two-dimensional flow, which can then be solved using a numerical scheme with few alterations from the planar case. Examples of interest include slender bodies, base flows and nozzle/plume flows. This effort is therefore worthwhile because the modification required is relatively straightforward and the range of application surprisingly wide.

Several examples of computational aerodynamics codes solving the axisymmetric Euler and Navier-Stokes equations appear in the literature, for example for base flow applications[35],[36],[37],[38],[39],[40],[41] and missile forebodies[42],[43],[44],[45]. Some other applications are hypersonic flow[46] and internal nozzle flow[47]. Ap-

plications to underexpanded jets are discussed in [48]. Some flow solvers use the present approach of considering the axisymmetric case as an extension of the two-dimensional case[35],[36],[39],[40],[41],[42],[44],[46],[47], thus allowing one flow solver to be used for two different types of flow. In this chapter the modification of an existing two-dimensional flow solver to axisymmetric flow is described. The flow solver will be applied to two classes of problem, supersonic forebody flow and high speed base flow. The aim of this chapter is to evaluate the effectiveness of a modern CFD method as a design and evaluation tool for these problem types. As well as examining the accuracy of the results from the flow solver, other issues which are important in an industrial context are considered: robustness, turnaround time, pre- and post-processing effort.

The flow solver used is based on the existing planar two-dimensional, steady-state flow solver PMB2D developed by the CFD group at the University of Glasgow, which has reached a level of maturity enabling application to a diverse range of aerodynamic problems. The main features of the flow solver are outlined here; for full details see [30],[31],[32],[33].

A cell-centred finite volume method is employed. Osher's scheme and MUSCL variable interpolation are used to discretise the convective terms and central differencing for the diffusive terms. The linear system arising at each implicit time step is solved using a Generalised Conjugate Gradient method. A Block Incomplete Lower-Upper (BILU) factorisation is used as a preconditioner. A structured multiblock grid system is employed. The BILU factorisation is decoupled between blocks to reduce communication, improving efficiency on distributed memory parallel computers. An important feature of the flow solver is the use of approximate Jacobian matrices for the left hand side of the linear system. This has led to substantial reductions in memory and CPU-time requirements compared to the use of exact Jacobians. The $k-\omega$ turbulence model is employed with MUSCL variable interpolation and the Engquist-Osher scalar conservation law for the convective terms.

This chapter begins by presenting the equations for axisymmetric flow. Comparison is made with the equations for two-dimensional planar flow. The alterations made to the original linear system in the implicit solution method are then discussed. The test case of laminar Poiseuille flow in a pipe is then examined. The

axisymmetric flow solver will be applied to engineering and scientific problems in subsequent chapters.

2.2 The equations for axisymmetric flow

2.2.1 Introduction

The Navier-Stokes equations and the two-equation $k-\omega$ turbulence model are presented in forms suitable for axisymmetric flow; the equations are written in cylindrical coordinates (r, θ, z) with the assumptions of zero angle of incidence and sideslip ($\partial/\partial\theta = 0$) and no spin ($v_\theta = 0$). A large part of their derivation from general vector/tensor forms is also included for completeness. This should help to highlight the origins and purpose of the 'extra' terms present in the axisymmetric equations compared to the two-dimensional equations.

2.2.2 Mass continuity

The equation of mass conservation, or continuity equation, is written in conservation form as [50],[51]:

$$\frac{\partial \rho}{\partial t} + \nabla.(\rho \mathbf{V}) = 0 \tag{2.1}$$

where ρ is the fluid density and $\mathbf{V}$ is the velocity vector which has components (v_r, v_θ, v_z) in cylindrical coordinates. The divergence of a vector $\mathbf{A}$ in a cylindrical frame is

$$\nabla.\mathbf{A} = \frac{\partial A_r}{\partial r} + \frac{A_r}{r} + \frac{1}{r}\frac{\partial A_\theta}{\partial \theta} + \frac{\partial A_z}{\partial z} \tag{2.2}$$

and therefore in a cylindrical frame equation (2.1) becomes

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial r}(\rho v_r) + \frac{1}{r}\frac{\partial}{\partial \theta}(\rho v_\theta) + \frac{\partial}{\partial z}(\rho v_z) = -\frac{\rho v_r}{r} \tag{2.3}$$

With our assumptions of axisymmetric flow with no spin this reduces to

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial r}(\rho v_r) + \frac{\partial}{\partial z}(\rho v_z) = -\frac{\rho v_r}{r} \tag{2.4}$$

2.2.3 Momentum conservation

The equations of motion or momentum equations neglecting body forces can be written in vector form as [50]

$$\rho\frac{\partial \mathbf{V}}{\partial t} + \rho\mathbf{V}.\nabla\mathbf{V} + \nabla p - \frac{\mu}{3}\nabla\left(\nabla.\mathbf{V}\right) - \mu\nabla^2\mathbf{V} \;=\; 0 \tag{2.5}$$

where $\mathbf{V}$ is the velocity vector. When expanding the vector terms in equation (2.5) it should be recalled that in cylindrical coordinates the unit vectors are not invariant in space [51],[52]. Following equation (2.2), the divergence of velocity in cylindrical coordinates is given by

$$\nabla.\mathbf{V} \;=\; \frac{\partial v_r}{\partial r} + \frac{v_r}{r} + \frac{1}{r}\frac{\partial v_\theta}{\partial \theta} + \frac{\partial v_z}{\partial z} \tag{2.6}$$

The strain tensor $\nabla\mathbf{V}$ in cylindrical coordinates is:

$$\nabla\mathbf{V} \;=\; \begin{pmatrix} \frac{\partial v_r}{\partial r} & \frac{\partial v_\theta}{\partial r} & \frac{\partial v_z}{\partial r} \\ \frac{1}{r}\frac{\partial v_r}{\partial \theta} - \frac{v_\theta}{r} & \frac{1}{r}\frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r} & \frac{1}{r}\frac{\partial v_z}{\partial \theta} \\ \frac{\partial v_r}{\partial z} & \frac{\partial v_\theta}{\partial z} & \frac{\partial v_z}{\partial z} \end{pmatrix} \tag{2.7}$$

Expanding the Laplacian of the velocity vector in cylindrical coordinates gives

$$\nabla^2\mathbf{V} \;=\; \nabla.\left(\nabla\mathbf{V}\right) \;=$$

$$\begin{pmatrix} \frac{\partial}{\partial r}\left(\frac{\partial v_r}{\partial r}\right) + \frac{1}{r}\frac{\partial v_r}{\partial r} + \frac{1}{r}\frac{\partial}{\partial \theta}\left(\frac{1}{r}\frac{\partial v_r}{\partial \theta} - \frac{v_\theta}{r}\right) + \frac{\partial}{\partial z}\left(\frac{\partial v_r}{\partial z}\right) - \frac{1}{r}\left(\frac{1}{r}\frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r}\right) \\ \frac{\partial}{\partial r}\left(\frac{\partial v_\theta}{\partial r}\right) + \frac{1}{r}\frac{\partial v_\theta}{\partial r} + \frac{1}{r}\frac{\partial}{\partial \theta}\left(\frac{1}{r}\frac{\partial v_\theta}{\partial \theta} + \frac{v_r}{r}\right) + \frac{\partial}{\partial z}\left(\frac{\partial v_\theta}{\partial z}\right) - \frac{1}{r}\left(\frac{1}{r}\frac{\partial v_r}{\partial \theta} - \frac{v_\theta}{r}\right) \\ \frac{\partial}{\partial r}\left(\frac{\partial v_z}{\partial r}\right) + \frac{1}{r}\frac{\partial v_z}{\partial r} + \frac{1}{r}\frac{\partial}{\partial \theta}\left(\frac{1}{r}\frac{\partial v_z}{\partial \theta}\right) + \frac{\partial}{\partial z}\left(\frac{\partial v_z}{\partial z}\right) \end{pmatrix} \tag{2.8}$$

Finally the pressure gradient term is

$$\nabla p \;=\; \begin{pmatrix} \frac{\partial p}{\partial r} \\ \frac{1}{r}\frac{\partial p}{\partial \theta} \\ \frac{\partial p}{\partial z} \end{pmatrix} \tag{2.9}$$

The equations (2.5) to (2.9) are the momentum conservation equations in cylindrical coordinates. It is convenient at this stage to introduce our assumptions concerning axisymmetry, viz. $\partial/\partial\theta = 0$ and $v_\theta = 0$. The momentum equations in the radial (r) direction and axial (z) direction then become respectively

$$\begin{aligned}\rho\frac{\partial v_r}{\partial t} + \rho\left(v_r\frac{\partial v_r}{\partial r} + v_z\frac{\partial v_r}{\partial z}\right) + \frac{\partial p}{\partial r} - \frac{\mu}{3}\frac{\partial}{\partial r}\left(\frac{\partial v_r}{\partial r} + \frac{v_r}{r} + \frac{\partial v_z}{\partial z}\right) \\ -\mu\left[\frac{\partial}{\partial r}\left(\frac{\partial v_r}{\partial r}\right) + \frac{1}{r}\frac{\partial v_r}{\partial r} + \frac{\partial}{\partial z}\left(\frac{\partial v_r}{\partial z}\right) - \frac{v_r}{r^2}\right] &= 0 \qquad (2.10)\end{aligned}$$

$$\begin{aligned}\rho\frac{\partial v_z}{\partial t} + \rho\left(v_r\frac{\partial v_z}{\partial r} + v_z\frac{\partial v_z}{\partial z}\right) + \frac{\partial p}{\partial z} - \frac{\mu}{3}\frac{\partial}{\partial z}\left(\frac{\partial v_r}{\partial r} + \frac{v_r}{r} + \frac{\partial v_z}{\partial z}\right) \\ -\mu\left[\frac{\partial}{\partial r}\left(\frac{\partial v_z}{\partial r}\right) + \frac{1}{r}\frac{\partial v_z}{\partial r} + \frac{\partial}{\partial z}\left(\frac{\partial v_z}{\partial z}\right)\right] &= 0 \qquad (2.11)\end{aligned}$$

These equations can be simplified, using the continuity equation (2.4), to

$$\frac{\partial}{\partial t}(\rho v_r) + \frac{\partial}{\partial r}\left(\rho v_r^2 + p\right) - \frac{\partial \tau_{rr}}{\partial r} + \frac{\partial}{\partial z}(\rho v_r v_z) - \frac{\partial \tau_{rz}}{\partial z} = -\frac{\rho v_r^2}{r} + \frac{(\tau_{rr} - \tau_{\theta\theta})}{r} \qquad (2.12)$$

$$\frac{\partial}{\partial t}(\rho v_z) + \frac{\partial}{\partial r}(\rho v_r v_z) - \frac{\partial \tau_{rz}}{\partial r} + \frac{\partial}{\partial z}\left(\rho v_z^2 + p\right) - \frac{\partial \tau_{zz}}{\partial z} = -\frac{\rho v_r v_z}{r} + \frac{\tau_{rz}}{r} \qquad (2.13)$$

where the shear stress components are written as

$$\begin{aligned}\tau_{rr} &= -\mu\left(2\frac{\partial v_r}{\partial r} - \frac{2}{3}\left(\frac{\partial v_r}{\partial r} + \frac{v_r}{r} + \frac{\partial v_z}{\partial z}\right)\right) \\ \tau_{zz} &= -\mu\left(2\frac{\partial v_z}{\partial z} - \frac{2}{3}\left(\frac{\partial v_r}{\partial r} + \frac{v_r}{r} + \frac{\partial v_z}{\partial z}\right)\right) \\ \tau_{\theta\theta} &= -\mu\left(2\frac{v_r}{r} - \frac{2}{3}\left(\frac{\partial v_r}{\partial r} + \frac{v_r}{r} + \frac{\partial v_z}{\partial z}\right)\right) \\ \tau_{rz} &= -\mu\left(\frac{\partial v_r}{\partial z} + \frac{\partial v_z}{\partial r}\right)\end{aligned} \qquad (2.14)$$

As will be seen later, it is convenient to re-arrange equations (2.12) and (2.13) to the following form (which resembles the planar equations)

$$\begin{aligned}\frac{\partial}{\partial t}(\rho v_r) + \frac{\partial}{\partial r}\left(\rho v_r^2 + p\right) - \frac{\partial}{\partial r}\left[\mu\left(2\frac{\partial v_r}{\partial r} - \frac{2}{3}\left(\frac{\partial v_z}{\partial z} + \frac{\partial v_r}{\partial r}\right)\right)\right] \\ + \frac{\partial}{\partial z}(\rho v_r v_z) - \frac{\partial}{\partial z}\left[\mu\left(\frac{\partial v_z}{\partial r} + \frac{\partial v_r}{\partial z}\right)\right] &= -\frac{\rho v_r^2}{r} + \frac{4\mu}{3}\left(\frac{1}{r}\frac{\partial v_r}{\partial r} - \frac{v_r}{r^2}\right)\end{aligned} \qquad (2.15)$$

$$\begin{aligned}\frac{\partial}{\partial t}(\rho v_z)+\frac{\partial}{\partial r}(\rho v_r v_z)-\frac{\partial}{\partial r}\left[\mu\left(\frac{\partial v_z}{\partial r}+\frac{\partial v_r}{\partial z}\right)\right]+\frac{\partial}{\partial z}\left(\rho v_z^2+p\right)&\\ -\frac{\partial}{\partial z}\left[\mu\left(2\frac{\partial v_z}{\partial z}-\frac{2}{3}\left(\frac{\partial v_z}{\partial z}+\frac{\partial v_r}{\partial r}\right)\right)\right] &= -\frac{\rho v_r v_z}{r}+\frac{\mu}{r}\left(\frac{1}{3}\frac{\partial v_r}{\partial z}+\frac{\partial v_z}{\partial r}\right)\end{aligned} \tag{2.16}$$

2.2.4 Energy equation

The equation for the conservation of energy can be written as[51], [53]

$$\rho\frac{de}{dt}+p\left(\nabla.\mathbf{V}\right) = -\nabla.\mathbf{q}+\frac{\partial Q}{\partial t}+\Phi \tag{2.17}$$

where e is the internal energy per unit mass, $\mathbf{q}$ is the heat transfer vector and Q is the heat added per unit volume by external agencies. Φ is the dissipation function, which can be written as

$$\Phi = \mu\left[\left(\nabla\mathbf{V}+\nabla\mathbf{V}^T\right):\nabla\mathbf{V}-\frac{2}{3}\left(\nabla.\mathbf{V}\right)^2\right] \tag{2.18}$$

We are interested in the form of the energy conservation equation suitable for axisymmetric flow. In cylindrical coordinates, with the assumptions that $\partial/\partial\theta = 0$ and $v_\theta = 0$, the dissipation function becomes

$$\begin{aligned}\Phi = \mu\Bigg[&2\left(\frac{\partial u_r}{\partial r}\right)^2+2\left(\frac{u_r}{r}\right)^2+2\left(\frac{\partial u_z}{\partial z}\right)^2+\left(\frac{\partial u_z}{\partial r}+\frac{\partial u_r}{\partial z}\right)^2\\ &-\frac{2}{3}\left(\frac{\partial u_r}{\partial r}+\frac{u_r}{r}+\frac{\partial u_z}{\partial z}\right)^2\Bigg]\end{aligned} \tag{2.19}$$

Equation 2.17 can then be written, with the same assumptions, in the form

$$\begin{aligned}\rho\frac{de}{dt}+p\left(\frac{\partial v_r}{\partial r}+\frac{v_r}{r}+\frac{\partial v_z}{\partial z}\right) &= -\frac{\partial q_r}{\partial r}-\frac{q_r}{r}-\frac{\partial q_z}{\partial z}\\ &+\frac{\partial v_r}{\partial r}\tau_{rr}+\left(\frac{\partial v_r}{\partial z}+\frac{\partial v_z}{\partial r}\right)\tau_{rz}+\frac{\partial v_z}{\partial z}\tau_{zz}+\frac{v_r}{r}\tau_{\theta\theta}\end{aligned} \tag{2.20}$$

assuming also that there is also no external heat addition. It can be shown using the continuity equation (2.4) that

$$\frac{\partial}{\partial t}(\rho v_r)+\frac{\partial}{\partial r}\left(\rho v_r^2\right)+\frac{\partial}{\partial z}(\rho v_r v_z)+\frac{\rho v_r^2}{r} = \rho\frac{dv_r}{dt} \tag{2.21}$$

$$\frac{\partial}{\partial t}(\rho v_z)+\frac{\partial}{\partial r}(\rho v_r v_z)+\frac{\partial}{\partial z}\left(\rho v_z^2\right)+\frac{\rho v_r v_z}{r} = \rho\frac{dv_z}{dt} \tag{2.22}$$

The following equation is obtained by substituting equation (2.21) into (2.12) multiplied by v_r, and adding this to the equation obtained by substituting (2.22) into equation (2.13) multiplied by v_z :

$$\rho\frac{d}{dt}\left(v_r^2/2+v_z^2/2\right)+v_r\frac{\partial p}{\partial r}+v_z\frac{\partial p}{\partial z}-v_r\frac{\partial \tau_{rr}}{\partial r}-v_r\frac{\partial \tau_{rz}}{\partial z}-v_z\frac{\partial \tau_{rz}}{\partial r}-v_z\frac{\partial \tau_{zz}}{\partial z} = \frac{v_r\left(\tau_{rr}-\tau_{\theta\theta}\right)}{r}+\frac{v_z\tau_{rz}}{r} \quad (2.23)$$

An equation representing the conservation of energy per unit mass is then obtained by adding together equations (2.20) and (2.23) :

$$\rho\frac{d}{dt}\left(e+v_r^2/2+v_z^2/2\right)+\frac{\partial}{\partial r}\left(pv_r+q_r\right)+\frac{pv_r}{r}+\frac{\partial}{\partial z}\left(pv_z+q_z\right) -\frac{\partial}{\partial r}\left(v_r\tau_{rr}+v_z\tau_{rz}\right)-\frac{\partial}{\partial z}\left(v_z\tau_{zz}+v_r\tau_{rz}\right) = \frac{v_r\tau_{rr}+v_z\tau_{rz}-q_r}{r} \quad (2.24)$$

The total energy per unit volume E_t is calculated as

$$E_t = \rho\left(e+\frac{1}{2}\left(v_r^2+v_z^2\right)\right)$$

It can be shown using the continuity equation (2.4) that

$$\rho\frac{d}{dt}\left(E_t/\rho\right) = \frac{\partial E_t}{\partial t}+\frac{\partial}{\partial r}\left(v_rE_t\right)+\frac{v_rE_t}{r}+\frac{\partial}{\partial z}\left(v_zE_t\right)$$

Hence equation (2.24) becomes

$$\frac{\partial E_t}{\partial t}+\frac{\partial}{\partial r}\left(v_r\left(E_t+p\right)\right)+\frac{\partial}{\partial z}\left(v_z\left(E_t+p\right)\right)-\frac{\partial}{\partial r}\left(v_r\tau_{rr}+v_z\tau_{rz}-q_r\right) -\frac{\partial}{\partial z}\left(v_z\tau_{zz}+v_r\tau_{rz}-q_z\right) = \frac{-v_r\left(E_t+p\right)+v_r\tau_{rr}+v_z\tau_{rz}-q_r}{r} \quad (2.25)$$

As will be seen later, it is convenient here to express this equation in the following form (which resembles the planar equations):

$$\frac{\partial E_t}{\partial t}+\frac{\partial}{\partial r}\left(v_r\left(E_t+p\right)\right)+\frac{\partial}{\partial z}\left(v_z\left(E_t+p\right)\right) -\frac{\partial}{\partial r}\left\{v_r\left[\mu\left(2\frac{\partial u_r}{\partial r}-\frac{2}{3}\left(\frac{\partial u_r}{\partial r}+\frac{\partial u_z}{\partial z}\right)\right)\right]+v_z\left[\mu\left(\frac{\partial u_r}{\partial z}+\frac{\partial u_z}{\partial r}\right)\right]-q_r\right\} -\frac{\partial}{\partial z}\left\{v_z\left[\mu\left(2\frac{\partial u_z}{\partial z}-\frac{2}{3}\left(\frac{\partial u_r}{\partial r}+\frac{\partial u_z}{\partial z}\right)\right)\right]+v_r\left[\mu\left(\frac{\partial u_r}{\partial z}+\frac{\partial u_z}{\partial r}\right)\right]-q_z\right\} = \frac{1}{r}\left[-v_r\left(E_t+p\right)+\mu\left(v_z\frac{\partial u_z}{\partial r}+\frac{v_z}{3}\frac{\partial u_r}{\partial z}-\frac{4v_r}{3}\frac{\partial u_z}{\partial z}\right)-q_r\right] \quad (2.26)$$

2.2.5 Non-dimensional, Reynolds-averaged form

The equations shown above are in dimensional form. In practise it is more convenient to use non-dimensional quantities. The procedure used for non-dimensionalising is described in sections A.2 and B.1. The Reynolds-averaging procedure, see appendix A.3, enables consideration of turbulent flow. The equations for mass continuity (2.4), momentum (2.16 and 2.15) and energy (2.26) become in non-dimensional, Reynolds-averaged form :

Mass continuity

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial r}\left(\rho v_r\right) + \frac{\partial}{\partial z}\left(\rho v_z\right) \;=\; -\frac{\rho v_r}{r} \tag{2.27}$$

Momentum

$$\begin{aligned}\frac{\partial}{\partial t}\left(\rho v_r\right) + \frac{\partial}{\partial r}\left(\rho v_r^2 + p\right) - \frac{\partial}{\partial r}\left[\frac{\mu+\mu_T}{Re}\left(2\frac{\partial v_r}{\partial r} - \frac{2}{3}\left(\frac{\partial v_z}{\partial z} + \frac{\partial v_r}{\partial r}\right) + \frac{2}{3}\rho k\right)\right] \\ + \frac{\partial}{\partial z}\left(\rho v_r v_z\right) - \frac{\partial}{\partial z}\left[\frac{\mu+\mu_T}{Re}\left(\frac{\partial v_z}{\partial r} + \frac{\partial v_r}{\partial z}\right)\right] \;=\; -\frac{\rho v_r^2}{r} + \frac{4\left(\mu+\mu_T\right)}{3Re}\left(\frac{1}{r}\frac{\partial v_r}{\partial r} - \frac{v_r}{r^2}\right)\end{aligned} \tag{2.28}$$

$$\begin{aligned}\frac{\partial}{\partial t}\left(\rho v_z\right) + \frac{\partial}{\partial r}\left(\rho v_r v_z\right) - \frac{\partial}{\partial r}\left[\frac{\mu+\mu_T}{Re}\left(\frac{\partial v_z}{\partial r} + \frac{\partial v_r}{\partial z}\right)\right] + \frac{\partial}{\partial z}\left(\rho v_z^2 + p\right) \\ - \frac{\partial}{\partial z}\left[\frac{\mu+\mu_T}{Re}\left(2\frac{\partial v_z}{\partial z} - \frac{2}{3}\left(\frac{\partial v_z}{\partial z} + \frac{\partial v_r}{\partial r}\right) + \frac{2}{3}\rho k\right)\right] \;=\; -\frac{\rho v_r v_z}{r} + \frac{\mu+\mu_T}{rRe}\left(\frac{1}{3}\frac{\partial v_r}{\partial z} + \frac{\partial v_z}{\partial r}\right)\end{aligned} \tag{2.29}$$

Energy

$$\begin{aligned}
&\frac{\partial E_t}{\partial t} + \frac{\partial}{\partial r}\left(v_r\left(E_t+p\right)\right) + \frac{\partial}{\partial z}\left(v_z\left(E_t+p\right)\right) \\
&\quad -\frac{\partial}{\partial r}\left\{v_r\left[\frac{\mu+\mu_T}{Re}\left(2\frac{\partial u_r}{\partial r} - \frac{2}{3}\left(\frac{\partial u_r}{\partial r}+\frac{\partial u_z}{\partial z}\right)+\frac{2}{3}\rho k\right)\right] + v_z\left[\frac{\mu+\mu_T}{Re}\left(\frac{\partial u_r}{\partial z}+\frac{\partial u_z}{\partial r}\right)\right]\right\} \\
&\quad -\frac{\partial}{\partial z}\left\{v_z\left[\frac{\mu+\mu_T}{Re}\left(2\frac{\partial u_z}{\partial z} - \frac{2}{3}\left(\frac{\partial u_r}{\partial r}+\frac{\partial u_z}{\partial z}\right)+\frac{2}{3}\rho k\right)\right] + v_r\left[\frac{\mu+\mu_T}{Re}\left(\frac{\partial u_r}{\partial z}+\frac{\partial u_z}{\partial r}\right)\right]\right\} \\
&\quad -\frac{\partial}{\partial r}\left\{\frac{1}{\left(\gamma-1\right)M_\infty^2}\left(\frac{\mu}{Pr}+\frac{\mu_T}{Pr_T}\right)\frac{\partial T}{\partial r}\right\} - \frac{\partial}{\partial z}\left\{\frac{1}{\left(\gamma-1\right)M_\infty^2}\left(\frac{\mu}{Pr}+\frac{\mu_T}{Pr_T}\right)\frac{\partial T}{\partial z}\right\} \\
&\quad = \frac{1}{r}\left[-v_r\left(E_t+p\right) + \frac{\mu+\mu_T}{Re}\left(v_z\frac{\partial u_z}{\partial r} + \frac{v_z}{3}\frac{\partial u_r}{\partial z} - \frac{4v_r}{3}\frac{\partial u_z}{\partial z} + \frac{2v_r}{3}\rho k\right)\right. \\
&\quad \left. + \frac{1}{\left(\gamma-1\right)M_\infty^2}\left(\frac{\mu}{Pr}+\frac{\mu_T}{Pr_T}\right)\frac{\partial T}{\partial r}\right]
\end{aligned} \tag{2.30}$$

2.2.6 The two-equation $k-\omega$ turbulence model

The $k-\omega$ turbulence model of Wilcox[54] is written in non-dimensional, general vector-tensor form in appendix B.1. In cylindrical coordinates, for axisymmetric flow with no spin, this becomes

Turbulence Kinetic Energy

$$\begin{aligned}
&\frac{\partial}{\partial t}\left(\rho k\right) + \frac{\partial}{\partial r}\left(\rho k v_r\right) + \frac{\partial}{\partial z}\left(\rho k v_z\right) - \frac{1}{Re}\left\{\frac{\partial}{\partial r}\left[\left(\mu+\sigma^*\mu_T\right)\frac{\partial k}{\partial r}\right] + \frac{\partial}{\partial z}\left[\left(\mu+\sigma^*\mu_T\right)\frac{\partial k}{\partial z}\right]\right\} \\
&\qquad = \mu_T P - \frac{2}{3}\rho k S - \beta^*\rho k\omega - \frac{\rho k v_r}{r} + \frac{1}{Re}\left\{\frac{1}{r}\left[\left(\mu+\sigma^*\mu_T\right)\frac{\partial k}{\partial r}\right]\right\}
\end{aligned} \tag{2.31}$$

Specific Dissipation Rate

$$\begin{aligned}
&\frac{\partial}{\partial t}\left(\rho\omega\right) + \frac{\partial}{\partial r}\left(\rho\omega v_r\right) + \frac{\partial}{\partial z}\left(\rho\omega v_z\right) - \frac{1}{Re}\left\{\frac{\partial}{\partial r}\left[\left(\mu+\sigma\mu_T\right)\frac{\partial \omega}{\partial r}\right] + \frac{\partial}{\partial z}\left[\left(\mu+\sigma\mu_T\right)\frac{\partial \omega}{\partial z}\right]\right\} \\
&\qquad = \alpha\frac{\omega}{k}\left[\mu_T P - \frac{2}{3}\rho k S\right] - \beta\rho\omega^2 - \frac{\rho\omega v_r}{r} + \frac{1}{Re}\left\{\frac{1}{r}\left[\left(\mu+\sigma\mu_T\right)\frac{\partial \omega}{\partial r}\right]\right\}
\end{aligned} \tag{2.32}$$

In the above relations,

$$P = \left(\frac{\partial v_z}{\partial r} + \frac{\partial v_r}{\partial z}\right)^2 + 2\left[\left(\frac{\partial v_r}{\partial r}\right)^2 + \left(\frac{\partial v_z}{\partial z}\right)^2 + \left(\frac{v_r}{r}\right)^2\right] - \frac{2}{3}\left(\frac{\partial v_r}{\partial r} + \frac{\partial v_z}{\partial z} + \frac{v_r}{r}\right)^2$$

$$S = \frac{\partial v_r}{\partial r} + \frac{\partial v_z}{\partial z} + \frac{v_r}{r}$$

2.2.7 Curvilinear form

Mean flow equations

Compare the equations for axisymmetric flow, equations (2.27) to (2.30), with those for planar flow (see appendix A), swapping the radial ordinate r for y and the axial ordinate z for x; the left hand sides of the equations are identical. Hence the axisymmetric equations can be considered as consisting of the 2-D equations plus a source-like correction term for axisymmetry. See section (2.3.1) for a discussion of the numerical implications. The transformation of the left-hand side of the equations into (ξ, η) space is therefore identical to that described for the two-dimensional equations in appendix A.4. The right hand side of the transformed system of equations is written simply as

$$\frac{1}{J}\left(\hat{\mathbf{S}}^i + \hat{\mathbf{S}}^v\right) \tag{2.33}$$

after splitting the source-like term into inviscid and viscous parts.

Two equation $k-\omega$ turbulence model

The axisymmetric (equations 2.31 and 2.32) and the two-dimensional (see appendix B) formulations for the k and ω equations can be compared in a similar manner to above; the axisymmetric equations can be considered as consisting of the 2-D equations plus a correction for axisymmetry. This correction is treated as an additional source term. Again the transformation of the left-hand side of the equations into (ξ, η) space is the same as for the planar equations, see equation (B.2). The right hand side of the transformed system of equations can be written as

$$\frac{1}{J}\left(\hat{\mathbf{S}}_T + \hat{\mathbf{S}}_T^a\right) \tag{2.34}$$

where $\hat{\mathbf{S}}_T$ is the 'original' source term from the two-dimensional equations and $\hat{\mathbf{S}}_T^a$ contains the additional axisymmetric terms.

2.3 Modified numerical scheme

2.3.1 Axisymmetric source terms

In the present method, the equations for axisymmetric flow are formulated to look like the planar flow equations except for a non-zero right-hand side which is treated as a source term. The fluxes on the left-hand side are treated as in the planar case. In this way an existing planar flow solver can be modified easily for axisymmetric flow. This approach, which we will call here approach A, is popular in the literature, see[35],[36],[39],[40],[41],[42],[44],[46],[47]. Another approach appearing in the literature[44],[45],[49], approach B, uses an alternative formulation. The equations (2.35) show the axisymmetric Euler equations written in this manner. In this approach the source terms of approach A do not appear, being contained in the radial flux terms on the left-hand side. The source term here consists only of a pressure term in the radial momentum equation. The manner in which the fluxes are calculated for approach B cannot be taken directly from a planar method since the flux quantities are different.

$$\begin{aligned}
\frac{\partial}{\partial t}(r\rho) + \frac{\partial}{\partial r}(r\rho v_r) + \frac{\partial}{\partial z}(r\rho v_z) &= 0 \\
\frac{\partial}{\partial t}(r\rho v_r) + \frac{\partial}{\partial r}\left(r\left[\rho v_r^2 + p\right]\right) + \frac{\partial}{\partial z}(r\rho v_r v_z) &= p \\
\frac{\partial}{\partial t}(r\rho v_z) + \frac{\partial}{\partial r}(r\rho v_r v_z) + \frac{\partial}{\partial z}\left(r\left[\rho v_z^2 + p\right]\right) &= 0 \\
\frac{\partial}{\partial t}(r\rho E) + \frac{\partial}{\partial r}(r\rho v_r H) + \frac{\partial}{\partial z}(r\rho v_z H) &= 0
\end{aligned} \tag{2.35}$$

Good results are reported in the literature for both approaches and neither approach is reported to out-perform the other concerning accuracy or numerical implementation issues. Accepting then that both approaches are valid, it is nonetheless interesting here to briefly discuss and compare the approaches since such a discussion does not appear in the literature, and at the same time hopefully gain some in-

sight into the physical meaning of the source terms. For guidance we can refer to the application of the integral form of the conservation laws to a control volume fixed in space, which can form part of the derivation of the partial differential form of the equations [50]. This will shed some light on the origin and purpose of the source terms. Diagrams of control volumes for derivations in Cartesian (x, y, z) and cylindrical (r, θ, z) space are shown in Figures 2.1 and 2.2.

Note that in Figure 2.2, the areas of the faces in the (θ, z) plane of the control volume are not equal; one has area $(r - dr/2)$ and the other $(r + dr/2)$. Note also that a pressure force acting normal to the control volume faces which are of area $drdz$ has a component in the radial direction. This means that when the integral forms of the conservation laws are evaluated for this control volume, involving fluxes through and normal stresses acting on each face, terms are retained in the resulting equations which cancel out due to symmetry in the equivalent procedure for the Cartesian control volume. These terms are the axisymmetric source terms. An example is shown below; first the equation for conservation of x-momentum is derived using the Cartesian control volume, then the radial momentum equation is derived using the cylindrical control volume and assuming $\partial/\partial\theta = 0$ and $v_\theta = 0$. The equation for the conservation of momentum, discounting viscous effects and heat transfer, can be written in integral form as [50]

$$\frac{\partial}{\partial t}\int_\Omega \rho \mathbf{V} d\Omega \;+\; \oint_S \rho \mathbf{V}\,(\mathbf{V}.d\mathbf{S}) \;=\; -\oint_S p\, d\mathbf{S} \tag{2.36}$$

where Ω denotes the control volume and S its surface.

Momentum conservation in Cartesian coordinates (x-direction)

Refer to equation (2.36) and Figure 2.1 :

$$\begin{aligned}\frac{\partial(\rho u)}{\partial t}dxdydz - dydz\left[\rho u^2 - \frac{\partial(\rho u^2)}{\partial x}\frac{dx}{2} - \left(\rho u^2 - \frac{\partial(\rho u^2)}{\partial x}\frac{dx}{2}\right)\right] \\ - dxdz\left[\rho uv - \frac{\partial(\rho uv)}{\partial y}\frac{dy}{2} - \left(\rho uv - \frac{\partial(\rho uv)}{\partial y}\frac{dy}{2}\right)\right] \\ - dxdy\left[\rho uw - \frac{\partial(\rho uw)}{\partial z}\frac{dz}{2} - \left(\rho uw - \frac{\partial(\rho uw)}{\partial z}\frac{dz}{2}\right)\right] \\ = \; dydz\left[p - \frac{\partial p}{\partial x}\frac{dx}{2} - \left(p - \frac{\partial p}{\partial x}\frac{dx}{2}\right)\right]\end{aligned}$$

which reduces to

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}\left(\rho u^2 + p\right) + \frac{\partial}{\partial y}(\rho uv) + \frac{\partial}{\partial z}(\rho uw) = 0$$

Momentum conservation in cylindrical coordinates with axisymmetry (r-direction)

Refer to equation (2.36) and Figure 2.2 :

$$\begin{aligned}\frac{\partial(\rho v_r)}{\partial t} r d\theta dr dz - r d\theta dr \left[\rho v_r v_z - \frac{\partial(\rho v_r v_z)}{\partial z}\frac{dz}{2}\right] + r d\theta dr \left[\rho v_r v_z + \frac{\partial(\rho v_r v_z)}{\partial z}\frac{dz}{2}\right] \\ - \left(r - \frac{dr}{2}\right) d\theta dz \left[\rho v_r^2 - \frac{\partial\left(\rho v_r^2\right)}{\partial r}\frac{dr}{2}\right] + \left(r + \frac{dr}{2}\right) d\theta dz \left[\rho v_r^2 + \frac{\partial\left(\rho v_r^2\right)}{\partial r}\frac{dr}{2}\right] \\ = \left(r - \frac{dr}{2}\right) d\theta dz \left[p - \frac{\partial p}{\partial r}\frac{dr}{2}\right] - \left(r + \frac{dr}{2}\right) d\theta dz \left[p + \frac{\partial p}{\partial r}\frac{dr}{2}\right] \\ + p\frac{d\theta}{2} dr dx + p\frac{d\theta}{2} dr dx\end{aligned}$$

which reduces to

$$\frac{\partial}{\partial t}(\rho v_r) + \frac{\partial}{\partial r}\left(\rho v_r^2 + p\right) + \frac{\partial}{\partial z}(\rho v_r v_z) = -\frac{\rho v_r^2}{r}$$

The axisymmetric source terms can be interpreted physically as the additional mass, momentum or energy, compared to the planar case, which enters the control volume normal to the (r, z) plane due to the axisymmetry of the flow. The effect of these terms is therefore equivalent to the effect of a surface source acting on the (x, y) plane in the planar case. Restated, the axisymmetric equations written as in sections 2.2.5 and 2.2.6 can be considered as the planar two-dimensional equations with additional surface sources of mass, momentum and energy which account for the shape of the control volume in cylindrical coordinates. We can therefore conclude that the present treatment of our right-hand side as source terms, approach A, is reasonable. Approach B may be attractive to the researcher developing an axisymmetric flow solver 'from scratch' due to the neater appearance of the governing equations when written this way. The inclusion of the radial ordinate in the flux quantities, a feature which does not occur naturally from a direct application of the integral form of the conservation laws as shown above, does appear slightly artificial in that it is difficult to interpret physically.

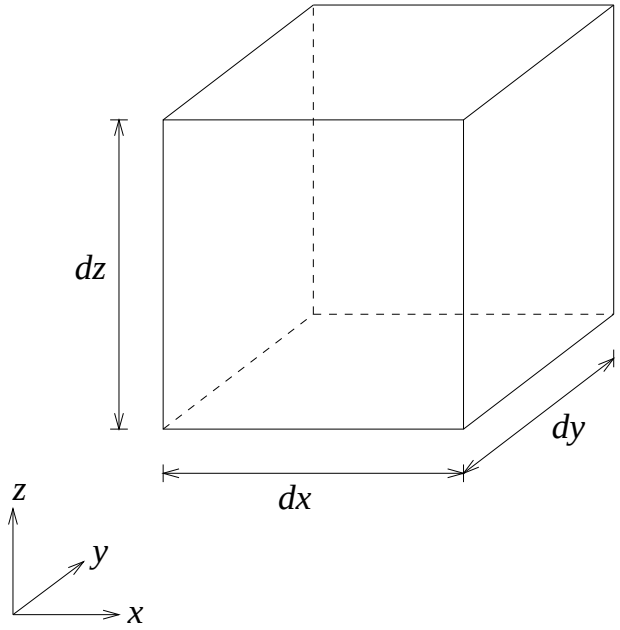


Figure 2.1: *Volume element in Cartesian coordinates*

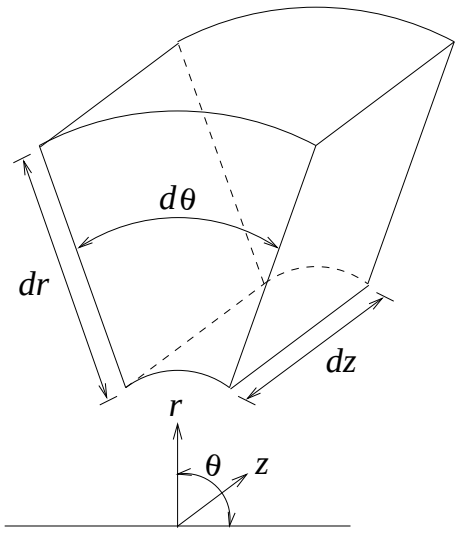


Figure 2.2: *Volume element in cylindrical coordinates*

2.3.2 Implicit scheme, mean flow equations

The integration in time of the discretised equations to a steady state is done using an implicit time-marching scheme. The linear system arising at each time step for the 2-D planar formulation can be summarised as [31] :

$$\left(\frac{I}{\Delta t}\frac{\partial \mathbf{W}}{\partial \mathbf{P}} + \frac{\partial \mathbf{R}}{\partial \mathbf{P}}\right)\Delta \mathbf{P} \;=\; -\mathbf{R}^{(n)} \tag{2.37}$$

where $\mathbf{P} = 1/J\,(\rho, v_r, v_z, p)^T$ is the vector of cell based primitive variables, $\mathbf{W} = 1/J\,(\rho, \rho v_r, \rho v_z, \rho E)^T$ is the vector of cell based conservative variables, $\Delta\mathbf{P}$ is the update in $\mathbf{P}$ from time level n to n+1, $\mathbf{R}^n$ is the flux residual arising from the spatial discretisation at the time level n, and Δt is the time step. The updates are written in terms of primitive rather than conservative variables since the calculation of the linearisation matrices proves more efficient with respect to $\mathbf{P}$ than $\mathbf{W}$. For the axisymmetric case, there are extra terms on the right-hand side, see equation (2.33). The axisymmetric inviscid part is treated implicitly, but the viscous part is treated explicitly. Numerical experiments have shown that it is necessary to have an implicit treatment for the axisymmetric inviscid terms if a tight restriction on the allowable time step is to be avoided. The explicit treatment of the axisymmetric viscous terms does not have a deleterious effect on stability or limit the allowable time step, on comparison with the original planar code, so an implicit treatment was not attempted. See Section 3.2.4 for an example of the importance of the implicit treatment for the axisymmetric inviscid terms. The modified linear system for the axisymmetric case is then written as :

$$\left(\frac{I}{\Delta t}\frac{\partial \mathbf{W}}{\partial \mathbf{P}} + \frac{\partial\,(\mathbf{R} - \mathbf{H}^i)}{\partial \mathbf{P}}\right)\Delta \mathbf{P} \;=\; -\mathbf{R}^{(n)} + \mathbf{H}^{i(n)} + \mathbf{H}^{v(n)} \tag{2.38}$$

where $\mathbf{H}^i$ and $\mathbf{H}^v$ are the inviscid and viscous parts respectively of the discretised source term. System (2.37) is solved using an identical scheme [31] as used for (2.38). The inviscid source term Jacobian is evaluated as

$$\frac{\partial \mathbf{H}^i}{\partial \mathbf{P}} = -\frac{1}{r}\begin{bmatrix} v_r & 0 & \rho & 0 \\ v_r v_z & \rho v_r & \rho v_z & 0 \\ v_r^2 & 0 & 2\rho v_r & 0 \\ v_r \frac{|\vec{V}|^2}{2} & \rho v_r v_z & \frac{\gamma}{\gamma-1}p + \rho\frac{|\vec{V}|^2}{2} + \rho v_r^2 & \frac{\gamma}{\gamma-1}v_r \end{bmatrix} \tag{2.39}$$

where $|\vec{V}|^2 = v_r^2 + v_z^2$.

2.3.3 Implicit scheme, $k-\omega$ equations

The equations forming the turbulence model are solved in essentially the same manner as the mean flow equations. The linear system arising at each implicit time step for the 2-D planar formulation can be summarised as

$$\left(\frac{I}{\Delta t}\frac{\partial \mathbf{W}_T}{\partial \mathbf{P}_T} + \frac{\partial\left(\mathbf{R}_T - \mathbf{H}_T\right)}{\partial \mathbf{P}_T}\right)\Delta \mathbf{P}_T = -\mathbf{R}_T^{(n)} + \mathbf{H}_T^{(n)}$$

where $\mathbf{P}_T = 1/J\left(k,\omega\right)^T$ is the vector of cell based primitive variables, $\mathbf{W}_T = 1/J\left(\rho k, \rho\omega\right)^T$ is the vector of cell based conservative variables, $\Delta\mathbf{P}_T$ is the update in $\mathbf{P}_T$ from time level n to n+1 and $\mathbf{R}_T^{(n)}$ and $\mathbf{H}_T^{(n)}$ are the flux and source term residuals arising from the spatial discretisation respectively. For the axisymmetric case, there are extra terms on the right-hand side, see equation (2.34). The 'inviscid' parts of the additional source term are treated implicitly. The modified linear system for the axisymmetric case is then written as :

$$\left(\frac{I}{\Delta t}\frac{\partial \mathbf{W}_T}{\partial \mathbf{P}_T} + \frac{\partial\left(\mathbf{R}_T - \mathbf{H}_T - \mathbf{H}_T^a\right)}{\partial \mathbf{P}_T}\right)\Delta \mathbf{P}_T = -\mathbf{R}_T^{(n)} + \mathbf{H}_T^{(n)} + \mathbf{H}_T^{a(n)}$$

where $\mathbf{H}_T^{a(n)}$ are the additional source term elements of the axisymmetric formulation. Its Jacobian is written as (discarding viscous terms)

$$\frac{\partial \mathbf{H}_T^a}{\partial \mathbf{P}_T} = -\frac{1}{r}\begin{bmatrix} \frac{5}{3}\rho v_r & 0 \\ 0 & \left(1+\frac{2}{3}\alpha\right)\rho v_r \end{bmatrix} \tag{2.40}$$

2.4 Laminar Poiseuille flow

2.4.1 Purpose of test case

An analytic solution of the Navier-Stokes equations exists for the case of laminar, incompressible, fully developed flow through a straight pipe of constant circular cross-section. A simulation of this type of flow using the laminar, axisymmetric version of PMB2D therefore provides a useful check on the formulation.

2.4.2 Description of test case

Fully developed flow in a pipe is characterised by a zero pressure gradient across the pipe, a constant pressure gradient along the pipe and a velocity profile which is invariant along the pipe. This situation arises because the pressure forces which drive the flow are exactly balanced by shear forces such that no acceleration can occur. For fully developed, steady, incompressible, laminar flow through a pipe of radius r_o^* (axisymmetric Poiseuille flow) the analytic solution for the velocity components is written as [52] :

$$\begin{aligned} v_r^* &= 0 \\ v_z^* &= \frac{1}{4\mu^*}\frac{dp^*}{dz^*}\left(r^{*2} - r_o^{*2}\right) \end{aligned}$$

where r and z are the radial and axial directions respectively. The superscript $(^*)$ denotes dimensional quantities. The flow solver uses non-dimensional quantities, so it is more convenient to use this expression in the form

$$\begin{aligned} v_r &= 0 \\ v_z &= \frac{Re_l}{4\mu}\frac{dp}{dz}\left(r^2 - r_o^2\right) \end{aligned} \tag{2.41}$$

where

$$r = \frac{r^*}{l^*}, \qquad z = \frac{z^*}{l^*}, \qquad v_r = \frac{v_r^*}{V_\infty^*}, \qquad v_z = \frac{v_z^*}{V_\infty^*},$$

$$p = \frac{p^*}{\rho^*_\infty V^{*2}_\infty}, \qquad \mu = \frac{\mu^*}{\mu^*_\infty}, \qquad Re_l = \frac{\rho^*_\infty V^*_\infty l^*}{\mu^*_\infty}$$

and l^* is a characteristic length, for example the overall length of the pipe. Here the reference conditions are taken as the conditions at the centre-line of the inlet. A subscript ∞ is retained here to denote such conditions in order to follow the convention used in section A.2 . The Mach number and Reynolds number of the flow considered correspond to low speed laminar flow: $M_\infty = 0.01$ and $Re_l = 500$.

2.4.3 Grid generation

The grid generation for this test case is straightforward. Two single block grids were used. Details of the grid dimensions and spacings are summarised in Table 2.1. The grids used are shown in Figures 2.3 and 2.4. The flow is in the direction of increasing z. The grids are refined slightly towards the wall because of the higher viscous stresses expected in this area.

Name	*Dimensions*	*Grid spacing at wall*
Grid A	15 x 25	0.010
Grid B	31 x 51	0.005

Table 2.1: *Grids used for Poiseuille flow test case*

2.4.4 Boundary and initial conditions

At the outlet, the pressure is imposed at a value of $p = 1.0$ and the density and velocity components are extrapolated from the interior. At the inlet, the velocity is imposed using the analytic expression (2.41) normalised to unity at the centreline. The density is imposed at $\rho = 1.0$, the flow being incompressible, and the pressure is extrapolated from the interior. The walls are modelled as being adiabatic with no slip; the velocity components are set to zero and the pressure and density are extrapolated from the interior. The following initial conditions were used throughout the domain: $\rho = 1.0$, $v_r = 0.0$, $v_z = 1.0$, $p = 1.0$.

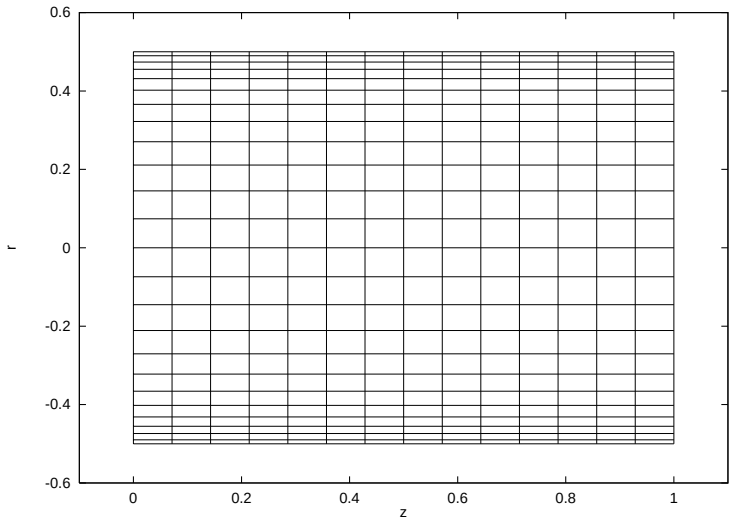


Figure 2.3: *Grid A used for Poiseuille flow test case*

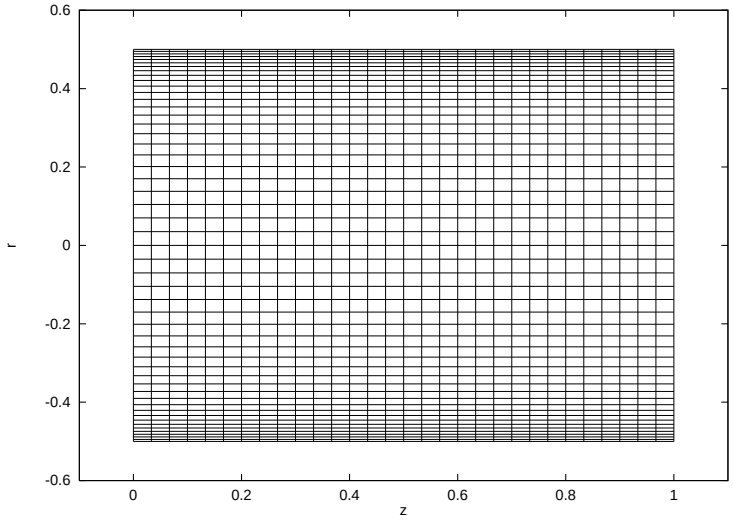


Figure 2.4: *Grid B used for Poiseuille flow test case*

2.4.5 Results

Solutions were obtained successfully using both grids. The convergence criterion used was the reduction by eight orders of magnitude of the L2 norm of the residual. The rate of convergence was slow in both cases, taking around 8000 work units in total[1]. This was expected when using a compressible flow solver for such a low speed flow, but is unimportant here where we are interested solely in the accuracy of the solution. The solutions obtained with the coarser grid A are identical to those obtained with grid B therefore the solutions can be considered grid converged. The pressure coefficient at every cell centre is plotted in Figure 2.5 for the calculations on both grids. This clearly shows features which correspond with the analytic solution: there is a constant pressure gradient in the axial direction and no radial pressure variation. Figures 2.6 and 2.7 show the calculated velocity profile for grids A and B respectively. Both are compared with the exact solution for the calculated pressure gradient. There is excellent agreement between the theory and the calculation. The computed profiles shown were taken from central sections; any section could have been used because the profile does not change along the pipe.

Here we are concerned with axisymmetric flow. The analytic solution for planar Poiseuille flow[52] is similar but the maximum velocity is twice the magnitude of the axisymmetric case for the same axial pressure gradient. Planar Poiseuille flow has also been calculated using PMB2D, see [55]. The same approach was used as above and again very good agreement with theory was obtained. This underlines the important role played by the 'additional' viscous terms (section 2.2.7) in an axisymmetric formulation.

2.5 Conclusions

In this chapter the adaption of a two-dimensional flow solver to axisymmetric flows has been described. The equations for axisymmetric flow have been presented in full. It has been demonstrated how the axisymmetric flow equations can be cast in a form very similar to that of the two-dimensional equations. The equations can

[1] 1 work unit corresponds to the CPU time for 1 explicit time step

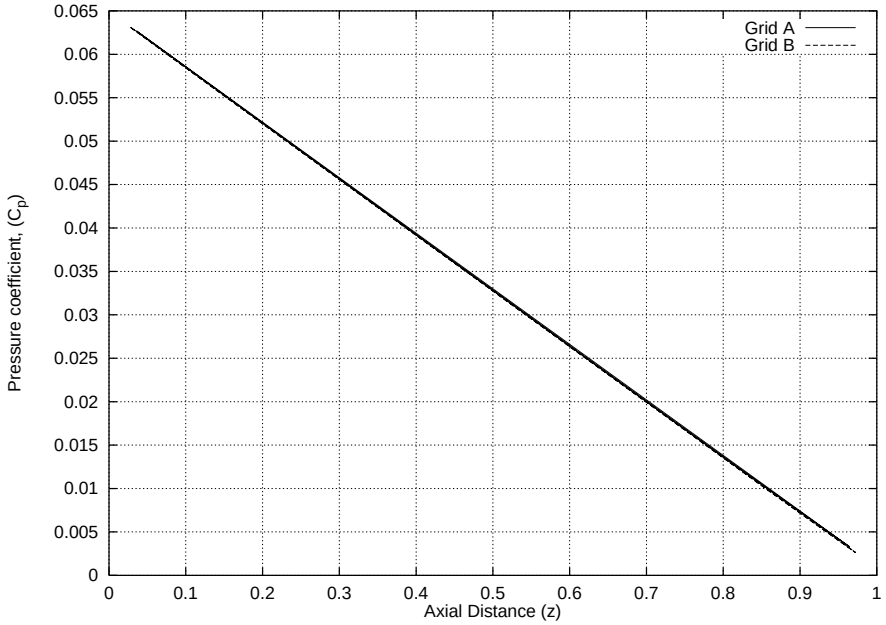


Figure 2.5: *Pressure coefficient for Poiseuille flow test case*

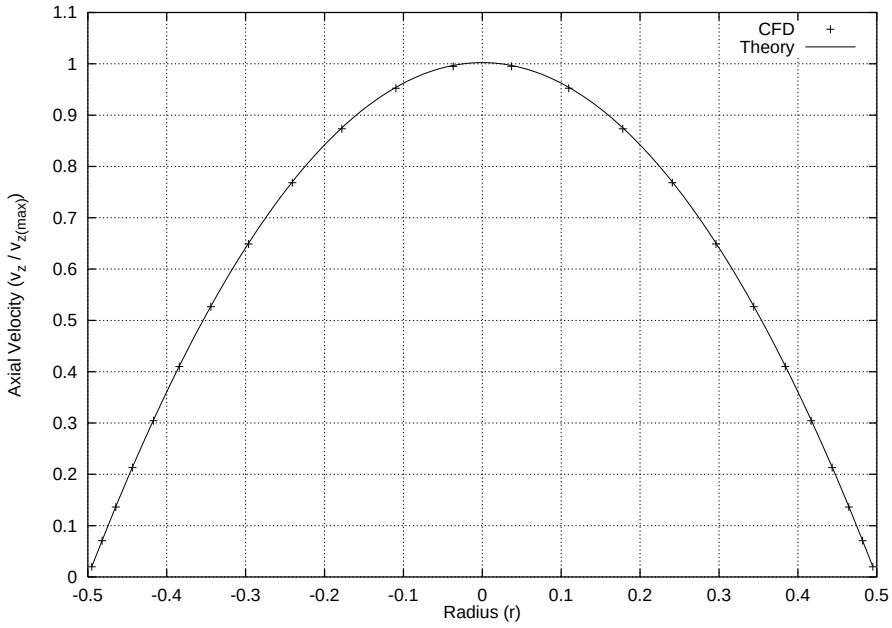


Figure 2.6: *Computed and theoretical velocity profiles for Grid A, Poiseuille flow test case*

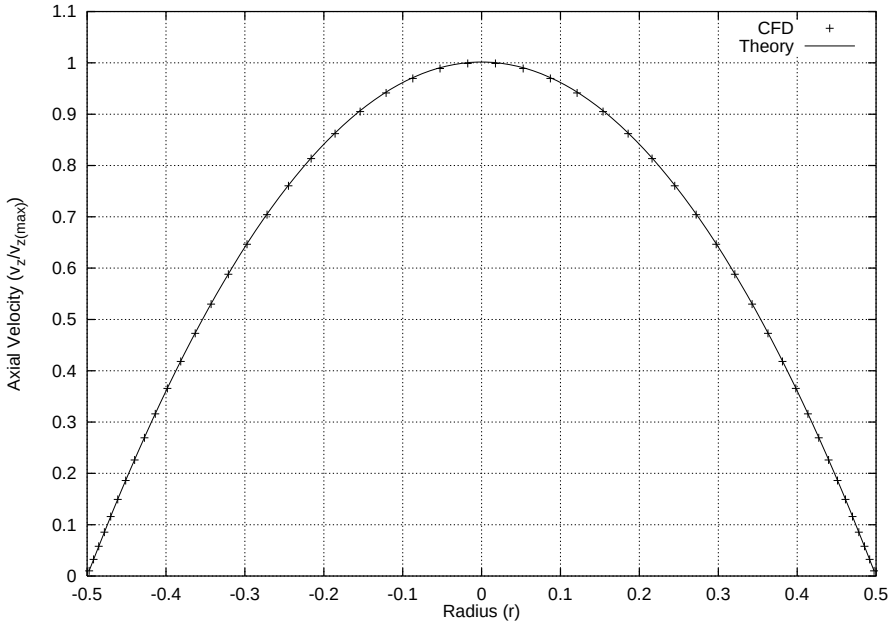


Figure 2.7: *Computed and theoretical velocity profiles for Grid B, Poiseuille flow test case*

then be solved using essentially the same numerical scheme, the only alteration being the introduction of source terms to account for axisymmetry. The introduction of the discretised source terms into the implicit scheme is also presented, the inviscid parts are treated implicitly and the viscous parts explicitly. The simple test case of laminar Poiseulle flow through a pipe was examined. Excellent agreement between theory and computational results was obtained. The accuracy of the results establishes confidence in the axisymmetric viscous treatment. In Chapters 3 and 4 the axisymmetric flow solver is applied to engineering and scientific problems. In each case comparison is made between experimental and computational results. Section 3.4 includes an example of the performance advantage obtained by using an axisymmetric solver over a fully three-dimensional method.

Chapter 3

Engineering Evaluation: Forebody and Base Flows

3.1 Introduction

The engineer's choice of aerodynamic analysis method has always been a trade-off between the cost of implementing the method and the accuracy of the results obtained. When faced with the task of evaluating multiple configurations the engineer would like to base conclusions on the results of exhaustive wind tunnel testing, but must usually employ a less expensive method. The purpose of this chapter is to evaluate the axisymmetric flow solver described in Chapter 2 as a tool for the aerodynamic analysis of engineering problems. Two classes of flow problem are considered, axisymmetric forebody and base flow. For these problems it is likely that a large number of configurations over a wide range of flow conditions would be considered in an evaluation study, so the performance gains of an axisymmetric over a three-dimensional flow solver are important. A number of test cases for which experimental data is available will be examined. Accuracy, robustness, speed and pre- and post-processing effort required will be assessed, with emphasis on the potential for routine calculation.

3.2 ONERA B1 and B2 ogive cylinders

3.2.1 Description of test cases

The ONERA B1 and B2 test case configurations appear frequently in the literature as benchmark test cases for slender-body supersonic flow, see for example [56]. Data from the original wind tunnel tests and from other computations are available for comparison. These are therefore useful test cases for code validation.

ONERA B1

The ONERA B1 configuration consists of a pointed convex forebody continued tangentially by a circular cylinder of diameter D. The forebody is of length 3D and is described by the arc of a circle of radius 9.25D. The test conditions reported from the original experiment are as follows:

$$
\begin{aligned}
&\text{Laminar flow} \\
\text{Freestream Mach number, } M_\infty &= 2.0 \\
\text{Reynolds number, } Re_D &= 0.16 * 10^6 \\
\text{Freestream stagnation pressure, } p_{t\infty} &= 50 * 10^3 Pa \\
\text{Freestream stagnation temperature, } T_{t\infty} &= 330K \\
\text{Wall temperature, } T_w &\simeq 315K\,(adiabatic) \\
\text{Incidence, } \alpha &= 0^o
\end{aligned}
$$

ONERA B2

The ONERA B2 geometry is very similar to that of the B1. The convex forebody is described by a parabolic profile, equation (3.1) rather than a circular arc. Again the forebody is of length 3D. The test conditions reported from the original experiment

are as follows:

$$\begin{aligned}
&\text{Turbulent flow} \\
\text{Fixed transition at } z/D &= 0.15 \\
\text{Freestream Mach number, } M_\infty &= 2.0 \\
\text{Reynolds number, } Re_D &= 1.2 * 10^6 \\
\text{Freestream stagnation pressure, } p_{t\infty} &= 120 * 10^3 Pa \\
\text{Freestream stagnation temperature, } T_{t\infty} &= 300K \\
\text{Wall temperature, } T_w &\simeq 285K\ (adiabatic) \\
\text{Incidence, } \alpha &= 0^o
\end{aligned}$$

$$\frac{r}{D} = \frac{1}{3}\left(\frac{z}{D}\right) - \frac{1}{18}\left(\frac{z}{D}\right)^2 \tag{3.1}$$

3.2.2 Grid generation

The grids used in this study were standard grids supplied by ONERA as part of a GARTEUR workshop. Two grids were supplied for each case, the coarser intended for inviscid (Euler) calculations and the finer for viscous calculations. Details of the grids are summarised in Table 3.1. Grid B1c, the coarser grid for the B1 case, is shown in Figures 3.1 and 3.2. The other grids are very similar. All of the grids include a small nose boom, one cell in width, of very small but finite radius ($1.0\text{x}10^{-7}$D). This feature was intended to aid contributors to the workshop using three-dimensional flow solvers which would not handle the singularity at the nose. It was not needed here, but was retained since experiments using a modified grid with the nose boom removed showed that it has no effect on the solution.

Name	*Dimensions*	*Grid spacing on cylinder surface*
B1c	61 x 53	$1.74 * 10^{-2}$ D
B1f	61 x 85	$2.00 * 10^{-4}$ D
B2c	61 x 53	$1.74 * 10^{-2}$ D
B2f	61 x 85	$2.50 * 10^{-5}$ D

Table 3.1: *Grids used for ONERA B1 and B2 test cases*

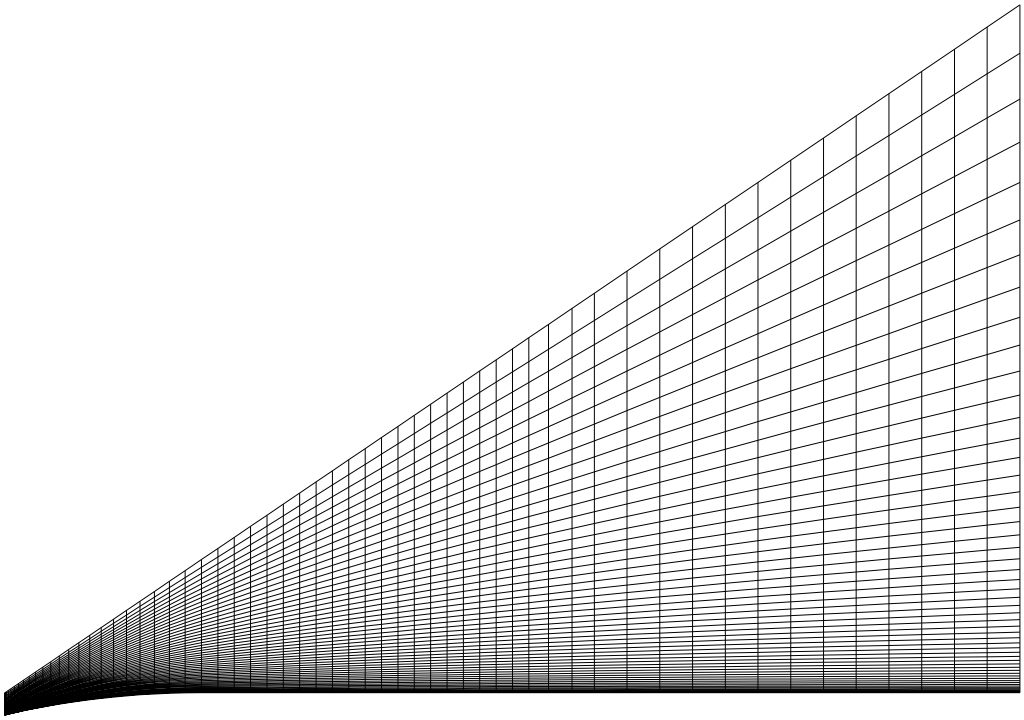


Figure 3.1: *Euler grid, ONERA B1 test case*

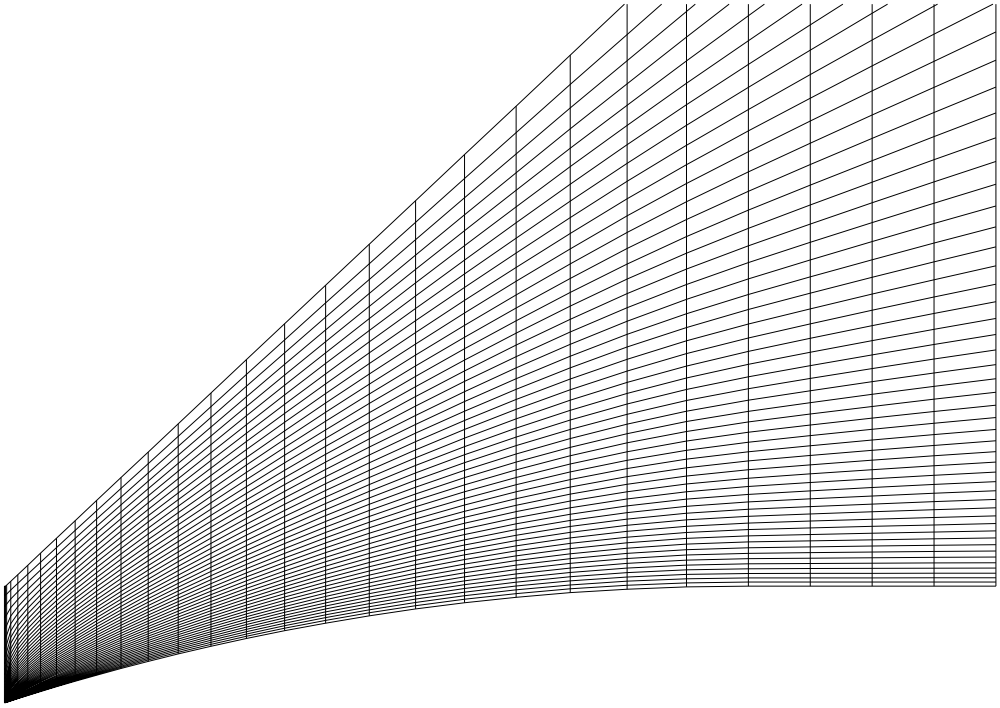


Figure 3.2: *Nose region detail of Euler grid, ONERA B1 test case*

3.2.3 Boundary and initial conditions

All variables were extrapolated from the interior across the outflow boundary. The wall boundary was modelled as being adiabatic with no slip. A characteristic-based far-field boundary condition was employed at the remaining two domain boundaries.

3.2.4 Results

Solutions were obtained for all four cases: inviscid (Euler) calculations for B1 and B2, laminar Navier-Stokes for B1 and Reynolds-averaged Navier-Stokes with $k-\omega$ turbulence model for B2. A summary of the calculations performed is shown in Table 3.2. Included in this table are the CPU times for each calculation on a Silicon Graphics Indy R5000. In each case, the calculation was considered converged when the L2 norm of the residual had reduced by eight orders of magnitude. Convergence histories for each case are shown in Figures 3.3 to 3.6. The B1 Euler calculation was also attempted using an explicit treatment for the axisymmetric inviscid terms to examine the effect of the implicit treatment, see Section 2.3.2. In order to obtain a solution it was necessary to use twice as many explicit steps before switching to the implicit scheme, and the implicit CFL number was limited to 50, rather than a value of 250 used in the calculation shown. As a result the overall time taken for the calculation was increased by 50%. This supports the present method where the implicit treatment is used. The B1 Euler case has also been examined using a fully three-dimensional version of the present method. The calculation takes approximately 100 times as long and requires 50 times as much memory. The solutions are identical. This clearly demonstrates the utility of an axisymmetric flow solver.

Calculation	*Grid used*	*CPU time*
B1, Euler	B1c	50 s
B2, Euler	B2c	47 s
B1, Laminar	B1f	288 s
B2, Turbulent	B2f	822 s

Table 3.2: *Summary of calculations for ONERA B1 and B2*

Figures 3.7 to 3.12 show the calculated values of pressure coefficient, skin friction coefficient and local axial force. Comparison is made with experimental data [57] where possible and otherwise with other computations [58]. Table 3.3 shows a summary of the calculated total axial force coefficients: Ca_p denotes the pressure component, Ca_f the viscous component and Ca is the total.

Calculation	Ca_p	Ca_f	Ca
B1, Euler	0.0953	-	0.0953
B2, Euler	0.0947	-	0.0947
B1, Laminar	0.0985	0.0511	0.1496
B2, Turbulent	0.0982	0.1310	0.2292

Table 3.3: *Summary of calculated axial force coefficients*

Good agreement was obtained with the experimental values of pressure coefficient for the B1 case, see Figure 3.7. The calculated skin friction coefficient curve, see Figure 3.8, agrees well with the ONERA computational results over the forebody. However, the two curves begin to diverge downstream, and at $z/D = 15$ the ONERA computation predicts nearly twice as much skin friction. The calculated local contribution to the axial force for the B1 case, Figure 3.9, shows up the same differences between the results i.e. a good match for the pressure component and a poor match for the viscous component. The axial force coefficient values quoted in Table 3.3 are calculated as the area underneath the local axial force curves. The good agreement of the pressure values with experiment shown (and with ONERA pressure results not shown) is encouraging from the point of view of verification of the flow solver. It is not possible at present to say much about the skin friction results since we only have the results from two computations, although the difference in results is disappointing.

The calculated pressure coefficient for the B2 case matches the experimental values very well over the forebody, but over the remainder of the surface the computational results seem to be offset slightly, see Figure 3.10. Comparison with the ONERA results for Cp values is not shown, but the agreement is very good. The calculated skin friction coefficient curves for the present calculation and from ONERA

are in fairly good agreement, see Figure 3.11. Note that the $k-\omega$ turbulence model was used for the present calculation, and ONERA used a Baldwin-Lomax turbulence model. Comparing the local contribution to the axial force for the B2 case with the B1 case, Figures 3.9 and 3.12, for the B2 case the effect of viscous drag appears to be relatively more important. This is a trend that we expect since the B2 case is turbulent with a higher Reynolds number.

3.2.5 Numerical implementation of the turbulence model

In the present method, a number of explicit (backwards Euler) iterations are performed before switching to the implicit scheme with a high, constant CFL number (say 250). Experience has shown that this is an effective way of initiating the calculation. During this explicit stage in the turbulent B2 calculation the scheme became unstable. This seemed to be caused by the appearance of small and negative values of k and ω. Other workers have also experienced such difficulties in the initial stages of a calculation when using two- and one-equation turbulence models [36], [59], [60], [61]. The various remedies reported apply specifically to implicit schemes. Here the problem arises during the explicit stage, and the straightforward remedy of limiting the values of k and ω to be no less than the freestream values was applied. These limits were only used during the explicit stage. Figure 3.5 shows a convergence plot of the calculation. It is noted that the number of explicit iterations required is relatively large and that the residual for the turbulent quantities is small in the initial stages. An explicit CFL number of 0.4 was used here for both the mean flow and the turbulence equations. In an attempt to speed up the calculation by making the turbulent quantities do more work, the calculation was re-run using an explicit CFL number of 0.4 for the mean flow equations and 0.6 for the turbulence equations. Figure 3.6 shows a convergence plot of the calculation. In this case less explicit steps were required and the overall CPU time for the calculation was reduced by nearly 20%.

When using an implicit scheme and a two- or one-equation turbulence model, the treatment of the source term Jacobian arising from the time linearisation of the updates for the turbulent quantities is reported to be important for stability, partic-

ularly during the initial stages of the calculation[36],[54],[59],[60],[61]. As discussed above, in our case the initial instability problem is dealt with during the explicit stage. The effect of the suggested modified implicit schemes was investigated in any case for the B2 problem. The modified schemes all involve some variation of the turbulent source term Jacobian in the form of neglecting off-diagonal terms, varying the size of coefficients or altering the terms in the matrix according to sign changes. The modified schemes showed no improvement, either regarding robustness (the number of explicit steps required was unchanged) or speed of convergence.

3.2.6 Conclusions

The axisymmetric version of PMB2D has been successfully applied to two supersonic slender-body aerodynamics problems. The results have been compared with experimental data and computational data from other sources. The agreement with other data is good. Together with other successful applications of the code to this type of flow [62], this gives confidence in the accuracy of the code for this type of problem. Some useful insights into the numerical implementation of the $k-\omega$ turbulence model have also been gained. An implicit treatment of the inviscid part of the axisymmetric source term allows larger time steps to be used than an explicit treatment, and hence reduces run-time. The axisymmetric flow solver has been demonstrated to be significantly faster than a fully three-dimensional method, and also requires much less memory.

3.3 GARTEUR Base Flow

3.3.1 Description of test case

The aerodynamics of the base region strongly influences the drag of a projectile. However, reliable prediction of base flow for the wide range of possible conditions (and geometric configurations) that a designer may wish to examine has still to be attained. Semi-empirical and multi-component methods are very useful in this field but the time-averaged Navier-Stokes approach is the most credible and promising[63],[64]. However, despite the apparent suitability of a Navier-Stokes ap-

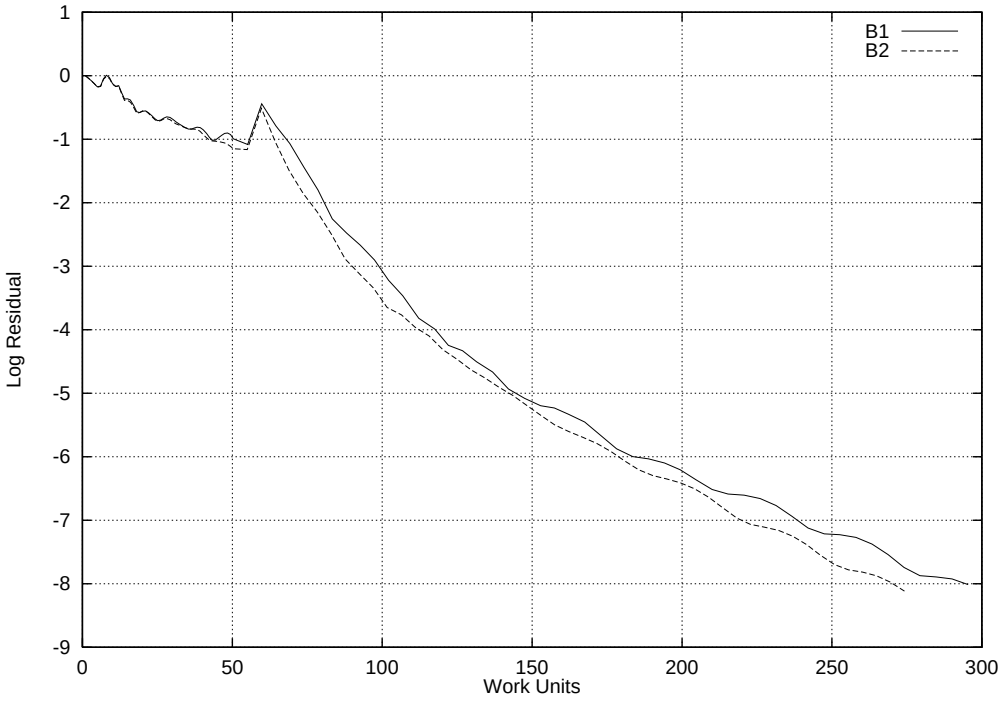


Figure 3.3: *Convergence histories for B1 and B2 Euler calculations*

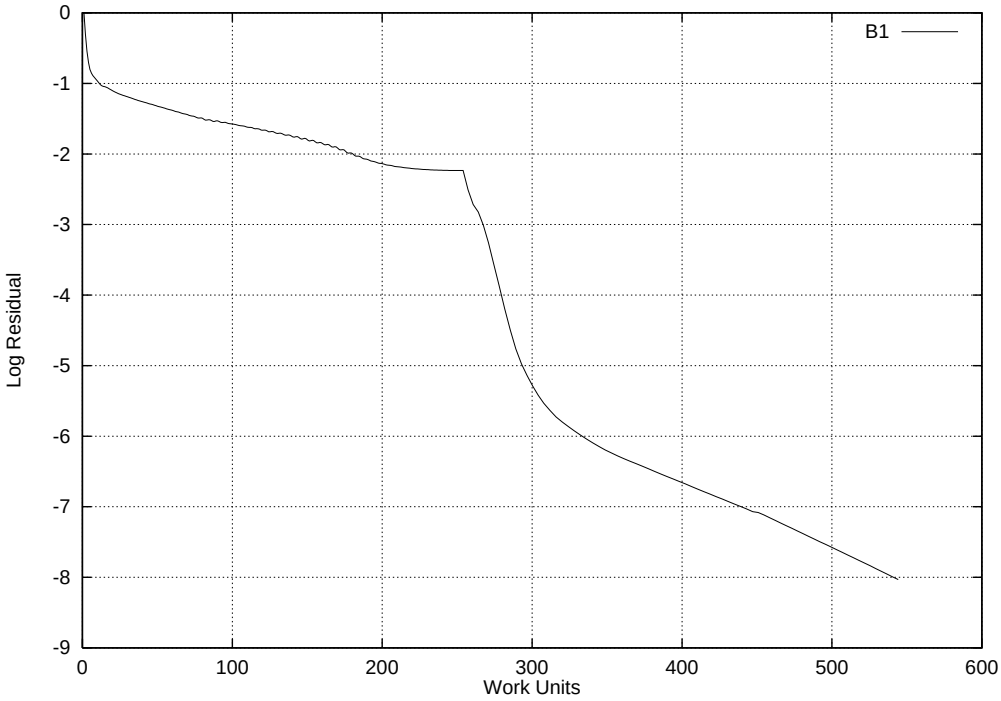


Figure 3.4: *Convergence histories for B1 laminar Navier-Stokes calculation*

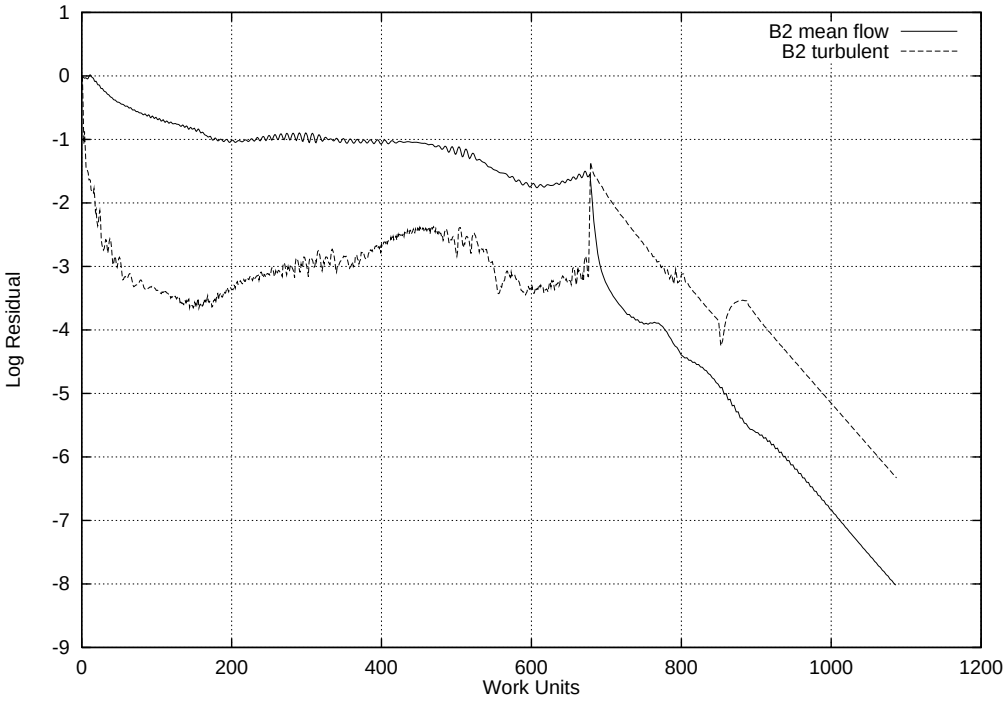


Figure 3.5: *Convergence history for B2 turbulent Navier-Stokes calculation (explicit CFL=0.4)*

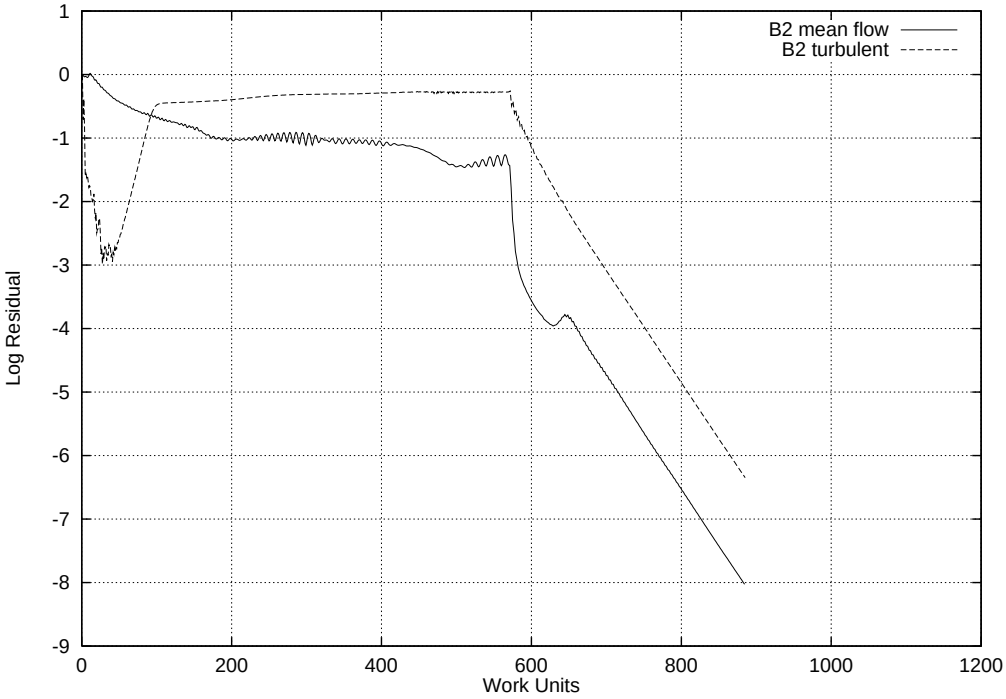


Figure 3.6: *Convergence history for B2 turbulent Navier-Stokes calculation(explicit CFL=0.6)*

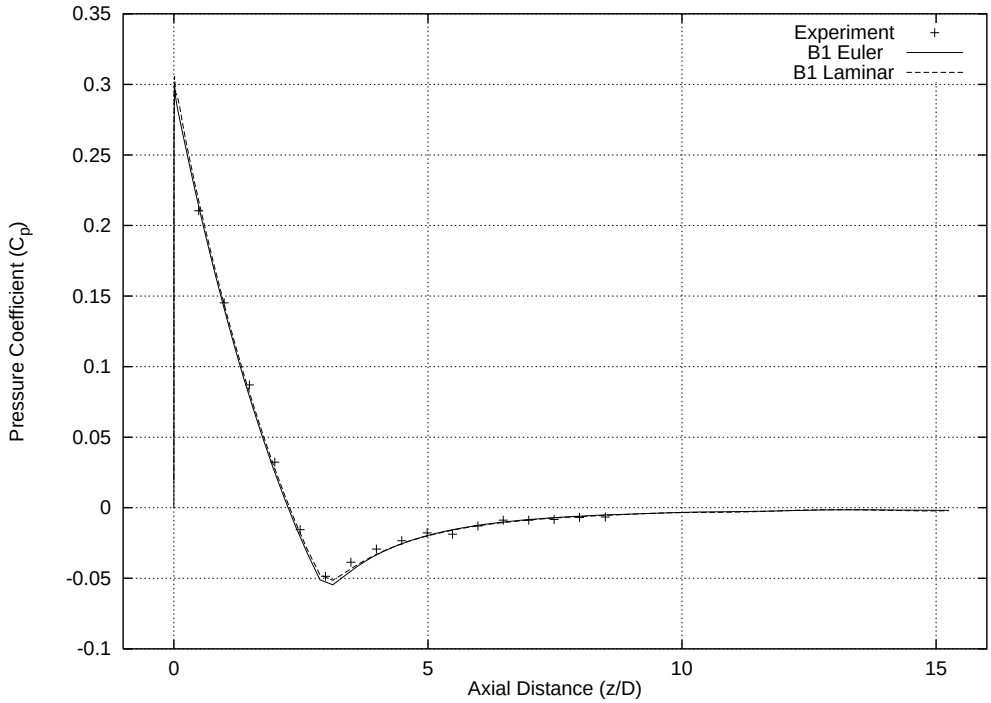


Figure 3.7: *Experimental and calculated pressure coefficient, ONERA B1*

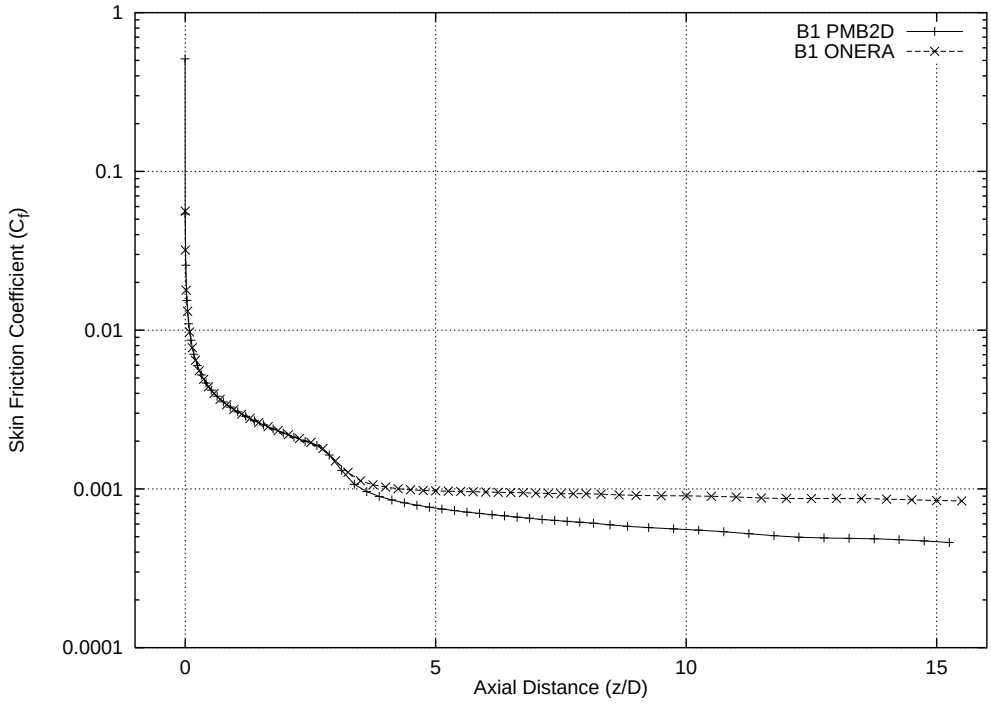


Figure 3.8: *Calculated skin friction coefficient, ONERA B1. (Comparison with ONERA results)*

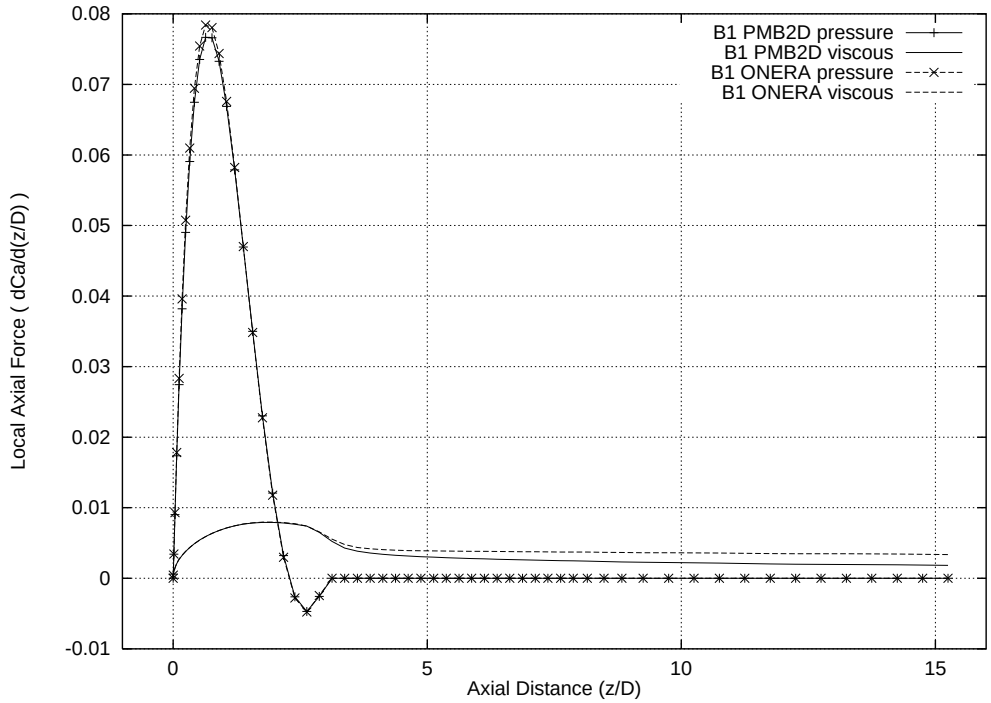


Figure 3.9: *Calculated local axial force coefficient, ONERA B1. (Comparison with ONERA results)*

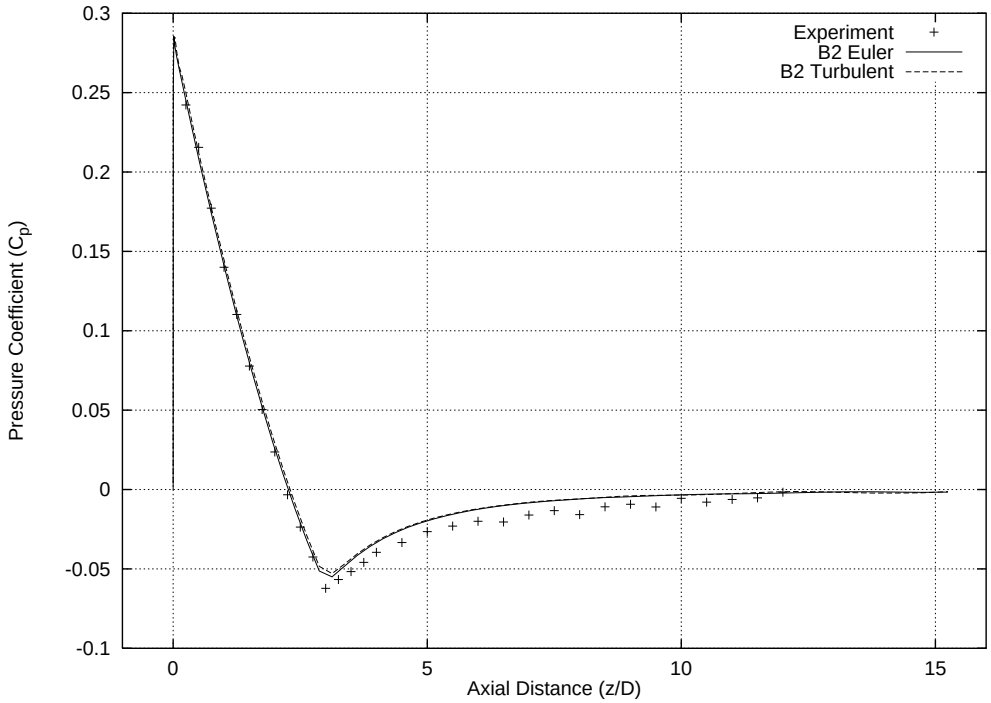


Figure 3.10: *Experimental and calculated pressure coefficient, ONERA B2*

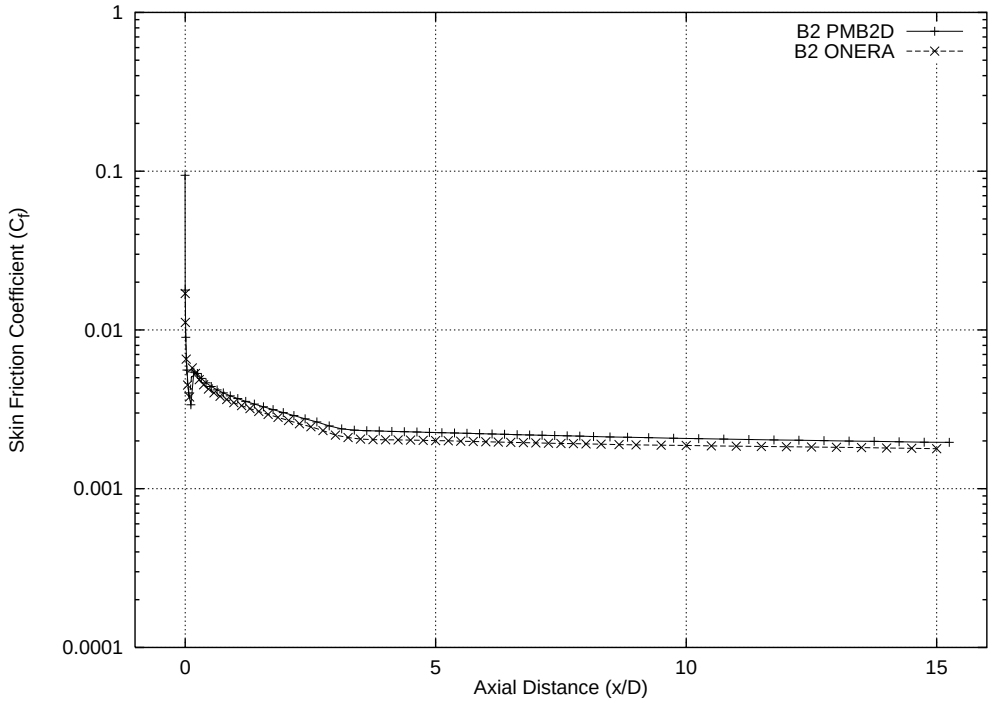


Figure 3.11: *Calculated skin friction coefficient, ONERA B2. (Comparison with ONERA results)*

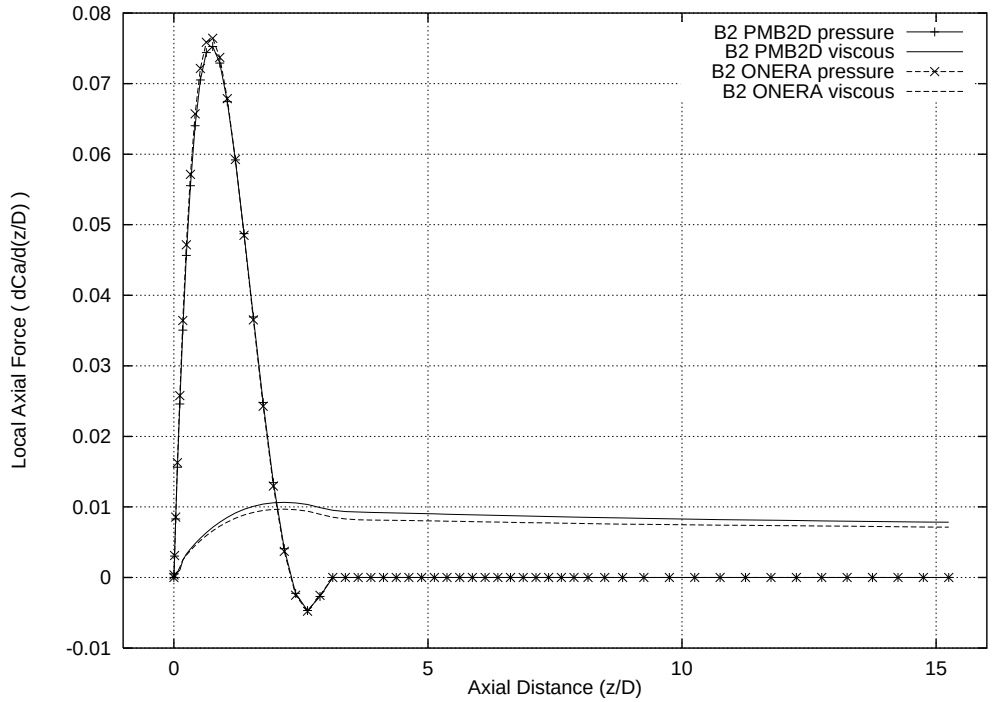


Figure 3.12: *Calculated local axial force coefficient, ONERA B2. (Comparison with ONERA results)*

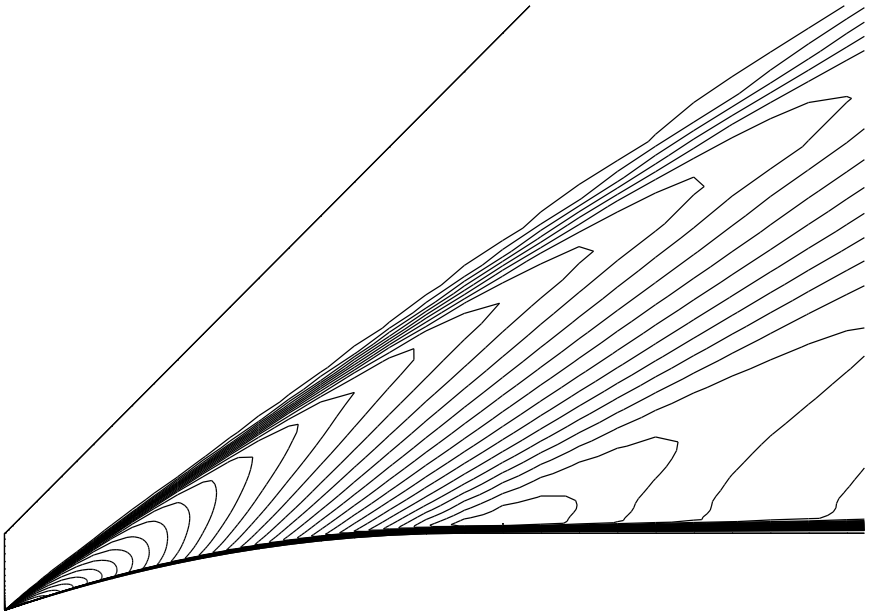


Figure 3.13: *Mach number contours for ONERA B2 test case*

proach to this type of aerodynamically complex problem results of studies to date have not always been quantitatively satisfactory[63]. Recent studies[36],[37],[65],[66], [67], have indicated the importance of grid generation and turbulence modelling . In particular, the algebraic Baldwin-Lomax turbulence model is dismissed as wholly inappropriate for base flows and the results for $k-\epsilon$ models and variations are better although inconsistent. Some improvement is reported through the use of more sophisticated turbulence models[36]. The present study aims to evaluate the ability of the present method, which uses a $k-\omega$ turbulence model, to provide accurate base flow predictions by examining a test case particularly designed for Navier-Stokes flow solver validation. At the same time the robustness of the present method, the effort required by the engineer in its application and the overall calculation time are kept in mind since accuracy is not the only consideration of the designer operating in a commercial/industrial environment.

AFTERBODY TEST CASE 1B: CONICAL BOAT-TAIL

The afterbody geometry consists of a short cylindrical section followed by a conical

boat-tail at 6^o to the cylinder's surface and one cylinder diameter D in length. The geometry and test conditions are described in detail in [68]. The flow conditions are summarised as follows

Fully turbulent flow

$$\begin{aligned} \text{Freestream Mach number, } M_\infty &= 0.35 \\ \text{Reynolds number, } Re_D &= 1.54 * 10^5 \\ \text{Freestream stagnation pressure, } p_{t\infty} &= 10^5 Pa \\ \text{Freestream stagnation temperature, } T_{t\infty} &= 330K \\ \text{Incidence, } \alpha &= 0^o \end{aligned}$$

3.3.2 Grid generation

The grids used for previous numerical studies of this test case[64] vary widely in fineness, topology, stretching and far-field boundary extent. In addition, grid convergence checks were absent from these studies. In the present work, the far-field boundary extent was set at the largest values used in the previous studies (15 diameters downstream of the base and 5 diameters normal to the axis of symmetry). In order to determine the number of grid points to use, the number of points in each direction from the coarser grids in [64] was noted, and a grid with four times as many points in each direction was constructed. Successively coarser grids were then obtained by extracting points from this very fine grid, see Table 3.4. This hierarchy of grids formed the basis of the grid convergence study, see Section 3.3.4. The finest grid used here has more than twice as many points as any used in the previous studies. Figure 3.14 shows the coarse grid.

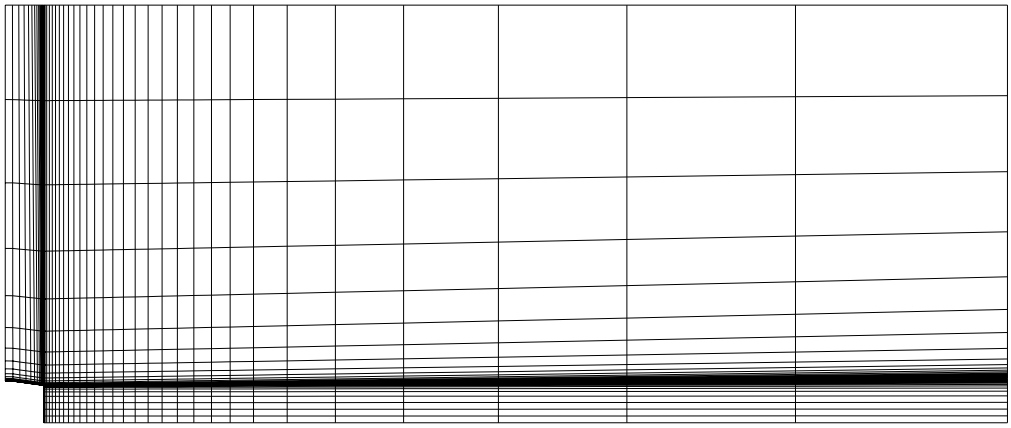


Figure 3.14: *Coarse grid used for GARTEUR afterbody 1B*

Number of points	Grids used : a(*very fine*)	b(*fine*)	c(*medium*)	d(*coarse*)
Along base	121	61	31	16
Along boat-tail	161	81	41	21
Along symmetric line	281	141	71	36
Normal to symmetric line	281	141	71	36
Total	105163	26583	6793	1773

Table 3.4: *Summary of grid dimensions*

3.3.3 Boundary and initial conditions

The boundary layer thickness at the inflow boundary is included in the report of the experimental results[68]. In order to obtain values to impose at the inflow boundary for the main calculation, a short preliminary calculation was performed using the same conditions on a cylindrical body to simulate the flow upstream of the afterbody. At the axial position where the calculated boundary layer has grown to

the reported thickness the values were extracted and used for the inflow condition of the main calculation. All of the flow variables are imposed except the pressure which is extrapolated from the interior since the flow is subsonic. A description of how the inflow boundary condition was tackled in the previous calculations was not included in their respective reports. The conditions at the remaining boundaries are more straightforward. The wall boundary was modelled as being adiabatic with no slip. Symmetry was imposed along the axis of symmetry and a characteristic based far-field boundary condition was employed at the remaining two domain boundaries.

The calculation was initiated from freestream conditions in order to obtain the coarse grid solution. This solution was used as the initial condition for the subsequent medium grid solution and so on. In this way the calculations on the finer grids were initiated from already 'good' conditions thus reducing overall run times.

3.3.4 Results

Results were obtained on the coarse and medium grids without any problem. However, on the two finer grids it was not possible to obtain a solution without altering the turbulence model implementation in an attempt to circumvent an instability problem. The solution would proceed apparently normally before becoming unstable in the vicinity of the free stagnation point and crashing. The initial manifestation of this instability is a sharp increase in the calculated turbulent kinetic energy production term $\hat{\mathbf{P}}_k$ (see Section B.2). A variety of alternative turbulent source term Jacobian matrices, see Section 3.2.5, were implemented in an attempt to improve stability with no success. In order to obtain a solution, the ratio of production to dissipation $\hat{\mathbf{P}}_k/\hat{\mathbf{D}}_k$ was limited. Using the fine and very fine grids the maximum value of this ratio resulting in a stable solution were 1.7 and 1.6 respectively. Note that for the coarser grid calculations (and for calculations on the finer grids employing first-order convective accuracy) this ratio could reach 4.0 in the converged solutions. Figure 3.26 shows a contour plot of this ratio for the solution on the medium grid. The highest values occur at the beginning of the boat-tail on the cylinder, in the free shear layer and in the recirculation region. Imposing a limit on this ratio forces a reduction on the amount of turbulent kinetic energy in the

flow and aids stability in the vicinity of the free stagnation point. Note that at the free stagnation point the ratio becomes negative. In addition, a stable solution was also obtained on the fine grid by 'freezing' the turbulent quantities at their values 20 iterations before the failure and continuing to update the mean flow quantities normally. The justification for this is that before the solution becomes unstable the residuals for the mean flow and turbulent quantities have already decreased by more than three orders of magnitude, the calculation having been initiated from the medium grid solution, so the turbulence field should be a reasonable approximation to the 'real' solution. At the least a solution obtained in this way provides a useful comparison with the solution obtained by using a limit as described above.

Figure 3.15 shows the calculated pressure coefficient distribution along the symmetric line using all four grid levels. The results shown for the fine and very fine grids are those obtained with the production-dissipation limit described above. Figure 3.16 shows how the calculated pressure coefficient distribution along the symmetric line for the 'frozen turbulence' and 'limit' calculations on the fine grid differ slightly. From these figures it is clear that a grid converged solution has not been obtained. It is not possible to blame the differences between the fine and very fine grids solely on the uncertainty caused by the limit used in the calculation. To help indicate whether the grid hierarchy should be sufficient to obtain grid independent results, laminar calculations were also performed. The calculated pressure coefficient distributions along the symmetric line are shown in Figure 3.17. These are also not grid converged. The calculated pressure coefficient along the base compared with experimental data is shown in Figure 3.18. These results again indicate that grid independence has not been achieved and also show poor agreement with experiment. The present pressure coefficient results are similar to the numerical results presented in [64] regarding the location of the maximum and minimum pressures on the symmetric line and generally poor prediction of the base pressure. The present study has strongly indicated the necessity of performing a grid independence study, raising considerable doubt over the validity of computational results obtained without the benefit of such a study even before possible modelling shortcomings are considered. Previous experience and CFD results from other researchers had suggested that the grids used here would be sufficiently fine so the lack of grid independence is disap-

pointing. To complete the study an even finer grid should be used, although solving the instability problems noted above is perhaps a higher priority.

Figures 3.19 to 3.24 show the calculated axial velocity and turbulence kinetic energy profiles for the medium and fine grids compared with experimental data. Figures 3.25 and 3.27 show the calculated pressure and velocity vector field for the medium grid respectively.

The initial calculation performed on the coarse grid took 18 minutes for the residual to converge by 8 orders of magnitude using a Silicon Graphics R5000 processor. The medium and fine grid calculations required 1 hour 17 minutes and 3 hours 20 minutes to converge by 4 orders of magnitude on the same machine. The very fine grid calculation required 6 hours 2 minutes to converge by four orders of magnitude using a 200MHz Intel Pentium Pro processor. The strategy used for obtaining initial conditions is explained in Section 3.3.3. The convergence criteria used here in terms of residual levels are conservative. The overall execution time for these analyses is therefore very reasonable using widely available desktop computing power. For this case the problem geometry and grid topology are straightforward so the time required for preprocessing should also not be excessive. It is reasonable to conclude that the necessary effort and time required to perform this type of analysis for base flows with the present method should not be restrictive to the design or evaluation engineer.

3.3.5 Conclusions

The present method has been applied to an axisymmetric base flow test case designed specifically for the validation of Navier-Stokes flow solvers. The issue of grid convergence has been shown to be very important for this type of flow. Validation of the present approach has been hampered by numerical instability thought to be due to the implicit treatment of the source term in the $k-\omega$ turbulence model. The results which have been obtained are in reasonable agreement with calculations by other researchers. The promise of this type of analysis for base flow problems has been underlined. The potential for relatively inexpensive and fast calculations has been demonstrated.

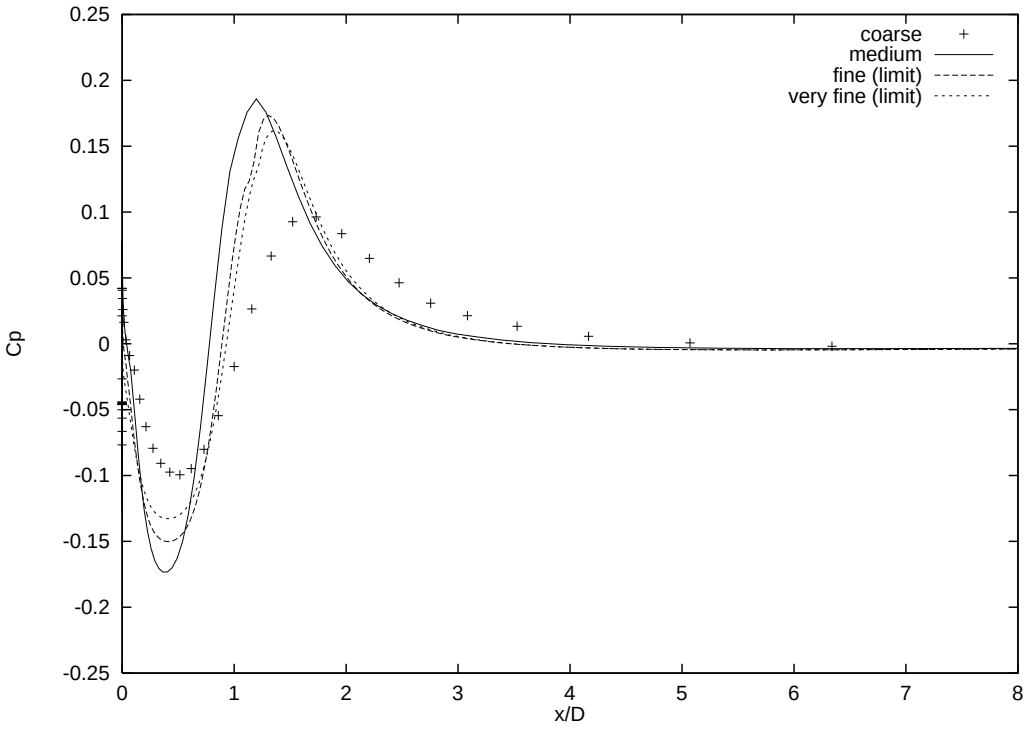


Figure 3.15: *Symmetric line pressure coefficient, 1B*

3.4 Discussion

In this chapter the applicability of the present method to aerodynamic problems of interest to industry has been assessed. Test cases representative of two classes of problem, namely missile forebody and base flows, have been examined. It should be noted that other types of problem, for example aerofoil flows, have been examined elsewhere[31],[32],[34].

It has been demonstrated that the present method performs well for supersonic missile forebody calculations involving strong oblique shocks. This conclusion is drawn not only from the results presented in this chapter but also from [62] where a range of forebody geometries and freestream Mach number were considered. The calculations were performed using widely available desktop computing power on a timescale measured in minutes. In contrast, a calculation performed using a three-dimensional flow solver achieved the same results, but took approximately 100 times as long and requires 50 times as much memory.

Application of the method to base flow proved more problematic. Although

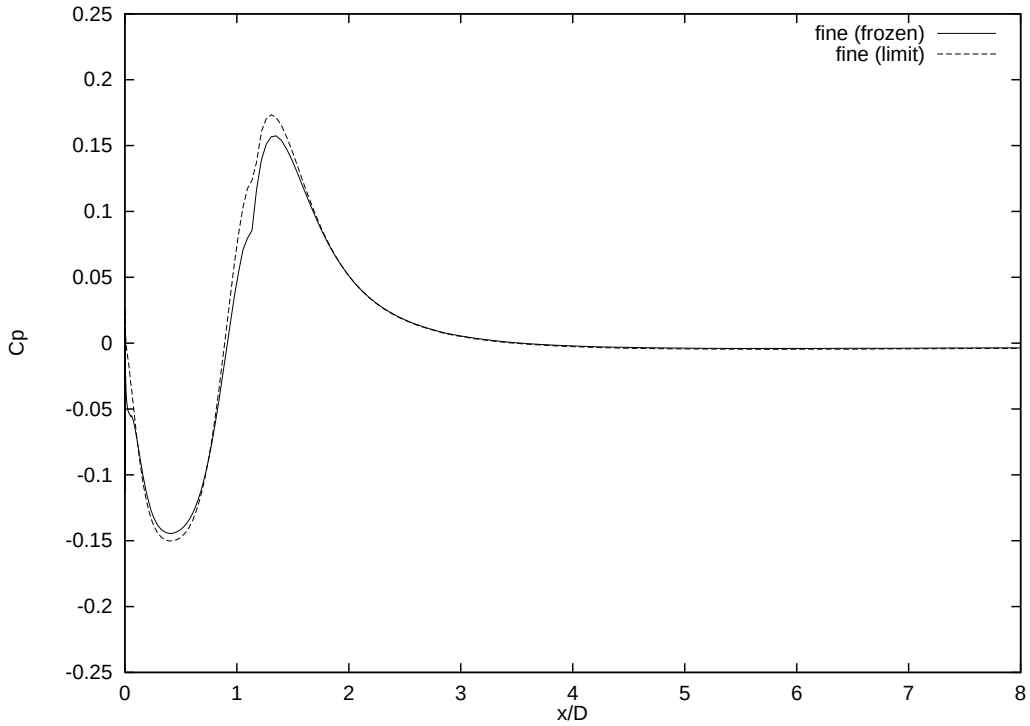


Figure 3.16: *Fine grid results, symmetric line pressure coefficient, 1B*

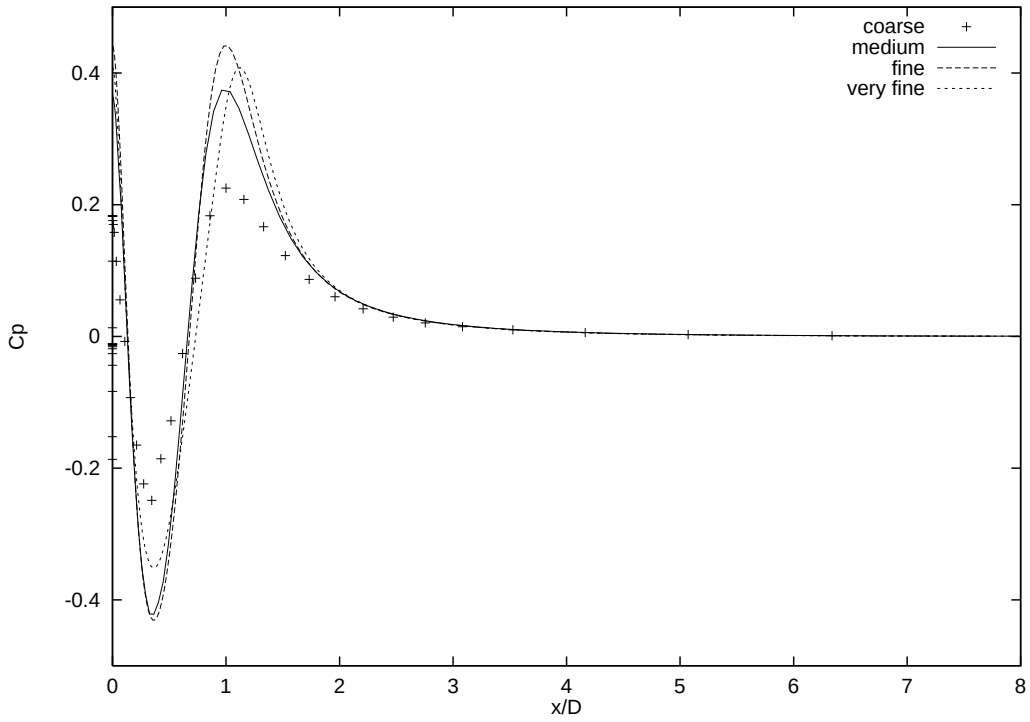


Figure 3.17: *Laminar results, symmetric line pressure coefficient, 1B*

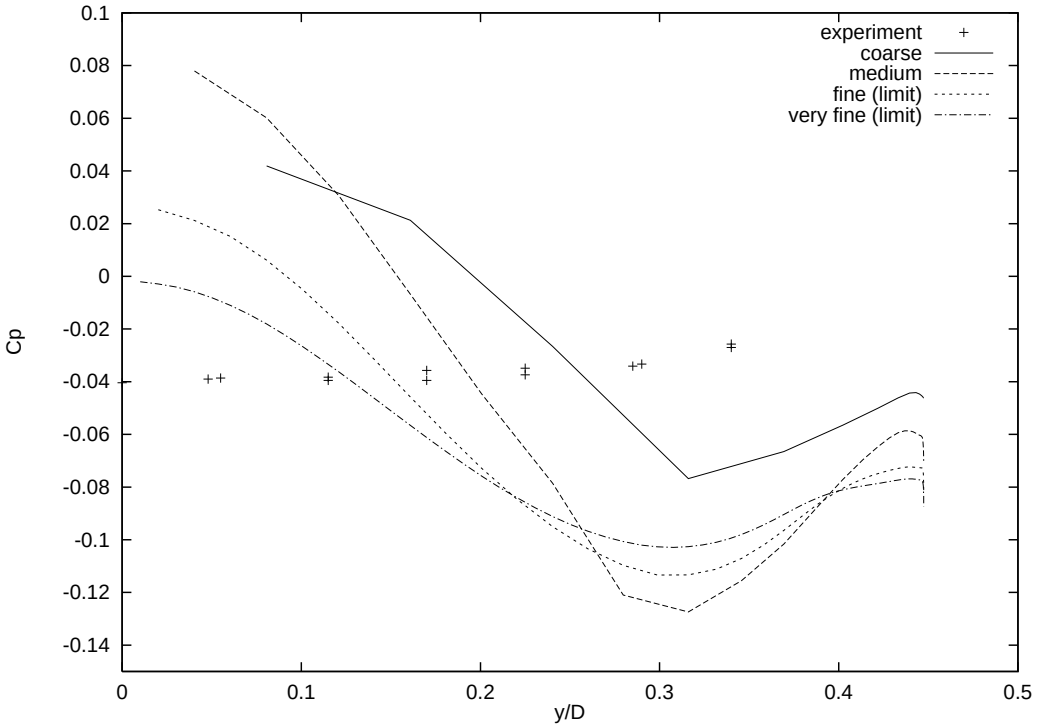


Figure 3.18: *Pressure coefficient along base, 1B*

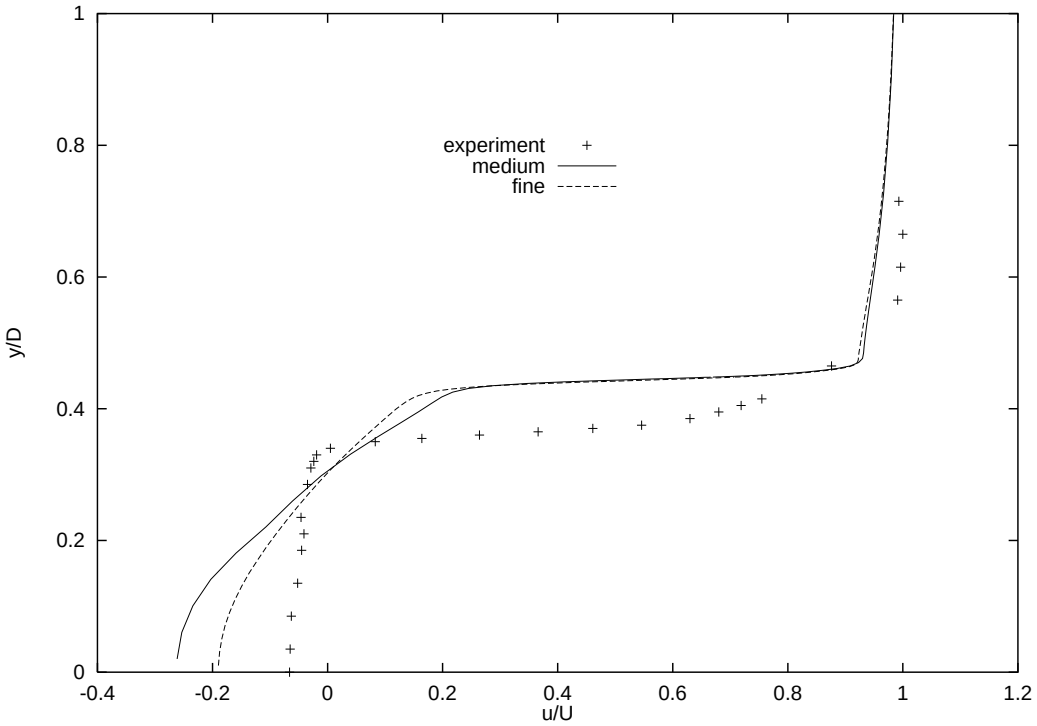


Figure 3.19: *Axial velocity profile at x/D=0.1, 1B*

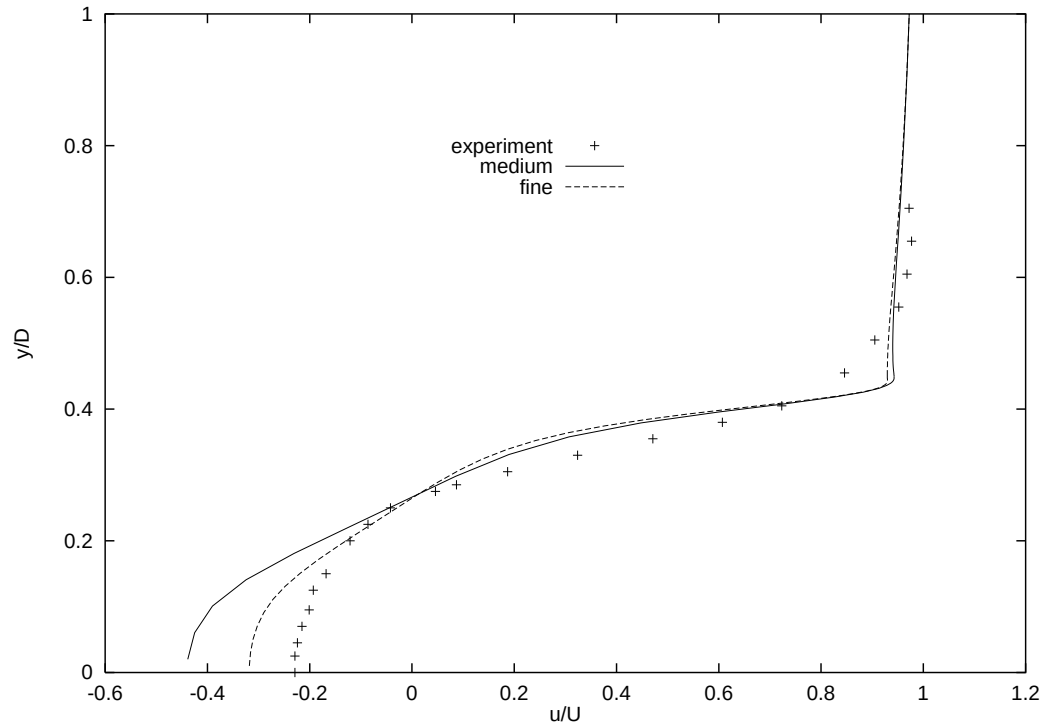


Figure 3.20: *Axial velocity profile at x/D=0.5, 1B*

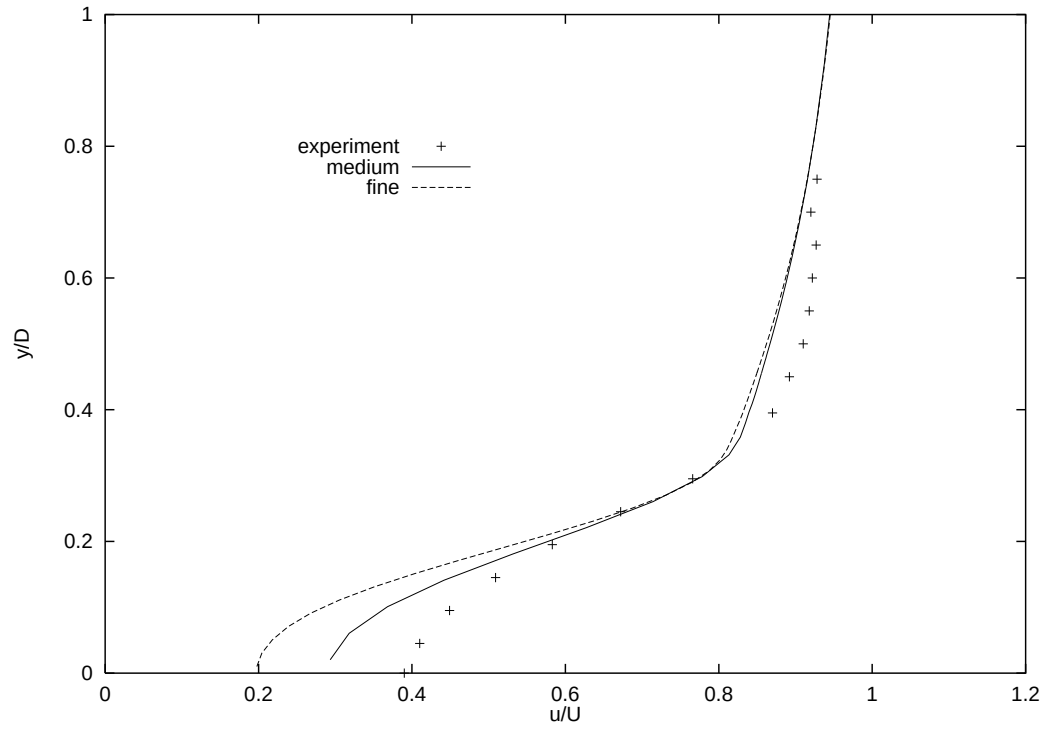


Figure 3.21: *Axial velocity profile at x/D=1.5, 1B*

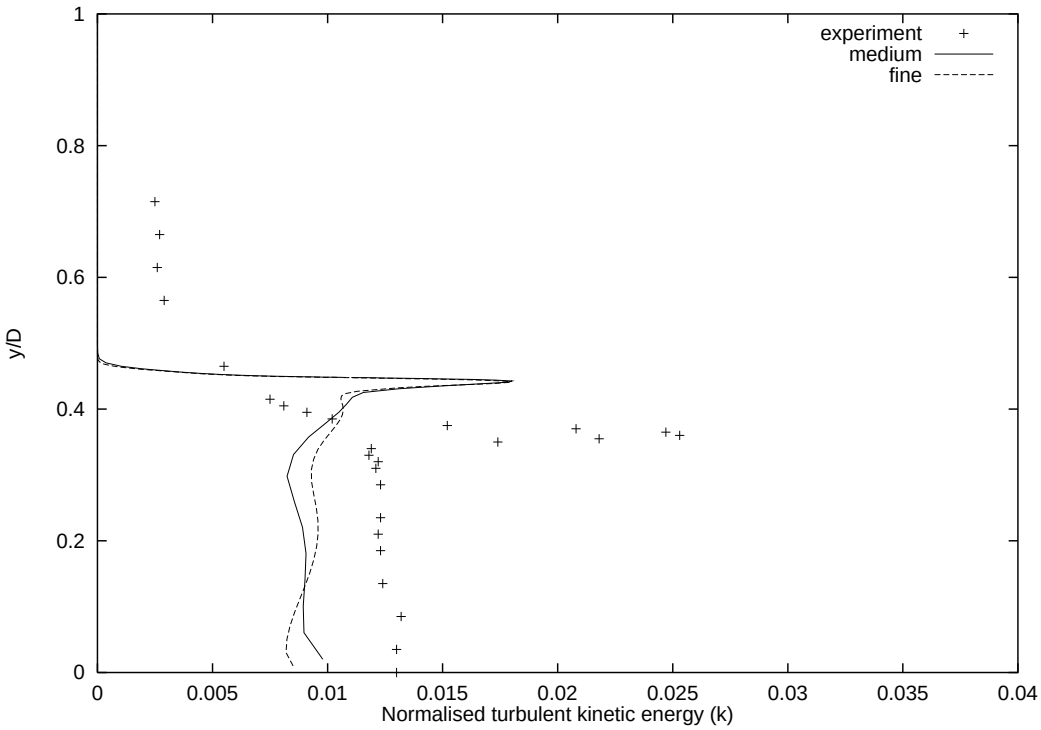


Figure 3.22: *Turbulent kinetic energy profile at x/D=0.1, 1B*

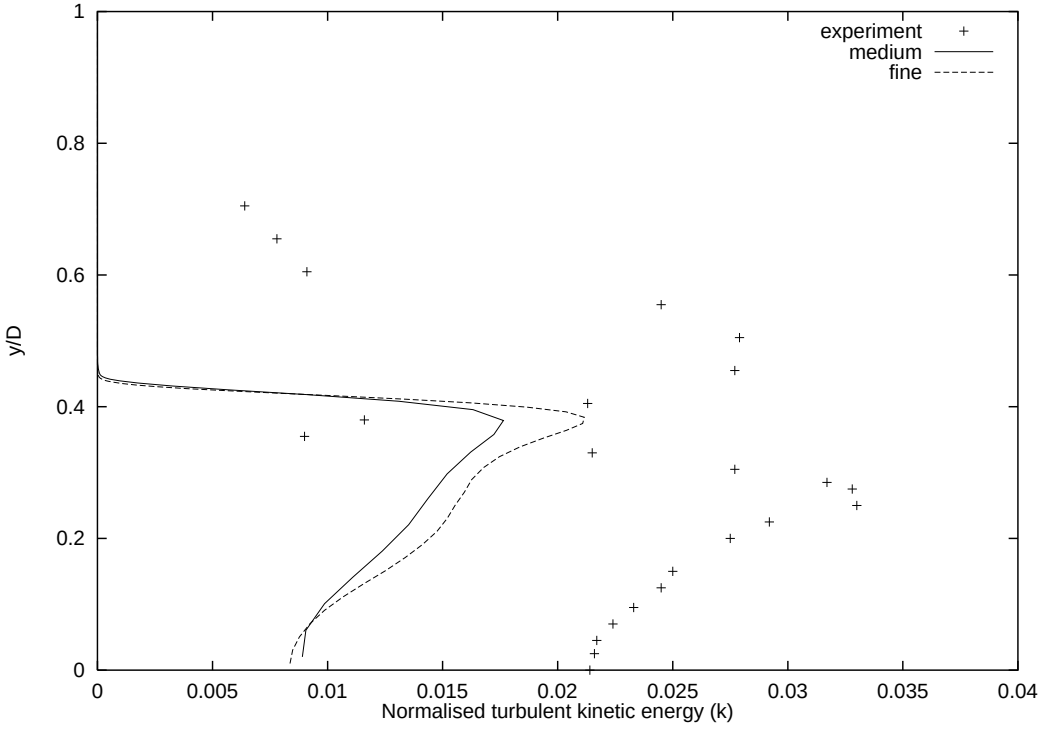


Figure 3.23: *Turbulent kinetic energy profile at x/D=0.5, 1B*

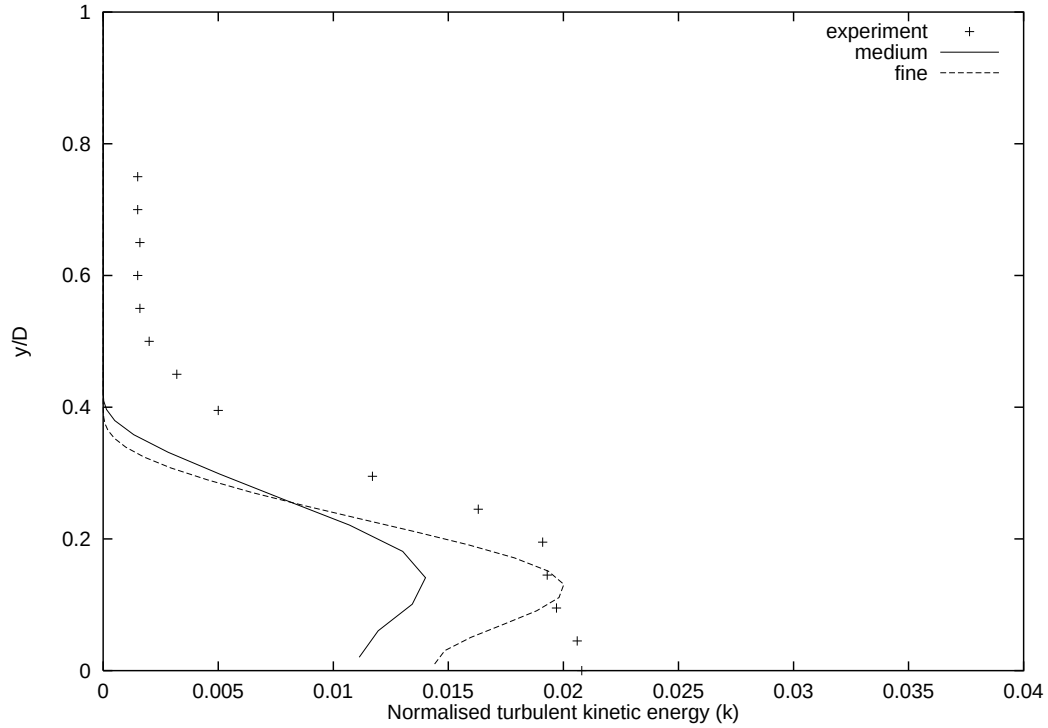


Figure 3.24: *Turbulent kinetic energy profile at x/D=1.5, 1B*

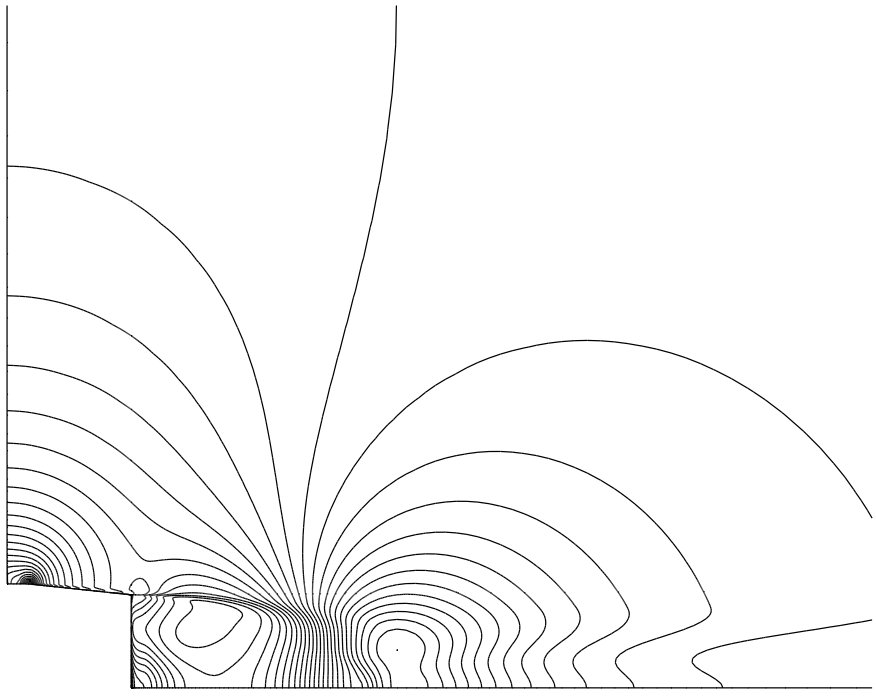


Figure 3.25: *Pressure contours, 1B*

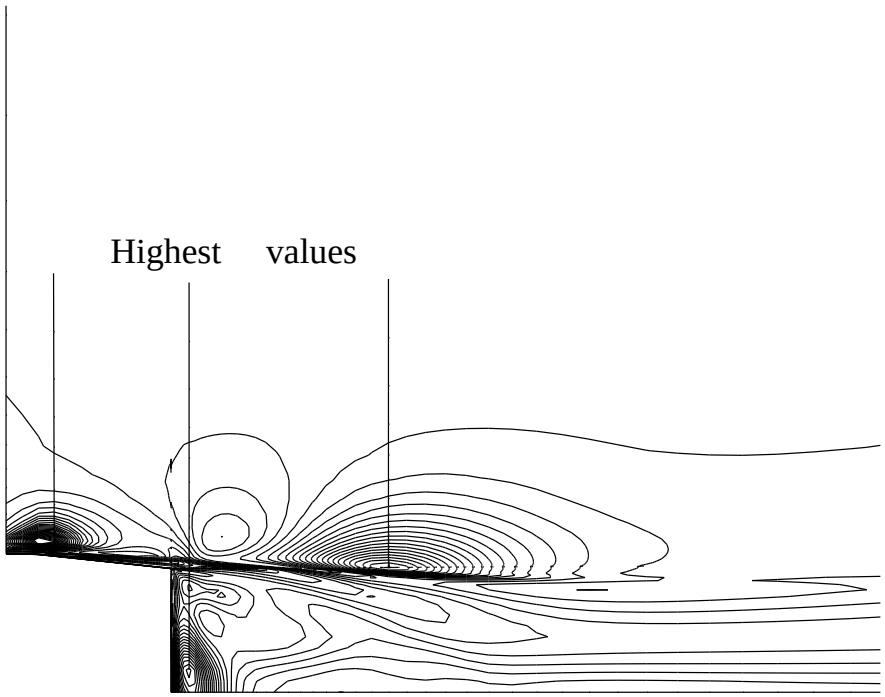


Figure 3.26: $\hat{\mathbf{P}}_k/\hat{\mathbf{D}}_k$ *contours, 1B*

it is still possible to obtain solutions relatively quickly, the method is not robust due to an instability associated with the implementation of the $k-\omega$ turbulence model. Before the method can be applied routinely and with confidence to flows of this type this shortcoming must be redressed. An improved implicit treatment of the turbulent source term may provide the solution to the robustness problem. On a more fundamental level, the deficiencies of two-equation turbulence models including the Boussinesq approximation are well known, see for example [54],[69]. The $k-\omega$ turbulence model gives accurate results for two-dimensional boundary layer flows. However, when the normal components of the Reynolds-stress tensor become non-negligible compared to the shear components, such as in flows with boundary layer separation and sudden changes in shear strain rate, the Boussinesq approximation becomes inaccurate. It is therefore unlikely that close agreement with experiment can be obtained for base flow problems, as seen in the present results. This obvious disadvantage has to be seen in the correct context. Simpler analyses, using semi-empirical methods or CFD with an algebraic turbulence model, give

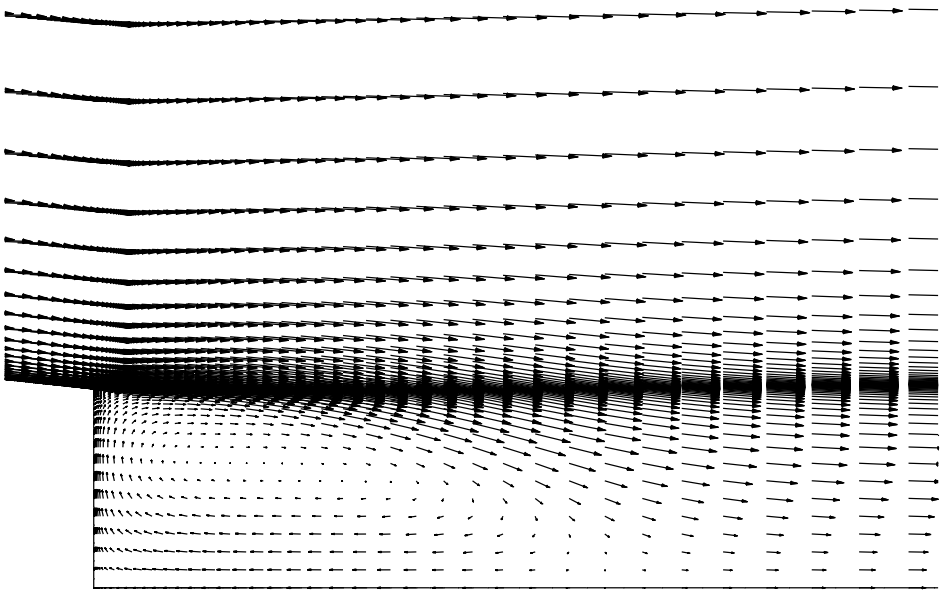


Figure 3.27: *Velocity vectors, 1B*

less accurate results in general and/or require case-dependent fine tuning. A CFD analysis employing a more advanced non-linear turbulence model entails prohibitive added complexity.

For axisymmetric problems of this nature a bottleneck in the analysis process often associated with other aerodynamic problems is avoided; grid generation is straightforward due to the relatively simple geometries. An engineer familiar with a structured grid generation tool should be able to construct a grid within a few hours, or modify an existing grid within a few minutes. The post-processing stage of an analysis is now also straightforward due to the wide availability of accomplished software for this purpose. For missile forebody and base flows the pre- and post-processing associated with the present method should not impede the engineer who requires routine and efficient analyses. For the calculation of axisymmetric forebody flows the present method therefore fulfills the criteria of accuracy and efficiency. Before the present method can be used with the same confidence for base flows further refinement of the numerical method is necessary, although the potential of

a Navier-Stokes flow solver for these demanding problems is clear. At present, a standard two-equation turbulence model provides a good balance between accuracy and complexity.

Chapter 4

Scientific Investigation: Shock Reflection Hysteresis in an Underexpanded Jet

4.1 Introduction

4.1.1 Underexpanded Jets

A jet is said to be underexpanded if the gas pressure at the nozzle exit is greater than the ambient pressure. When this pressure ratio is large, the jet is said to be highly underexpanded and the jet plume is characterised by a complex repeated shock structure. Many examples of real aerodynamic flows where knowledge of the behaviour of this type of jet is necessary can be found in the literature. Rocket exhausts at high altitude may have highly underexpanded plumes. The study of such flows is important for predicting propulsive efficiency and plume signatures[70],[71],[72]. Experimental studies are also important for the validation of CFD codes which are used extensively in plume signature prediction work[71]. Vehicle manoeuvring thrusters may also give rise to underexpanded plumes[70],[71]. Proposed scramjet engine designs include supersonic underexpanded fuel injectors for which detailed modelling of the mixing process is required[73]. The behaviour of highly underexpanded jets must be understood for accurate consequence and risk assessment studies

for accidental and operational releases of high pressure gas[74]. Underexpanded exhaust plumes interacting with the freestream may arise in aeropropulsion testing in wind tunnels[75]. An appreciation of underexpanded jet plume structures is important for the problems of plume-surface and plume-plume interaction and avoiding wall interference when studying thruster nozzles in ground facilities like vacuum chambers[76]. Supersonic underexpanded jets are used in experiments to examine the aeroacoustic properties of strong shock cell structures [77]. Underexpanded hypersonic jets are used to study aerothermodynamic characteristics of hypersonic vehicle models in wind tunnels[78]. The same paper includes a very comprehensive account of how underexpanded jets are used in experimental and numerical studies of nonequilibrium thermo- and gasdynamic processes in hypersonic flow. The experimental studies of Crist[70] and Abbett[79] established the basic wave structure of a highly underexpanded jet plume and that regular or Mach reflection may occur depending on the conditions. The method of characteristics has been employed by many authors[79],[80],[81],[82] in an attempt to develop predictive models for the core expansion and Mach disc location.

A phenomenon associated with low density highly underexpanded jets which has yet to be fully understood is shock reflection hysteresis as reported by Welsh[71]. For a (laminar) nitrogen jet exhausting from a nominally Mach 3 nozzle a set of conditions exist at which either regular or Mach reflection may occur depending on the history of the plume development. Since the reflection type strongly influences the interaction of the jet with its environment an understanding of the phenomenon and definition of the hysteresis loop limits are important. Quantitative experimental investigation of this problem, aside from being expensive, suffers from probe interference with the jet structure, necessitating the development of non-intrusive measurement techniques[71]. However, these promising methods have yet to reach full maturity and the potential of a CFD analysis is clear, providing the motivation for this study.

4.1.2 Shock Reflection Hysteresis

The existence of a hysteresis effect in the type of reflection of a two-dimensional oblique shock wave at a wall or symmetric line has been established in recent years. The reflection of the oblique shock wave may take the form of a regular reflection (RR) or Mach reflection (MR). The type of reflection which occurs depends on the Mach number upstream of the incident shock and the shock angle. However, there is a dual solution domain where either type may occur and the solution exhibits a hysteresis effect. A summary of the elements of this topic which are of interest to this study is included in section 4.2.

4.1.3 CFD and Underexpanded Jets

Axisymmetric Euler and Navier-Stokes solvers have been used to obtain solutions for underexpanded jet plumes with impressive results, see for example[73],[74],[83],[84]. These calculations demonstrate good agreement with experiment for a wide range of conditions using parameters such as Mach disc location and centreline velocity and are reported to capture the complex wave structure in detail. No CFD study of the hysteresis phenomenon in underexpanded jets has been found. The hysteresis phenomenon associated with two-dimensional shock reflection has been successfully modelled numerically, see section 4.2. In this case the crucial quantities (upstream Mach number and incident shock angle) are relatively easy to control and model correctly in a computational approach. However in the case of shock reflection in the underexpanded jet, these quantities are inherent parts of the calculation rather than being "set" a priori. All of the interacting features of the complex flow field must be resolved accurately, making this problem far more demanding. The application of a Navier-Stokes flow solver to shock reflection hysteresis in an underexpanded jet is described in section 4.3, with the objective of contributing to the understanding of this type of flow by combining the known features of two-dimensional shock reflection (see section 4.2) with the detailed solutions provided by a CFD analysis. The axisymmetric flow solver described in Chapter 2 will be used. In Chapter 3 it was demonstrated how the axisymmetric flow solver has a considerable performance advantage over a three-dimensional flow solver. This feature is important to the

present problem in particular and to jet plume studies in general since multiple solutions are typically required over a wide range of parameters, for example the ratio of nozzle exit and ambient pressures.

4.2 Two-Dimensional Shock Reflection Hysteresis

4.2.1 Introduction

Two different types of shock wave reflection, now known as Regular Reflection (RR) and Mach Reflection (MR), were first recorded by Ernst Mach in 1878. Analytic models for RR and MR were first developed by von Neumann in the 1940s. The existence of a hysteresis effect in the transition between types was first suggested in 1979[85]. Subsequent experimental[86] and numerical studies[87],[88],[89] have since confirmed the existence of the phenomenon. These references together with review papers[90],[91] provide an extensive introduction to the topic of shock reflections and associated phenomena. This chapter summarises the parts of the above references relevant to the main study of underexpanded jets for which it is useful to introduce the theory and terminology of the two-dimensional case, and leans particulary on [86] and [90]. In addition, the current explanation for the hysteresis phenomenon is discussed. An attempt is made to fill the gaps in the explanation by applying the principle of minimum entropy production.

4.2.2 Shock Reflection Types

Schematic diagrams of the Regular and Mach reflection types are shown in Figure 4.1. In the figures, **i** is the incident shock wave, **r** is the reflected shock wave, **m** is the Mach stem and **s** is the slip line. The reflection and triple points are labelled **R** and **T** respectively. ϕ_i and ϕ_r are the incidence angles of **i** and **r** respectively. θ_i, θ_r and θ_m are the flow deflections on passing through **i**, **r** and **m** respectively. The regular reflection, as shown in Figure 4.1(a), consists of an incident and reflected shock wave meeting at point **R** on the reflecting surface. The incidence angle ϕ_i is small enough such that the flow deflection caused by the reflected shock wave is sufficient to cancel that caused by the incident shock wave. In this way the condition of flow tangency

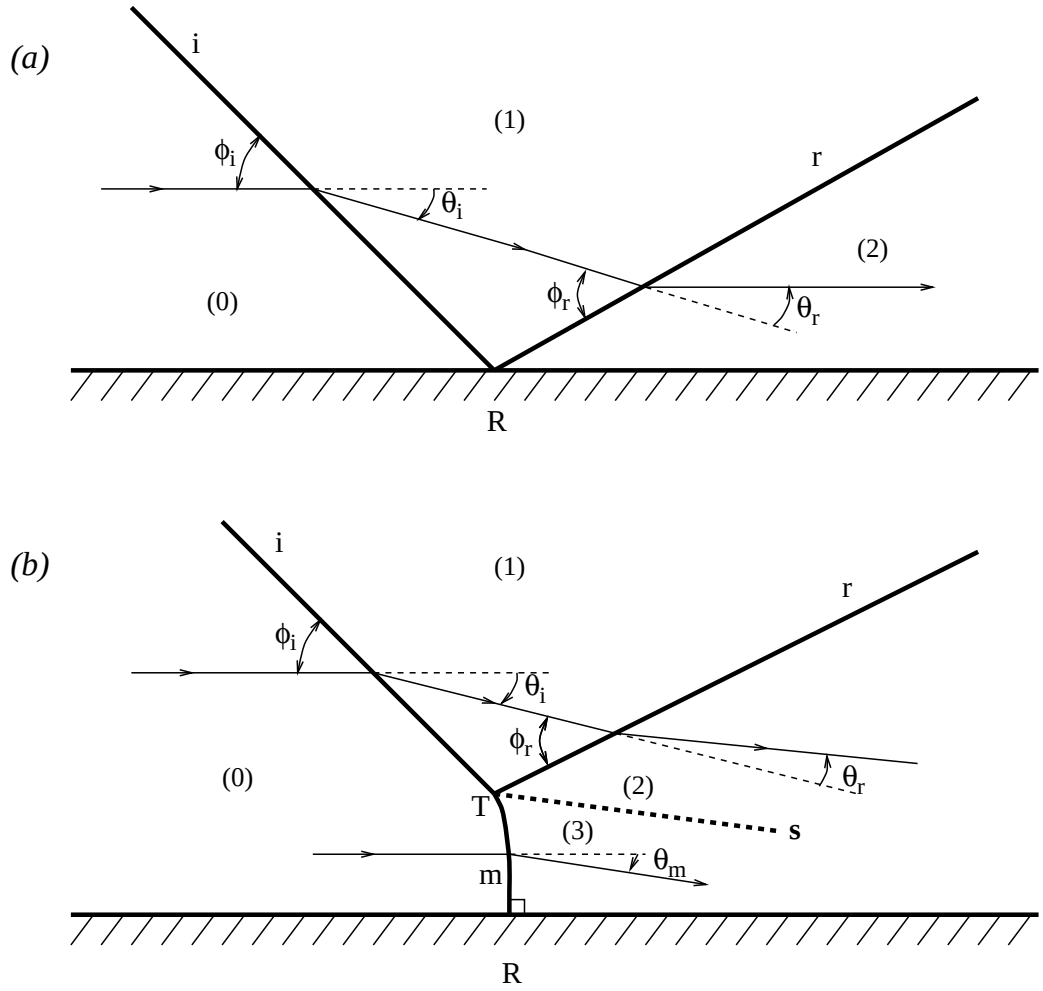


Figure 4.1: *Schematic diagrams of* (*a*) *regular reflection and* (*b*) *Mach reflection*

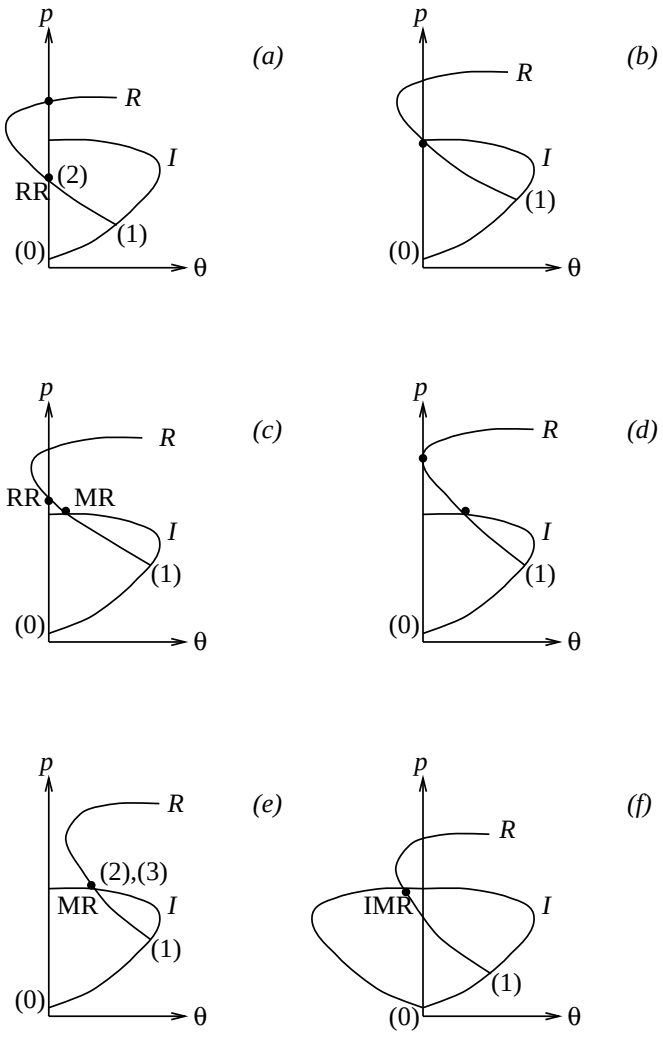


Figure 4.2: *Pressure-deflection diagrams*

at the reflecting surface is maintained. Thus the boundary condition for RR is

$$\theta_i - \theta_r = 0 \tag{4.1}$$

The Mach reflection type, shown in Figure 4.1(b), consists of incident and reflected shock waves, a Mach stem and a slip line, all of which meet at the triple point. The Mach stem is perpendicular to the reflecting surface and may curve as shown to become concave as viewed from upstream. The flow being processed by the Mach stem may be considered to form a buffer region between the flow tangency condition at the reflecting surface and a detached regular reflection which cannot maintain this condition on its own. The flow behind the Mach stem is subsonic. The net flow deflection behind the triple point is in general non-zero. Note that since the Mach stem is curved the conditions in region (3) are non-uniform; conditions behind the Mach stem in the vicinities of **R** and **T** will be denoted with the subscripts $_{3,R}$ and $_{3,T}$ respectively. Since flow on either side of the slip line must be parallel, the boundary conditions for a Mach reflection are

$$\begin{aligned} \theta_i - \theta_r &= \theta_m \\ p_2 &= p_{3,T} \end{aligned} \tag{4.2}$$

It is important to note that equation (4.1) considers local conditions in the vicinity of **R** only. In the same way equations (4.2) consider local conditions in the vicinity of **T**. To apply these relations globally the shock waves **i** and **r** and slip line **s** must be straight, implying regions of uniform flow.

4.2.3 The Dual Solution Domain

Graphical solutions in the pressure-deflection ($p-\theta$) plane, which are obtained from oblique shock theory[92], are useful for understanding shock-wave phenomena, and in particular the conditions for which each reflection type is possible. Figures 4.2 are examples of ($p-\theta$) diagrams. In these figures, the I polar represents the locus of all possible solutions (1) when the free-stream state (0) is deflected through an angle θ via an oblique shock wave. Similarly, the R polar represents the locus of all possible solutions (2) when the free-stream (1) is deflected through an angle θ via

an oblique shock wave. In the figures θ_i is increasing as we progress from Figure 4.2(a) to Figure 4.2(e). The RR boundary condition (4.1) implies that the solution of a regular reflection is at a point where the R polar intersects the p-axis, i.e. where θ is zero. Figure 4.2(a) shows two such points; the higher pressure point is observed to be unstable in experiments[86] and has been shown to be aphysical[93], a result which will be discussed in Section 4.2.6. Consequently, conditions at (2) are represented by the point RR. The MR boundary conditions (4.2) imply that the solution of a Mach reflection is at a point where the I and R polars intersect, e.g. Figure 4.2(e). States (2) and (3) of Figure 4.1(b) map onto the point MR indicated, state (2) being on R and state (3) on the I polar.

Three interesting cases which lie between those discussed above are shown in Figures 4.2(b) to 4.2(d). First, reconsider the case represented by Figure 4.2(a). If the upstream Mach number is held constant but the angle ϕ_i is gradually increased then the solution point RR moves up the p-axis until the condition represented by Figure 4.2(b) is reached. Since at this point both polars and the p-axis intersect, both RR and MR solutions are possible. The smallest incident shock angle for which MR is possible for a given upstream Mach number is represented at this point (except in the special case of *Inverted* Mach Reflection as discussed below). This condition is known as the *von Neumann criterion*. As ϕ_i is increased further the situation represented by Figure 4.2(c) occurs. The R polar intersects both the I-polar and the p-axis, so again both RR and MR solutions are possible although in this case with different values of p and θ. This figure represents typical solutions in the dual solution domain. The second limit of the dual solution domain is represented by Figure 4.2(d) where ϕ_i has been further increased such that the R-polar is tangent to the p-axis. The largest incident shock angle for which RR is possible for a given upstream Mach number is represented at this point. This condition is known as the *detachment criterion*. Any further increase in ϕ_i results in the situation shown in Figure 4.2(e) where the R-polar no longer intersects the p-axis so only MR is now possible.

As noted above, the von Neumann condition is at present accepted as the lower pressure limit to the dual solution domain. Recall that this condition is represented in the (p, θ) plane by Figure 4.2(b). If from this condition the incident shock angle

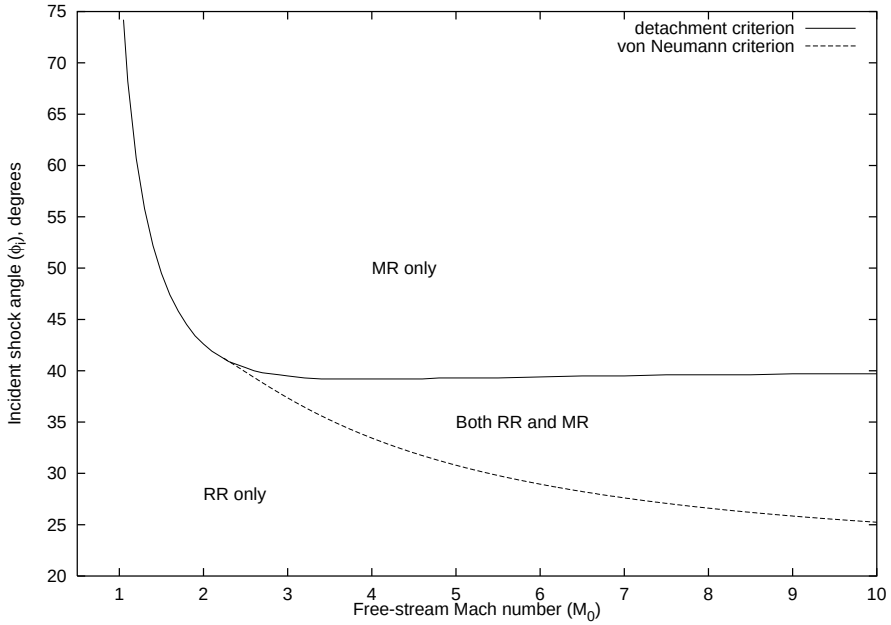


Figure 4.3: *Domains of possible reflection types*

is increased then RR is observed, see Figure 4.2(a). Hornung[90] discusses the alternative of *Inverted* Mach Reflection (IMR): if we consider that the flow may be deflected away from the wall by MR, i.e. θ_m in Figure 4.1(b) is negative, then this would be represented in the (p, θ) plane by the point IMR in Figure 4.2(f). The curvature of **m** is then necessarily convex as seen from upstream in order to achieve this deflection. Note that an IMR always has the alternative of a RR. The reflection type observed in experiment is RR unless it is suppressed by raising the downstream pressure, in which case the IMR occurs[90],[94]. This phenomenon has yet to be fully explained, and will be returned to in Section 4.2.6.

4.2.4 Analytic Solutions in the Dual Solution Domain

Analytic solutions for RR, MR the von Neumann and detachment criteria will be used in this study. They are readily obtained using the arguments of Section 4.2.3 and oblique shock theory, making certain simplifying assumptions for the MR calculations. Their calculation is straightforward and is outlined here for completeness. Figure 4.3 indicates the location of the dual solution domain in the (M_0,ϕ_i) plane. The overlap region where both RR and MR are possible is clearly shown.

Oblique Shocks

A result of oblique shock wave theory (see for example [95]) is that for any given upstream Mach number M_0 the same flow deflection θ can be achieved via two distinct straight shock solutions, provided that θ is less than the maximum deflection possible θ_{max}. This result is demonstrated in Figure 4.4 where the curve is the locus of all possible solutions in the (θ,β) plane when a free-stream of Mach number M_0 is deflected through an angle θ by an oblique shock wave at angle β to the free-stream. The solution corresponding to the larger value of β is termed the 'strong' solution since the changes across the shock are more severe than for the 'weak' solution at the shallower angle. The density, pressure, Mach number and temperature changes across an oblique shock are given by

$$\frac{\rho_1}{\rho_0} = \frac{(\gamma+1)\,M_0^2 \sin^2\beta}{(\gamma-1)\,M_0^2 \sin^2\beta + 2} \tag{4.3}$$

$$\frac{p_1}{p_0} = 1 + \frac{2\gamma}{\gamma+1}\left(M_0^2 \sin^2\beta - 1\right) \tag{4.4}$$

$$M_1^2 \sin^2(\beta-\theta) = \frac{M_0^2 \sin^2\beta + [2/(\gamma-1)]}{[2\gamma/(\gamma-1)]\,M_0^2 \sin^2\beta - 1} \tag{4.5}$$

$$\frac{T_1}{T_0} = \frac{p_1}{p_0}\frac{\rho_0}{\rho_1} \tag{4.6}$$

where (0) and (1) denote conditions before and after the shock respectively and γ is the ratio of specific heats. The flow deflection θ due to the oblique shock is given by

$$\tan\theta = 2\cot\beta\left[\frac{M_0^2 \sin^2\beta - 1}{M_0^2(\gamma + \cos 2\beta) + 2}\right] \tag{4.7}$$

Thus given the upstream conditions and shock angle β the downstream conditions can be calculated in a very straightforward manner.

Regular Reflection

A regular reflection solution involves two oblique shocks. Referring to Figure 4.1(a), the conditions at (1) and the deflection θ_i are calculated using the oblique shock

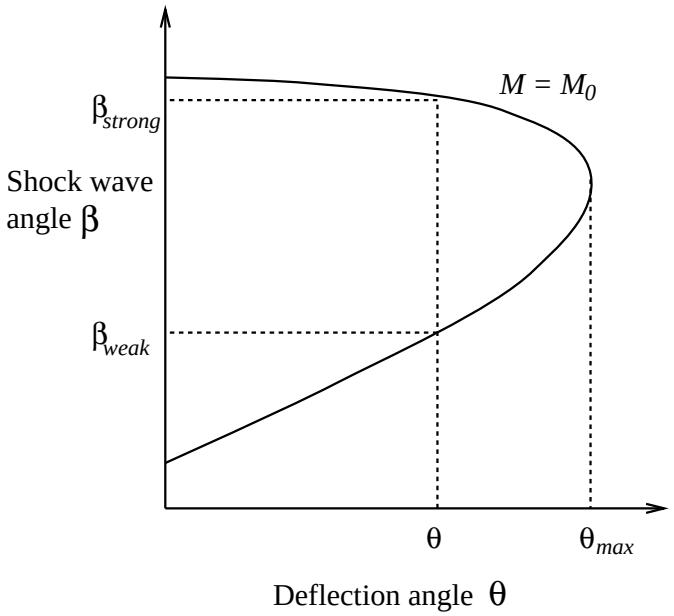


Figure 4.4: *Weak and strong solutions in the (θ,β) plane*

relations, see directly above. To calculate the conditions at (2), ϕ_r is obtained by invoking the condition (4.1) and solving equation (4.7). It is not possible to solve for ϕ_r directly so a simple iterative procedure (such as a bisection method) is required. Note that the 'weak' solution is assumed normally to be correct in the absence of additional boundary conditions (see Section 4.2.6). Figure 4.5 includes the pressure ratio $p_{2,RR}/p_0$ for various M_0 within the dual solution domain, the subscript RR referring to the regular reflection type discussed here. Note that in the figure each curve extends from $\phi_i = \phi_i^N$ on the left of each curve to $\phi_i = \phi_i^D$ on the right.

The Detachment Criterion

For a given free-stream Mach number M_0 the incident shock angle ϕ_i^D corresponding to the detachment criterion (see Section 4.2.3) is obtained by gradually increasing ϕ_i when calculating a RR until the solution for **r** can no longer be achieved. The variation of ϕ_i^D with M_0 is shown in Figure 4.3.

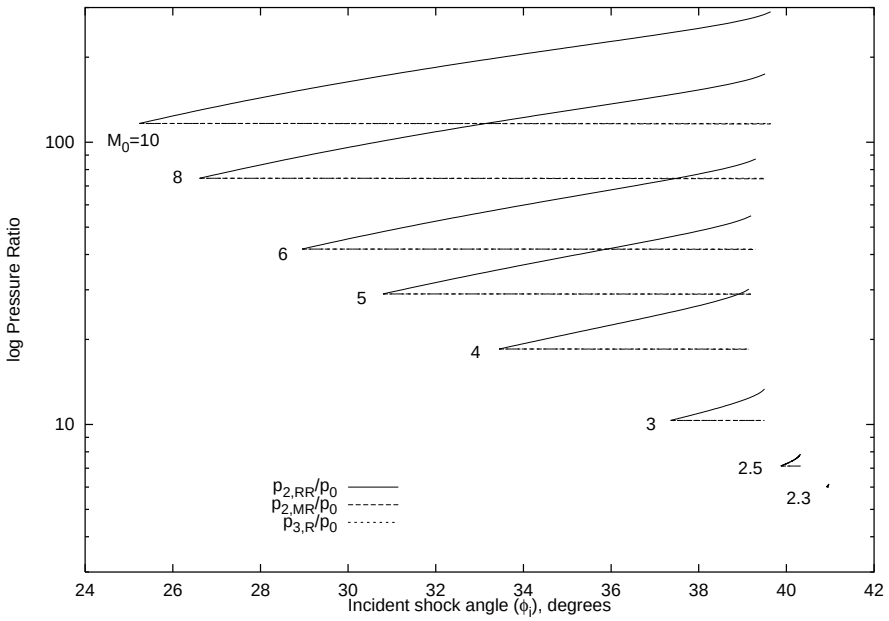


Figure 4.5: *Pressure ratios across shock reflections in dual solution domain*

The von Neumann Criterion

For a given free-stream Mach number M_0 the incident shock angle ϕ_i^N corresponding to the von Neumann criterion (see Section 4.2.3) is obtained by first calculating the pressure $p_{3,R}$ behind a normal shock. For some small value of ϕ_i a RR solution is then calculated; the pressure p_2 behind **r** should be less than $p_{3,R}$. ϕ_i is then gradually increased until p_2 equals $p_{3,R}$. The variation of ϕ_i^N with M_0 is shown in Figure 4.3. Note that for $M_0 \leq 2.20$ the von Neumann criterion does not exist.

Mach Reflection

Referring to Figure 4.1(b), a first guess for the conditions at (2) and (3,T), for which the boundary conditions (4.2) apply, is obtained by calculating the pressure behind **T** if **m** is locally normal to the flow, i.e. by taking $p_{3,R}$ as an initial guess for $p_{3,T}$. The deflections θ_m and θ_r are then calculated for this pressure value using equation (4.7) with equation (4.4), the conditions at (0) and (1) being known. The pressure $p_{3,T}$ is gradually decreased until the conditions (4.2) are true. Note that this analysis gives a 'Mach reflection solution' in that the gasdynamic conditions at (0),(1),(2) and (3) are known. However, note that the length of **m** and its inclination

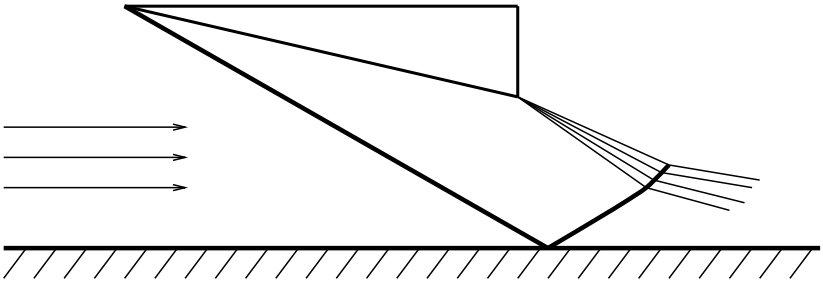
between **R** and **T** are not calculated. A more involved approach has been used[96] to estimate the length of **m** with some success. Regarding the conditions in region (3), note that this analysis yields two sets of conditions here; conditions (3,T) behind **T** which uphold the MR boundary conditions, and conditions (3,R) behind **R** where **m** is normal to the free-stream. Figure 4.5 includes the pressure ratios $p_{2,MR}/p_0$ and p_3/p_0 for various M_0 within the dual solution domain. Note that $p_{2,MR}/p_0$ and $p_{3,T}/p_0$ are equivalent due to condition (4.2). Note also that $p_{2,MR}/p_0$ and $p_{3,R}/p_0$ diverge by only a very small amount; in Figure 4.5 their respective curves are all but identical.

4.2.5 The Hysteresis Phenomenon

Pressure-deflection maps provide useful illustrations of how a dual solution domain can occur. However, when more than one reflection type is possible no clue is given by these maps as to which mode actually occurs. The existence of a hysteresis effect in the shock reflection type in the dual solution domain was first postulated by Hornung[85]. A number of relatively recent experimental studies have contributed to the understanding of this type of flow[85],[92],[94],[97] culminating in the first experimentally recorded shock reflection hysteresis[86]. Figure 4.6 shows schematically a typical experimental set up for examination of this problem. Wave diagrams for (a) Regular Reflection and (b) Mach Reflection are also shown. Recent numerical studies have also predicted the phenomenon[87],[88],[89],[98]. If the dual solution domain is approached from a condition for which only RR may occur, then RR persists until the detachment criterion is reached where the reflection type flips to MR. Likewise, if the dual solution domain is approached from a MR condition, then MR persists until the von Neumann condition is reached where the reflection type flips to RR. Figure 4.7 shows a schematic representation in the (ϕ_i,l_m/w) plane, l_m being the length of the Mach stem **m**, and w being the length of the wedge from leading to trailing edge. Note that as the von Neumann condition is approached, l_m becomes vanishingly small. The most complete explanation to date for the hysteresis is provided by Hornung[90] and is summarised below.

A feature of MR not present in RR is the existence of a characteristic length,

(a)



(b)

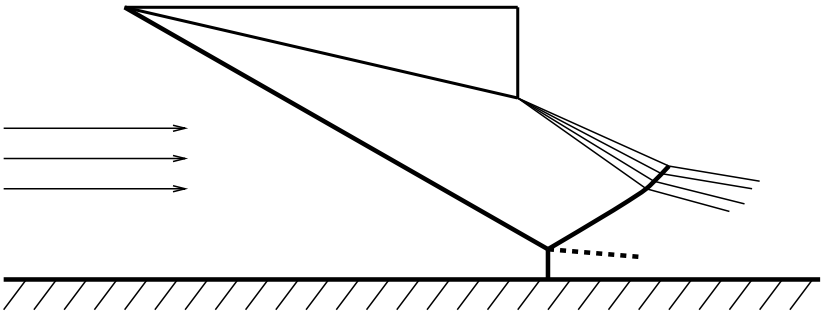


Figure 4.6: *Use of a wedge shock generator for* (*a*) *regular reflection and* (*b*) *Mach reflection experiments*

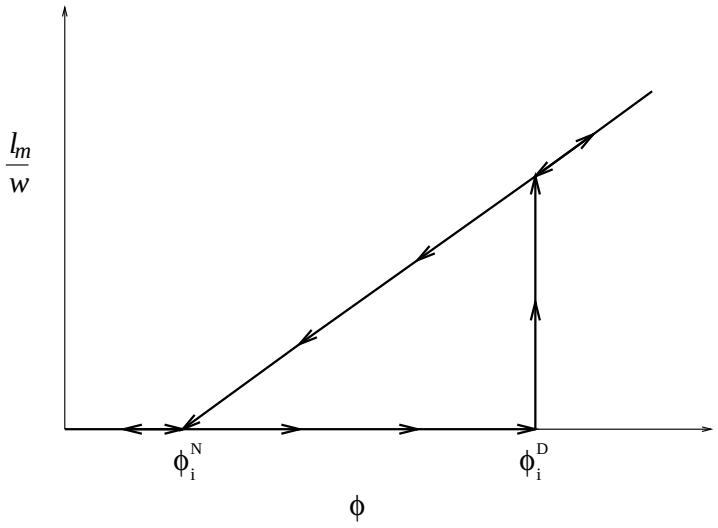


Figure 4.7: *Schematic illustration of the hysteresis loop in the (ϕ_i,l_m) plane*

namely the length of the Mach stem. In the absence of mechanisms which may provide a characteristic length scale such as viscous effects, heat dissipation and relaxation, the length scale must be provided by the geometry of the boundary conditions. The obvious candidate is the length w from the leading to trailing edge of the wedge creating the incident shock. Such a wedge has been used in all experiments to date. The *information condition* requires that for MR to occur an information path must be open from the trailing edge of the wedge to the interaction area in order to transmit the characteristic length information. This information path is provided by an expansion from the trailing edge reaching the subsonic area behind the Mach stem. The transition criteria are then explained using this condition. An information path does not exist in the case of RR, thus when the dual solution domain is approached from a RR condition, the RR persists until the detachment criterion is reached. An information path is open in the case of MR, thus when the dual solution domain is approached from a MR condition, there is no impediment to MR occurring so it persists until the von Neumann criterion is reached.

This description of the mechanisms causing the hysteresis does not seem complete. The explanation for the persistence of RR until the detachment condition is

reasonable, and is the only one to be suggested as yet in the literature. Here a new alternative explanation is proposed, based on upstream boundary conditions. Referring to Figure 4.1(b), a boundary condition for MR is that conditions immediately upstream of the contour formed by the **i**,**m** shock front are freestream conditions. The equivalent RR configuration satisfies this condition, so RR can conceivably occur spontaneously from the MR configuration (within the dual solution domain). However, referring to Figure 4.1(a), a boundary condition for RR is that conditions immediately upstream of the contour formed by the **i** shock front are freestream conditions. The equivalent MR configuration does not satisfy this condition, so MR cannot occur spontaneously from the RR configuration.

There is as yet no explanation for the persistence of MR, it has only been explained how there is no impediment to MR; it has not been explained why the MR should persist in preference to RR. An explanation for the MR persistence symmetrical to the RR persistence arguments would require an identified impediment to the existence of RR when an MR condition exists within the dual solution domain. An alternative approach is the possibility that when either RR or MR is possible, MR may be the preferred solution. Note that such an argument would not contradict the above explanations for the persistence of RR (where MR is impeded). An explanation of why MR is preferred would then constitute an explanation for the persistence of MR, i.e. for one half of the hysteresis loop. The principle of minimum entropy production will be applied to this problem in Section 4.2.6 in an attempt to contribute to the explanation of the hysteresis phenomenon.

4.2.6 The Principle of Minimum Entropy Production

The principle of minimum entropy production[99] states that if more than one steady state solution is compatible with the problem boundary conditions then nature prefers the solution of minimum dissipative structure i.e. the observed solution is that with the minimum rate of entropy production. The principle has been applied to the deflection of supersonic flow by wedges to explain the prevalence of 'weak' over 'strong' shock solutions[93],[100]. By extension, the prevalence of 'weak' over 'strong' regular reflections, a problem already mentioned in Section 4.2.3, has also

been explained using this principle[93]. Pseudo-steady shock reflection (as opposed to the steady shock reflections considered in this work) has also been examined using the principle[101]. Thus a precedent clearly exists for using the principle to help explain phenomena associated with shock wave reflections. Encouraged by this, the principle will be applied below to two shock reflection phenomena which have not yet been fully explained, namely why the von Neumann criterion is the lower pressure limit on the dual solution domain and IMR is not normally observed (see Section 4.2.3) and the persistence of MR in the hysteresis loop (see Section 4.2.5). As a preliminary, the principle is first re-applied to supersonic flow deflection and regular shock reflections.

Supersonic Flow Deflection

If a supersonic free-stream of Mach number M_0 is deflected by a wedge at incidence θ to the free-stream (θ being less than the shock detachment angle) then oblique shock theory admits two solutions (see Section 4.2.4). For an ideal gas[102]

$$ds = C_p \, d\ln T - R \, d\ln p \tag{4.8}$$

where s denotes entropy. This can be integrated directly to yield

$$s_1 - s_0 = C_p \, d\ln \frac{T_1}{T_0} - R \, d\ln \frac{p_1}{p_0} \tag{4.9}$$

which is an expression for the increase in entropy when an ideal gas is changed from state (0) to state (1) by some process. Here C_p is the specific heat at constant pressure and R is the specific gas constant. If the process is an oblique shock, then equations (4.3) to (4.6) can be substituted into equation (4.9) to obtain

$$\begin{aligned} s_1 - s_0 = C_v \Bigg\{ & \gamma \ln \left[\frac{2}{(\gamma+1) M_0^2 \sin^2 \beta} + \frac{\gamma-1}{\gamma+1} \right] \\ & + \ln \left[\frac{2\gamma}{\gamma+1} M_0^2 \sin^2 \beta - \frac{\gamma-1}{\gamma+1} \right] \Bigg\} \end{aligned} \tag{4.10}$$

where C_v is the specific heat at constant volume. Figure 4.8 shows the entropy increase across an oblique shock calculated using this expression for various free-stream Mach numbers M_0 with air as the working gas. It is evident that the entropy increase across the shock increases with shock angle, as might be expected. The

'strong' oblique shock solution has a greater shock angle than the 'weak' solution, and hence has a greater associated entropy rise. This can be stated as

$$\Delta s_{strong} > \Delta s_{weak} \tag{4.11}$$

If J is the shape of the reflected shock wave front then the rate of entropy production $\dot{S}$ across the shock is given by[93]

$$\dot{S} = \int_J \rho_0 u_0 \sin\beta (s_1 - s_0) dw \tag{4.12}$$

where dw is a differential line element along J. In this case the shock wave is straight and the entropy increase across the shock does not change along J. Since the total mass flow rate must remain constant (the upstream conditions (0) are not influenced by the reflected shock angle) an increasing entropy jump across the shock implies increasing entropy production. Thus condition (4.11) implies

$$\dot{S}_{strong} > \dot{S}_{weak} \tag{4.13}$$

and by the principle of minimum entropy production the 'weak' solution is the stable i.e. physical solution.

As noted in Salas[100] the principle of minimum entropy production explains the prevalence of 'weak' over 'strong' oblique shock solutions in the simple deflection of supersonic flow, but does not disprove the possibility of a 'strong' shock solution if the downstream pressure is given as a boundary condition. The principle of minimum entropy production applies only when multiple steady states occur which satisfy the same boundary conditions. Thus in this case for a fixed M_0 and θ from oblique shock theory both β_{strong} and β_{weak} could occur but the principle indicates β_{weak}. However if the downstream pressure is given as a boundary condition then the boundary condition set has changed and the only possible solution is some θ which satisfies the pressure.

Regular Reflection

As has been mentioned in Section 4.2.3, two possible RR solutions exist when $\phi_i < \phi_i^D$. This situation is represented in the (p, θ) plane in Figure 4.2(a). The two possible solutions, at the points where the R polar intersects the p axis, arise because

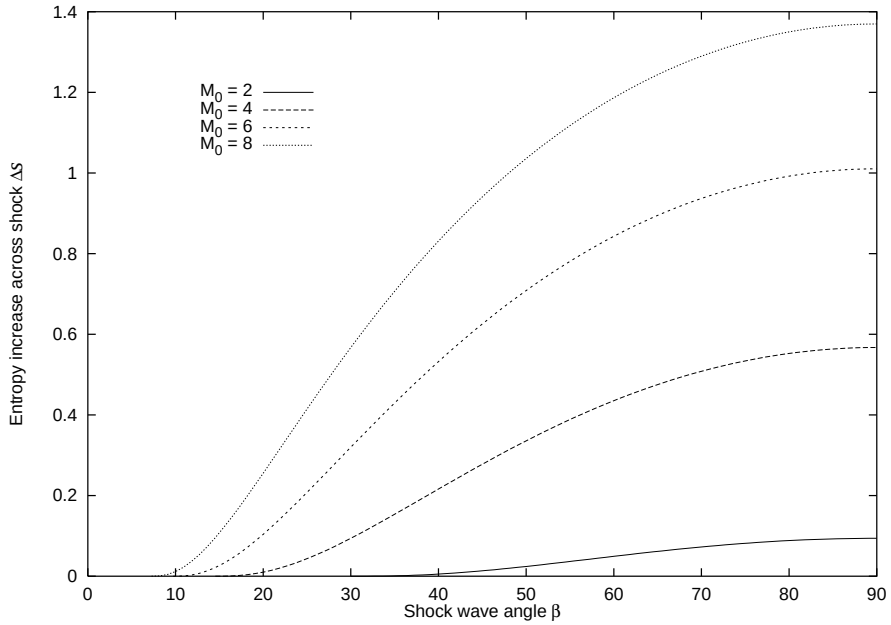


Figure 4.8: *Increase in entropy across an oblique shock*

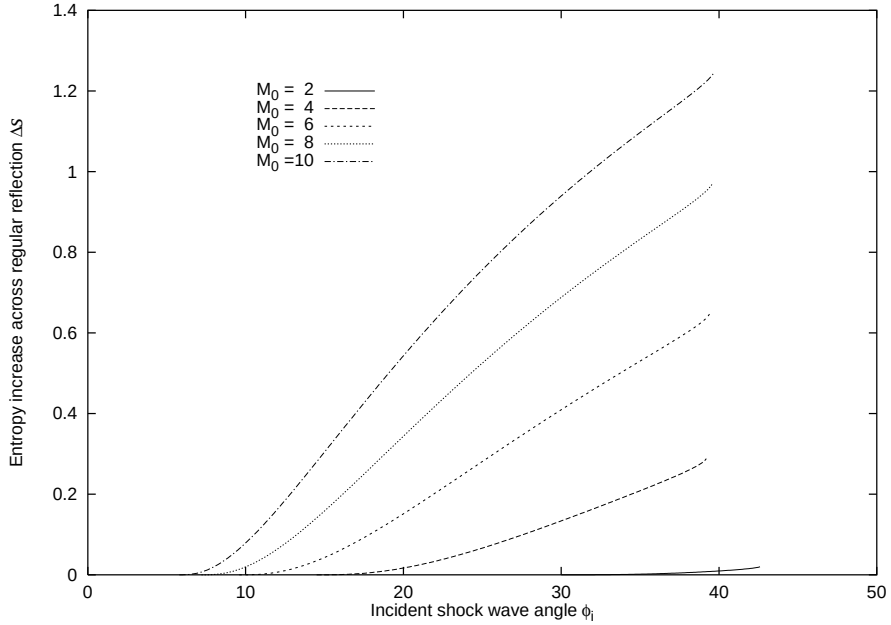


Figure 4.9: *Increase in entropy across a regular reflection*

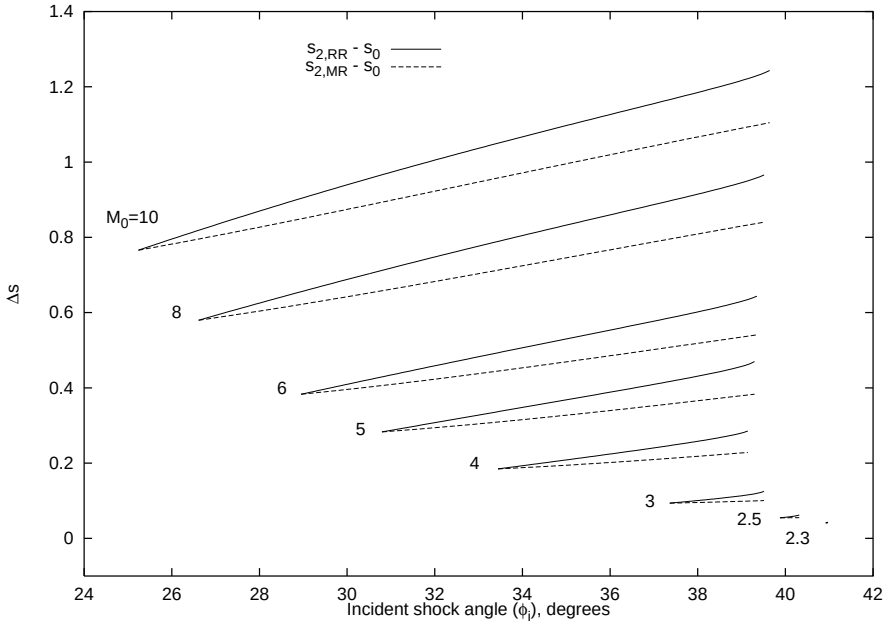


Figure 4.10: *Entropy increase across RR and oblique part of MR in dual solution domain*

two values of reflected shock angle $\phi_{r,weak}$ and $\phi_{r,strong}$ can achieve the θ_r necessary to satisfy the RR boundary condition (4.1). This situation is clearly very similar to supersonic flow deflection (see directly above) and the result is the same; the principle of minimum entropy production predicts that for a given M_0 and ϕ_i the 'weak' solution is observed since it entails a lower rate of entropy production. The entropy increase $\Delta s = s_2 - s_0$ variation with ϕ_i for various M_0 across an RR with a 'weak' reflected shock is shown in Figure 4.9. That the entropy increase in each case would be greater for a 'strong' solution is evident from Figure 4.8 if ϕ_r and M_1 are substituted for β and M_0. Note that the possibility of a 'strong' RR has not been disproved, see the end of Section 4.2.6.

Inverted Mach Reflection

As has been discussed in Section 4.2.3, why the von Neumann condition should mark the lower pressure end of the dual solution domain and IMR is not normally observed has yet to be fully explained. In this and in subsequent sections the conditions in region (3) behind **R** will be denoted with the subscript $_3$ rather than with $_{3,R}$ to shorten the notation. Note that the conditions in region (3) behind **T** are equivalent

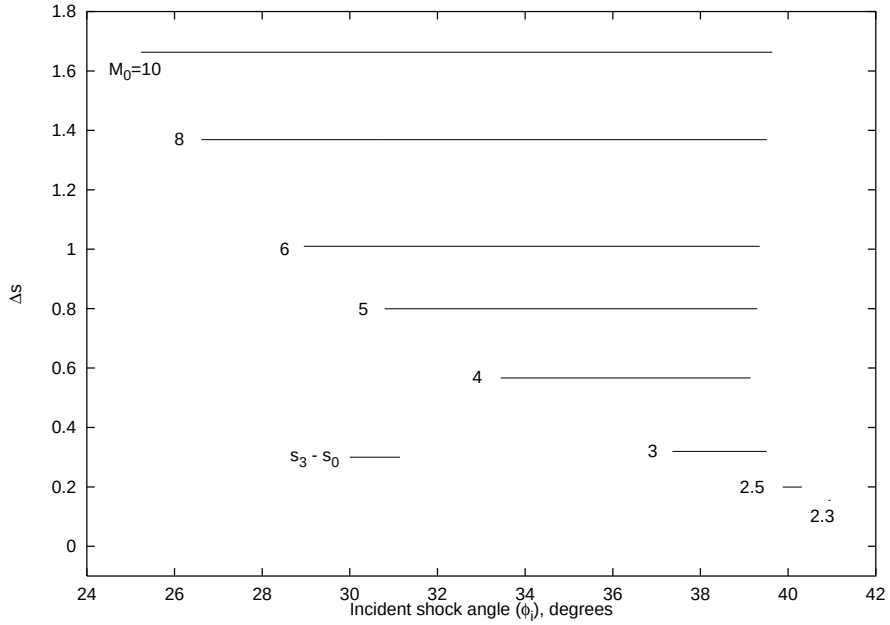


Figure 4.11: *Entropy increase across Mach stem part of MR in dual solution domain*

to those in region (2) behind **r** in the cae of MR so the subscript notation $_{3,T}$ is now discarded. Figures 4.10 and 4.11 show the calculated entropy increases across a regular reflection $(s_{2,RR} - s_0)$ and the oblique $(s_{2,MR} - s_0)$ and Mach stem $(s_3 - s_0)$ parts of a Mach reflection within the dual solution domain for a representative range of Mach numbers M_0. Each curve extends from the von Neumann $(\phi_i = \phi_i^N)$ to the detachment $(\phi_i = \phi_i^D)$ condition. Note that for each M_0, $(s_3 - s_0)$ is greater than both $(s_{2,RR} - s_0)$ and $(s_{2,MR} - s_0)$. Note also that at the von Neumann condition $(s_{2,RR} - s_0)$ and $(s_{2,MR} - s_0)$ are identical (because the Mach stem has vanished) and that $(s_{2,RR} - s_0)$ increases more sharply with incident shock angle i.e.

$$\frac{d\left(s_{2,RR} - s_0\right)}{d\phi_i} \quad > \quad \frac{d\left(s_{2,MR} - s_0\right)}{d\phi_i}$$

For $\phi_i < \phi_i^N$ (i.e. where an IMR is theoretically possible) this trend continues since $\phi_{r,MR}$ must be greater than the corresponding $\phi_{r,RR}$ in order to achieve the negative flow deflection. As a consequence for $\phi_i < \phi_i^N$ the entropy increase across both parts of the MR is greater than that across the corresponding RR. This can be written as

$$\begin{aligned} \left(s_3 - s_0\right)^{\phi_i < \phi_i^N} \quad &> \quad \left(s_{2,RR} - s_0\right)^{\phi_i < \phi_i^N} \\ \left(s_{2,MR} - s_0\right)^{\phi_i < \phi_i^N} \quad &> \quad \left(s_{2,RR} - s_0\right)^{\phi_i < \phi_i^N} \end{aligned} \tag{4.14}$$

The expression (4.12) for the rate of entropy production for an oblique shock is adapted here for the entropy production rates for regular and Mach reflection

$$\dot{S}_{RR} = \int_{J_{r,RR}} \rho_{2,RR} u_{2,RR} \sin\left(\phi_{r,RR} - \theta_{r,RR}\right) (s_{2,RR} - s_0) dw_{r,RR} \qquad (4.15)$$

$$\dot{S}_{MR} = \int_{J_{r,MR}} \rho_{2,MR} u_{2,MR} \sin\left(\phi_{r,MR} - \theta_{r,MR}\right) (s_{2,MR} - s_0) dw_{r,MR} + \int_{J_m} \rho_0 u_0 (s_3 - s_0) dw_m \qquad (4.16)$$

where $J_{r,RR}$, $J_{r,MR}$, J_m are the shapes of $\mathbf{r}$ (in RR and MR) and $\mathbf{m}$ respectively; $dw_{r,RR}$, $dw_{r,MR}$ and dw_m are their respective differential line elements. Note that in these expressions all sources of entropy production other than the incident shock, reflected shock and Mach stem are neglected. The total mass flow rate for a given M_0 and ϕ_i must be equal for RR and MR. However, not all of the flow is processed by the shock system (see Figure 4.6). We make the assumption that the same amount of flow is processed by our simplified models of the RR and MR systems. This is exactly true at the von Neumann condition and appears to be a good approximation in the vicinity of this condition from flow visualisations[85],[86],[92]. This assumption provides a continuity equation

$$\int_{J_{r,RR}} \rho_{2,RR} u_{2,RR} \sin\left(\phi_{r,RR} - \theta_{r,RR}\right) dw_{r,RR} = \int_{J_{r,MR}} \rho_{2,MR} u_{2,MR} \sin\left(\phi_{r,MR} - \theta_{r,MR}\right) dw_{r,MR} + \int_{J_m} \rho_0 u_0 dw_m \qquad (4.17)$$

From equations (4.15) and (4.16) the entropy production rate is the product of mass flow rate and entropy increase integrated across the shock system. Since the entropy increase across both parts of the IMR is greater than across the RR (see the inequalities (4.14)) and the total mass flow rate is the same in each case (see equation (4.17)), then we can conclude that the IMR entropy production rate is greater than the RR, regardless of the proportions of the total mass flow being processed by by the MR components. This is stated as

$$\left(\dot{S}_{MR}\right)^{\phi_i < \phi_i^N} > \left(\dot{S}_{RR}\right)^{\phi_i < \phi_i^N}$$

Hence by the principle of minimum entropy production RR is the observed solution.

The possibility of suppressing the RR to obtain the IMR does not violate this result for the same reasons as discussed in Section 4.2.6. In the present case for a fixed M_0 and ϕ_i both IMR and RR could occur from oblique shock theory but the principle selects RR. However if the downstream pressure is given as a boundary condition then the set of boundary conditions has changed and the principle does not apply. According to Hornung[92], if RR is impeded by setting the downstream pressure to a high value then IMR may occur, explaining the results of Henderson[94] where IMR was observed.

Shock Reflection Hysteresis

As discussed in Section 4.2.5, the persistence of MR in the dual solution domain i.e. for one half of the hysteresis loop has yet to be fully explained. In this section it will be argued that the MR may have a lower entropy production rate than the RR, and hence by the principle of minimum entropy production MR is the observed solution.

Figures 4.10 and 4.11 show the calculated entropy increases across a regular reflection ($s_{2,RR} - s_0$) and the oblique ($s_{2,MR} - s_0$) and Mach stem ($s_3 - s_0$) parts of a Mach reflection within the dual solution domain. From these figures it is evident that within the dual solution domain the entropy increase across the oblique part of the MR is less than that across the RR, but the entropy increase across the Mach stem is greater. This can be written as

$$\begin{aligned} \left(s_3 - s_0\right)^{\phi_i^N < \phi_i < \phi_i^D} &> \left(s_{2,RR} - s_0\right)^{\phi_i^N < \phi_i < \phi_i^D} \\ \left(s_{2,MR} - s_0\right)^{\phi_i^N < \phi_i < \phi_i^D} &< \left(s_{2,RR} - s_0\right)^{\phi_i^N < \phi_i < \phi_i^D} \end{aligned} \tag{4.18}$$

We would like to compare RR and MR entropy production rates. In this case to do this we must also examine the relative proportions of mass flow processed by each MR component. Expressions for $\dot{S}_{RR}$ and $\dot{S}_{MR}$, the rates of entropy production for RR and MR respectively, are given by equations 4.15 and 4.16. If we make the further assumption that these shocks are straight then the expressions simplify to

$$\dot{S}_{RR} = \rho_{2,RR} u_{2,RR} \sin\left(\phi_{r,RR} - \theta_{r,RR}\right) \left(s_{2,RR} - s_0\right) l_{r,RR} \tag{4.19}$$

$$\dot{S}_{MR} = \rho_{2,MR} u_{2,MR} \sin\left(\phi_{r,MR} - \theta_{r,MR}\right)\left(s_{2,MR} - s_0\right) l_{r,MR} + \rho_0 u_0 \left(s_3 - s_0\right) l_m \tag{4.20}$$

where $l_{r,RR}$, $l_{r,MR}$, and l_m are the lengths of **r** (in RR and MR), and **m** respectively. To help us examine this, we define **A**, **B** and **C** as follows

$$\begin{aligned} \mathbf{A} &= \rho_{2,RR} u_{2,RR} \sin\left(\phi_{r,RR} - \theta_{r,RR}\right)\left(s_{2,RR} - s_0\right) \\ \mathbf{B} &= \rho_0 u_0 \left(s_3 - s_0\right) \\ \mathbf{C} &= \rho_{2,MR} u_{2,MR} \sin\left(\phi_{r,MR} - \theta_{r,MR}\right)\left(s_{2,MR} - s_0\right) \end{aligned} \tag{4.21}$$

then equations (4.15) and (4.16) become respectively

$$\begin{aligned} \dot{S}_{RR} &= \mathbf{A} l_{r,RR} \\ \dot{S}_{MR} &= \mathbf{B} l_m + \mathbf{C} l_{r,MR} \end{aligned} \tag{4.22}$$

Figure 4.12 shows the variation of the parameters **A**, **B** and **C** with ϕ_i within the dual solution domain for a free-stream Mach number $M_0 = 4.96$ for which $\phi_i^N = 30.9^o$. The significance of this particular condition is explained later. As illustrated in Figure 4.7 a feature of the von Neumann condition is that the length of the Mach stem **m** has become vanishingly small; the RR and MR are effectively identical ($l_m = 0, \phi_{r,RR} = \phi_{r,MR}$). This can be seen in Figure 4.12 where $\mathbf{A}(\phi_i^N) = \mathbf{C}(\phi_i^N)$ as a consequence. On increasing ϕ_i, **A** increases more quickly than **C**. Within the dual solution domain the principle of minimum entropy production has the potential for allowing selection of the prevailing reflection type. However, as is clear from equations (4.19) and (4.20), knowledge of the shock wave lengths as well as the quantities **A**,**B**,**C** is required in order to make a direct comparison between $\dot{S}_{RR}$ and $\dot{S}_{MR}$. Introducing a relationship between the total mass flow rates through the RR and MR aids clarification. Assuming that the mass flow through the RR is the same as that through the MR and that the shocks are straight, equations (4.19) and (4.20) become

$$\rho_{2,RR} u_{2,RR} \sin\left(\phi_{r,RR} - \theta_{r,RR}\right) l_{r,RR} = \rho_0 u_0 l_m + \rho_{2,MR} u_{2,MR} \sin\left(\phi_{r,MR} - \theta_{r,MR}\right) l_{r,MR} \tag{4.23}$$

We want to show that MR is the observed solution by the principle of minimum entropy production i.e.

$$\dot{S}_{RR} > \dot{S}_{MR} \tag{4.24}$$

Combining this condition with equations (4.21),(4.22) and (4.23) gives a geometric condition for the prevalence of MR in the dual solution domain in terms of the gasdynamic conditions

$$\frac{l_m}{l_{r,RR}} < \frac{\rho_{2,RR} u_{2,RR} \sin\left(\phi_{r,RR} - \theta_{r,RR}\right)\left(s_{2,RR} - s_{2,MR}\right)}{\rho_0 u_0 \left(s_3 - s_{2,MR}\right)} \tag{4.25}$$

It is not possible to evaluate the LHS of this expression using the present analysis. Experimental results for shock reflection hysteresis at $M_0 = 4.96$ have been presented in [86]. The same problem has also been studied numerically[87],[88],[98]. From these results it is possible to extract a value for $l_m/l_{r,RR}$ within the dual solution domain and test the condition (4.25). However, it is useful to recap on how this expression was obtained. In particular, two important simplifying assumptions have been made:

The total mass flow rate through the RR and MR are equivalent. As shown in Figure 4.6 the expansion around the trailing edge of the wedge interacts with the reflected shock wave. Some of the flow which is processed by the incident shock is not processed by the reflected shock. As the Mach stem grows the inclination and length of the reflected oblique shock changes. As a result, the proportion of the flow processed by the reflected oblique shock is different for RR and MR.

Sources of entropy production other than shock waves are neglected. The entropy production due to the interacton of the reflected shock with the expansion is assumed to be equivalent for RR and MR i.e. does not influence comparisons of entropy production rates.

The errors associated with these assumptions increase on departing from the von Neumann condition. They are difficult to quantify; from flow visualisations[85],[86],[92] the lengths of the reflected shocks in the dual solution domain do not appear to differ

greatly for RR and MR but this does not provide sufficient justification for identifying some range of ϕ_i in the vicinity of ϕ_i^N for which condition (4.25) is valid, even before taking into account errors associated with measuring l_m and $l_{r,RR}$. It is therefore difficult to investigate whether the condition (4.24) is true for even one particular set of conditions.

Directly comparing $\dot{S}_{RR}$ and $\dot{S}_{MR}$ on a reliable basis using the present approach is therefore not possible. However, it is still possible to make use of the fact that our assumptions are exact at the von Neumann condition. If condition (4.24) is true for the dual solution domain then the condition

$$\left(\frac{d\dot{S}_{RR}}{d\phi_i}\right)^{\phi_i=\phi_i^N} > \left(\frac{d\dot{S}_{MR}}{d\phi_i}\right)^{\phi_i=\phi_i^N} \tag{4.26}$$

must also hold since

$$\left(\dot{S}_{RR}\right)^{\phi_i=\phi_i^N} = \left(\dot{S}_{MR}\right)^{\phi_i=\phi_i^N} \tag{4.27}$$

Noting that for MR any increase in the mass flow through the Mach stem due to its growth entails a corresponding reduction in the mass flow through the reflected shock i.e.

$$\frac{d}{d\phi_i}(\rho_{2,MR}u_{2,MR}\sin(\phi_{r,MR}-\theta_{r,MR})\,l_{r,MR}) + \frac{d}{d\phi_i}(\rho_0 u_0 l_m) = 0 \tag{4.28}$$

then after some manipulation equations (4.19),(4.20) and (4.26) yield

$$\frac{d}{d\phi_i}(s_{2,RR}-s_0) > \frac{d}{d\phi_i}(s_{2,MR}-s_0) + \frac{\rho_0 u_0(s_3-s_{2,MR})}{\rho_{2,RR}u_{2,RR}\sin(\phi_{r,MR}-\theta_{r,MR})l_{r,RR}}\frac{dl_m}{d\phi_i} \tag{4.29}$$

which is valid only at the von Neumann condition. If h_{in} is the distance between the leading edges of the two symmetric wedges ($h_{in}/2$ is then the distance from the wedge leading edge to the symmetric line/reflecting surface in Figure 4.6) then we have simply

$$\frac{dl_m}{d\phi_i} = \frac{h_{in}}{2}\frac{d(\frac{l_m}{h_{in}/2})}{d\phi_i}$$

Equation (4.29) then becomes

$$\frac{d}{d\phi_i}(s_{2,RR}-s_0) > \frac{d}{d\phi_i}(s_{2,MR}-s_0) + \frac{\rho_0 u_0(s_3-s_{2,MR})(h_{in}/2)}{\rho_{2,RR}u_{2,RR}\sin(\phi_{r,MR}-\theta_{r,MR})l_{r,RR}}\frac{d(\frac{l_m}{h_{in}/2})}{d\phi_i} \tag{4.30}$$

The quantity $\rho_0 u_0 (h_{in}/2)$ is the total mass flow rate between the wedge and the symmetric line. The quantity $\rho_{2,RR} u_{2,RR} \sin(\phi_{r,MR} - \theta_{r,MR}) l_{r,RR}$ is the total mass flow rate across the reflected oblique shock. Their ratio, which we denote $ratio_{flow}$, can be estimated from experiment along with the quantity $d(\frac{l_m}{h_{in}/2})/d\phi_i$. All other quantities in this expression can be obtained from our simplified analysis. The quantities pertaining to condition (4.30) for the von Neumann condition $M_0 = 4.96$, $\phi_i^N = 30.9^o$ are summarised in Table 4.1. Note that the necessary geometric values extracted from experimental data are deliberately estimated on the side of a high RHS to this condition in order to avoid uncertainty in this conclusion[1,2]. From this analysis the condition (4.30) is demonstrated to be true, and therefore condition (4.24) is true in the vicinity of the von Neumann condition for $M_0 = 4.96$.

Note that for the experiment referred to here the dual solution domain was found to terminate a few degrees before the detachment condition. It is possible that at this point the condition (4.24) is no longer true. An observation from [86] which supports this is that $d(\frac{l_m}{h_{in}/2})/d\phi_i$ increases on departing from the von Neumann condition. The present simplified approach is not adequate to conclusively demonstrate whether the principle of minimum entropy production can explain the persistence of MR as part of the hysteresis loop. However, the above result can at least be regarded as evidence which supports this idea. A conclusive analysis would require at least accurate experimental measurement of shock wave lengths. An alternative approach could be to directly measure the entropy production from numerical results.

[1]This estimate is taken from the flow visualisations in [86]. From these images between 40% and 50% of the total mass flow is processed by the reflected oblique shock. Here a value of 30% is assumed to ensure a valid conclusion.

[2]Estimated from the straight line part of Figure 10(a) in [86]. Using the least-squares fit would yield a smaller value.

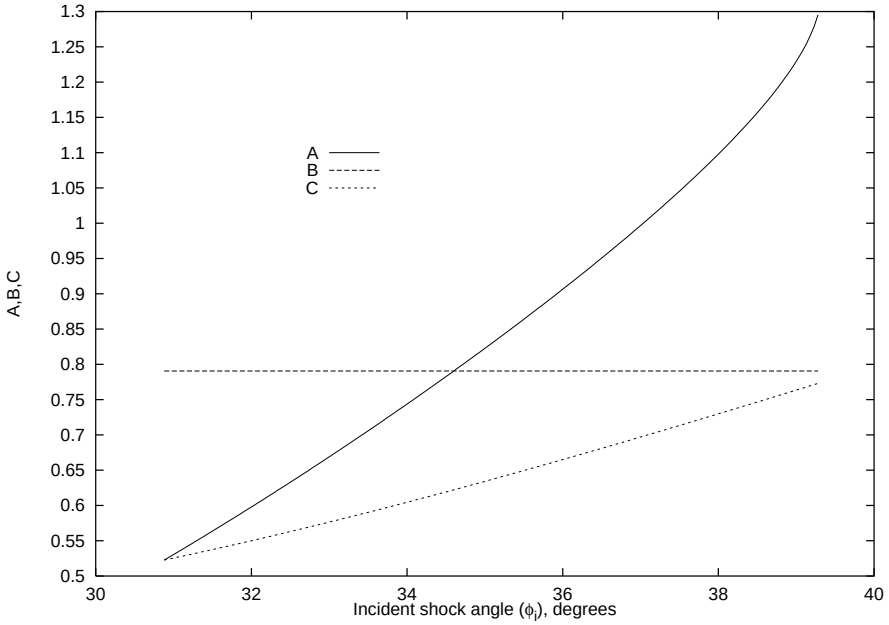


Figure 4.12: *Variation of parameters* **A**,**B**,**C** *in dual solution domain for* $M_0 = 4.96$

$\frac{d}{d\phi_i}(s_{2,RR} - s_0)$	1.2478
$\frac{d}{d\phi_i}(s_{2,MR} - s_0)$	0.6675
$(s_3 - s_{2,MR})$	0.5116
$ratio_{flow}$	3.3333[1]
$\frac{d}{d\phi_i}(\frac{l_m}{h_{in}/2})$	0.2843[2]
LHS of (4.30)	1.248
RHS of (4.30)	1.152

Table 4.1: *Values for condition (4.30) at* $M_0 = 4.96$, $\phi_i^N = 30.9^o$

4.2.7 Discussion

The principle of minimum entropy production has been applied to some shock wave and shock reflection phenomena in an attempt to explain experimental observations.

First, the conclusions of other authors who examined supersonic flow deflection and regular shock reflection were restated to establish that a precedent for this type of approach exists. It was then demonstrated how the principle can be used to explain why the well known von Neumann criterion marks one limit of the shock reflection dual solution domain and Inverted Mach Reflection is not normally observed. It is then suggested that the phenomenon of shock reflection hysteresis has yet to be fully explained; that the principle of minimum entropy production may provide an explanation for the persistence of Mach reflection when the dual solution domain is approached from a Mach reflection condition is proposed. The difficulties associated with applying the principle to this problem are discussed. Some evidence supporting this theory is presented for one particular hysteresis case for which experimental data is available. Some recommendations are made for continued study of this problem.

4.3 Numerical Method

4.3.1 Flow Solver

In this section, it is described how an axisymmetric (laminar) Navier-Stokes flow solver has been used to study the phenomenon of shock reflection hysteresis in an underexpanded jet. The results of this study will be examined in the light of the understanding of the two-dimensional hysteresis phenomenon established in section 4.2. The flow solver used is described in Chapter 3.

For this study it is assumed throughout that the working gas is in the continuum regime with no condensation and has constant specific heats. These assumptions are verified in a straightforward manner. The extremities of pressure, temperature etc. experienced in the experiments can be obtained from [71]. The Knusden number based on the shock cell length was calculated [103] [104] as being less than 0.15 at all times. The continuum Navier-Stokes equations hold up to Knusden numbers of 0.2 [105] so we are just within the continuum regime. Based on the experiments reported in [106] the present cases are also condensation free. Despite the high Mach numbers and strong shock waves encountered in the experiments the gas temperature remained relatively low at all times, well below the levels where molecular

dissociation or vibrational excitation become important [105]. These assumptions are confirmed by the Hypersonic Aerodynamics Group at DERA Farnborough where the experiments were carried out.

4.3.2 Boundary Conditions

Figure 4.13 shows a diagram of the computational domain with labelled boundary condition types (the size of the nozzle is exaggerated for clarity). The boundaries labelled A denote adiabatic wall boundaries with no slip and zero normal pressure gradient. At B a symmetry condition was applied. To decide which boundary conditions to apply at C we have the advantage that across all of this boundary we know that we should have outflow. In keeping with an inviscid characteristic analysis, the flow variables are extrapolated from the interior of the domain except for the case of locally subsonic outflow where the pressure was imposed at the background level. Two alternative treatments were tested and rejected. First, all of the flow variables were extrapolated across all of the boundary. This significantly impaired convergence in regions of subsonic outflow. Secondly, the background pressure was imposed across all of the boundary. This did not impair convergence but did result in significant spurious oscillations in regions of supersonic outflow. The boundary condition treatment at the nozzle inlet D requires a somewhat more involved treatment. We know the reservoir stagnation conditions (denoted here by r) but require boundary conditions for the nozzle inlet i. This is achieved by assuming that the total enthalpy and entropy are the same for the reservoir and nozzle inlet, thus obtaining expressions for p_i and ρ_i which are imposed. The velocity components are extrapolated from the interior of the domain. Note that assuming constant entropy s implies a constant entropy measure S defined by

$$S = \frac{p}{\rho^\gamma}$$

For convenience the non-dimensionalisation is constructed such that

$$\rho_r = 1, \qquad p_r = \frac{1}{\gamma} \tag{4.31}$$

are the values of density and pressure respectively in the reservoir. The reservoir sound speed, total enthalpy and the entropy measure are then

$$a_r = 1, \qquad H_r = \frac{1}{(\gamma - 1)}, \qquad S_r = \frac{1}{\gamma}$$

We now have two conditions to impose at the inlet i :

$$H_r = \frac{\gamma p_i}{(\gamma - 1)\,\rho_i} + \frac{(v_r)_i^2 + (v_z)_i^2}{2}, \qquad S_r = \frac{p_i}{\rho_i^\gamma}$$

The velocity components u_i and v_i are extrapolated from the interior of the domain. The inlet density and pressure are then

$$\rho_i = \left[1 - \frac{(v_r)_i^2 + (v_z)_i^2}{2}\,(\gamma - 1))\right], \qquad p_i = \frac{1}{\gamma}\rho_i^\gamma$$

For the boundary condition treatment at E the background stagnation conditions are known but the flow direction is not known *a priori*. This being similar to the nozzle inlet boundary treament, the boundary conditions here are treated in the same manner except that background conditions are used instead of reservoir conditions in equation 4.31.

It is possible to construct a Riemann invariant based boundary condition treatment for the boundaries at C and E analogous to that commonly used as a "far-field" condition for aerofoil flow studies. However the present approach was found to be adequate and its implementation straightforward in the framework of the existing flow solver. One more boundary condition was used; as described below, calculations were also carried out concerning only the nozzle flow. At the nozzle outlet, where the flow is supersonic except in the boundary layer, all flow variables were extrapolated from the interior of the domain.

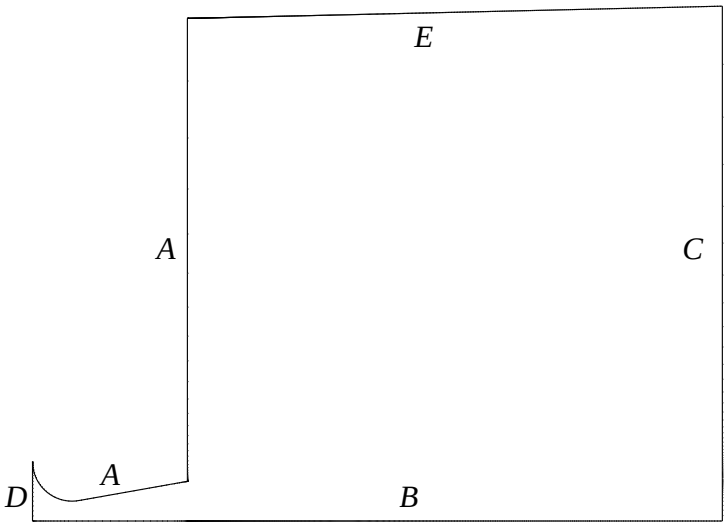


Figure 4.13: *Boundary conditions*

4.3.3 Initial Conditions and Quasi-Steady Approach

As a first step to studying the full problem a preliminary calculation for the nozzle only was carried out. Using a linear variation from reservoir to Mach 3 conditions along the axis from the inlet to the outlet as initial conditions was found to considerably reduce the calculation time compared with using uniform reservoir or sonic conditions. The solution from this calculation was used as the initial solution in the nozzle for the main calculation, and the calculated nozzle exit conditions were used as the initial conditions for the domain directly downstream of the nozzle exit. For the remainder of the domain the background conditions were applied as initial conditions. Calculations were performed over a range of pressure ratios from well inside the regular reflection range to well inside the Mach reflection range including the hysteresis loop. A quasi-steady approach was employed in order to account for time history effects. First, converged solutions were obtained for the conditions at the extremities of the range of interest. These were used as initial solutions for a calculation with a small change in pressure ratio, thus beginning to traverse the

range, this solution being used subsequently as the next initial solution etc. By using a small step change in pressure ratio between calculations this approach is very robust and converges quickly at each condition, as shown in Figure 4.14. Here it should be noted that the scaling residual used in the figure, the residual after the first step, is already small. A reduction of just over two orders of magnitude in the residual was found to be sufficient for the step size used. Further convergence did not alter the solution. This usually required around 100 steps to achieve[1] but may require up to 500 steps when a switch in shock reflection type occurs. By contrast, obtaining a converged solution (without the aid of a close initial solution) for the end points of the pressure ratio range is far more demanding, requiring approximately 30 times the computational effort. The step change in pressure ratio used is 2.857, corresponding to a step change in reservoir stagnation pressure of 0.1 *torr* for a background pressure of 35 *mtorr* in terms of the original experiments.

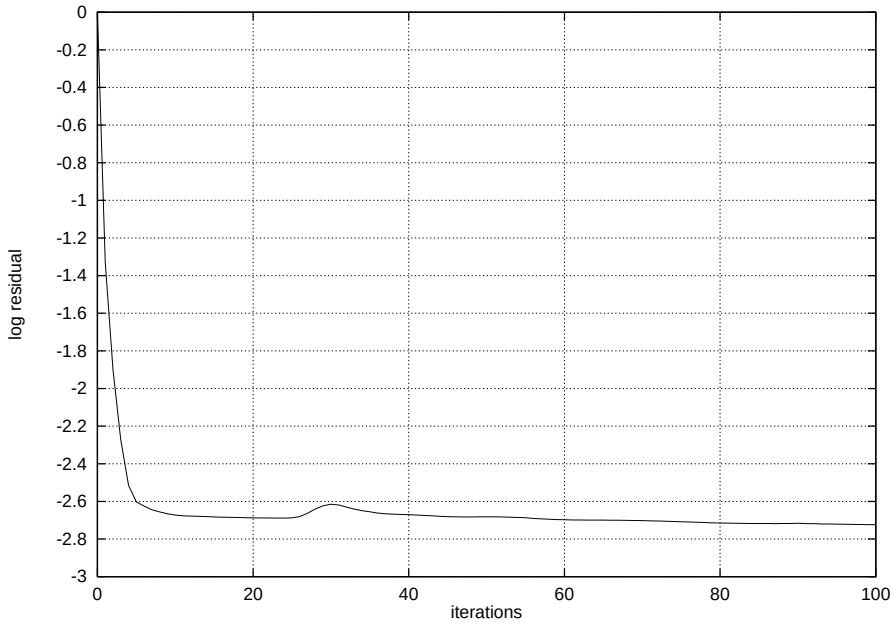


Figure 4.14: *Convergence behaviour at each step*

4.3.4 Grid

The grid generation for this case is straightforward due to the simple geometry. The computational domain extends 70 nozzle throat diameters downstream in order to

[1] corresponding to a CPU time of 950 seconds on a 200MHz Intel Pentium Pro processor

capture at least two shock cells and 20 diameters radially from the symmetric line. The grid within the nozzle consists of 58 and 21 points in the axial and radial directions respectively, this number having been determined from a grid convergence study carried out independently of the plume calculations. The plume calculations were also performed using a number of grids. The effect of the radial extent of the computational domain was examined by comparing results for calculations with grids extending 20 and 40 nozzle throat diameters from the symmetric line. The results are identical, see for example Figure 4.15. It was also found that the grid density in the radial direction can be surprisingly coarse compared to the axial direction, 65 points being sufficient. In order to obtain a grid converged solution, the necessary axial grid density was much finer. Results were obtained for three different levels of axial grid fineness, see Table 4.2. Using the grid convergence measure of the calculated limits of the hysteresis loop, a grid converged solution was obtained using 937 points in the axial direction. Centre-line values were also checked for grid convergence, see Figure 4.16. For any given pressure ratio the grid is excessively fine in places, but since the location of the shock reflections vary widely with pressure ratio and the same grid was used in each case this was unavoidable.

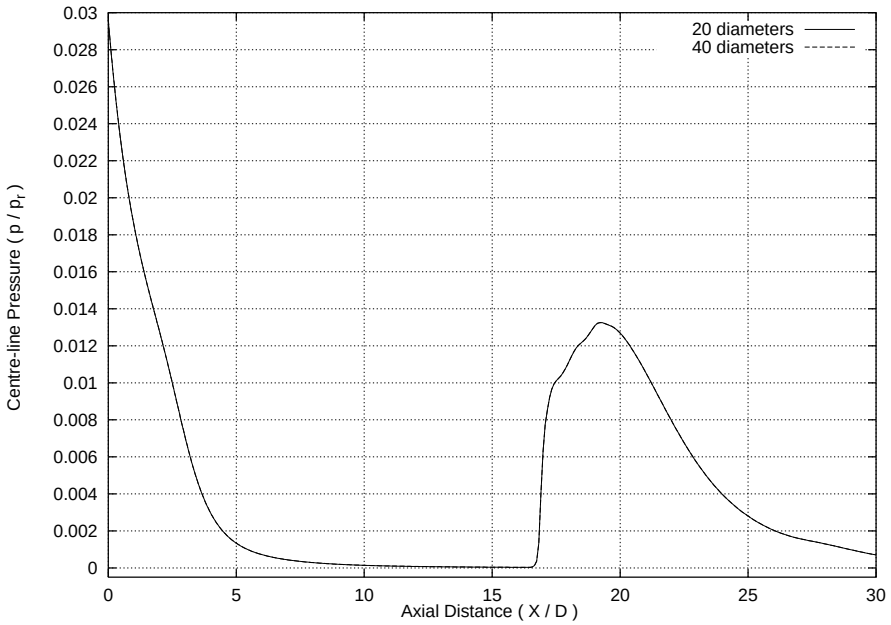


Figure 4.15: *Effect of radial extent of domain for* $p_r/p_b = 285.7$

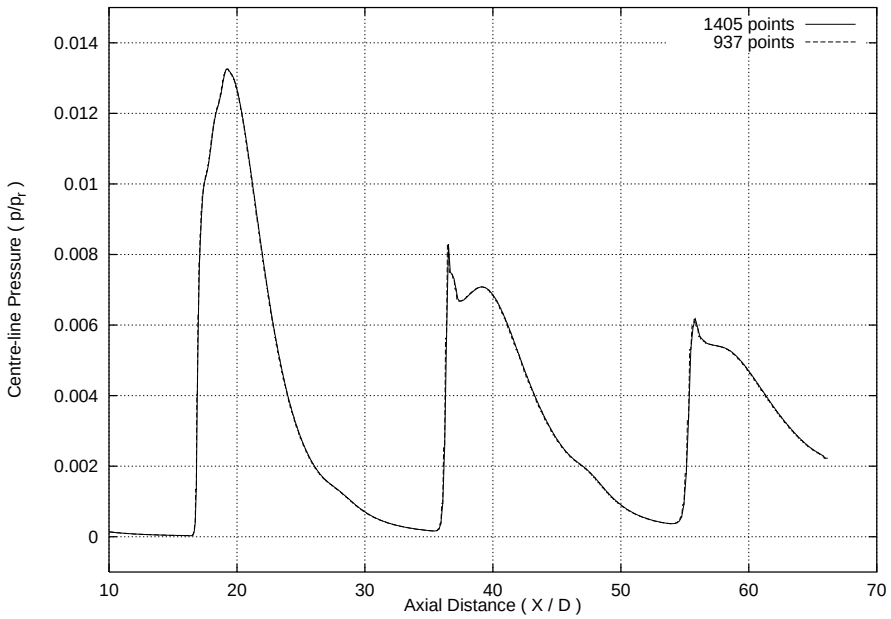


Figure 4.16: *Result of grid convergence study for* $p_r/p_b = 285.7$

No. points along axis	*Lower limit*[1]		*Upper limit*[2]	
	p_r/p_b	X/D	p_r/p_b	X/D
469	200.0 || 202.9	14.10 || 13.14	434.3 || 437.1	21.02 || 18.81
937	214.3 || 217.1	14.55 || 13.41	334.3 || 337.1	18.32 || 16.50
1405	214.3 || 217.1	14.55 || 13.41	334.3 || 337.1	18.32 || 16.50

Table 4.2: *Grid independence study for hysteresis loop*

4.4 Results

4.4.1 Nozzle Calculations

Welsh[71] describes a series of experiments where the effect of varying the ratio of reservoir stagnation pressure p_r to background pressure p_b on the plume of a highly underexpanded nitrogen jet is examined. The reported shock reflection hysteresis phenomenon provided the motivation for this CFD study. In the experiments p_r

[1]first value is highest pressure ratio not on MR curve, second is lowest ratio in loop

[2]first value is highest pressure ratio in loop, second is lowest ratio not on RR curve

was varied and p_b was kept constant. In this way the nozzle exit conditions as well as the pressure ratio were varied. In addition, experiments were carried out for a number of values of p_b and for two different nozzle sizes. It is difficult then to isolate the effect of the varying pressure ratio. For these reasons, as a preliminary to the nozzle-plume study, it is useful to perform calculations for the nozzle alone in order to examine the effect of the nozzle Reynolds number on the nozzle exit conditions. This will help to put subsequent nozzle-plume calculations and comparison with experiment in their proper context.

Calculations were performed for a range of reservoir stagnation pressures, from 2 *torr* to 70 *torr*, covering the range used in the experiments. The reservoir stagnation temperature T_o is constant at 288.0K and the two throat diameters used are 5.19 mm and 15.3 mm. The Reynolds number based on throat conditions Re can then be calculated for each p_r, assuming sonic conditions at the throat, using the isentropic relations and Sutherland's law for viscosity. The variation of Re with pressure ratio for each nozzle is shown in Figure 4.17. The present CFD method was then used to obtain results for the nozzle flow for a range of Re.

The calculated variation of the maximum nozzle exit Mach number M_{exit} with throat Reynolds number Re is shown in Figure 4.18. The crosses in the figure, which have been joined by straight lines, indicate the twenty calculation points. Calculated pressure contours for $Re = 800$ are shown in Figure 4.19. As expected, as Re decreases so does M_{exit}, caused by the displacement effect of the boundary layer decreasing the effective area of the divergent section of the nozzle. The thickness of the boundary layer can be visualised from the Mach number plot Figure 4.20. The magnitude of the trend confirms that for each of the experiments the nozzle exit conditions vary significantly.

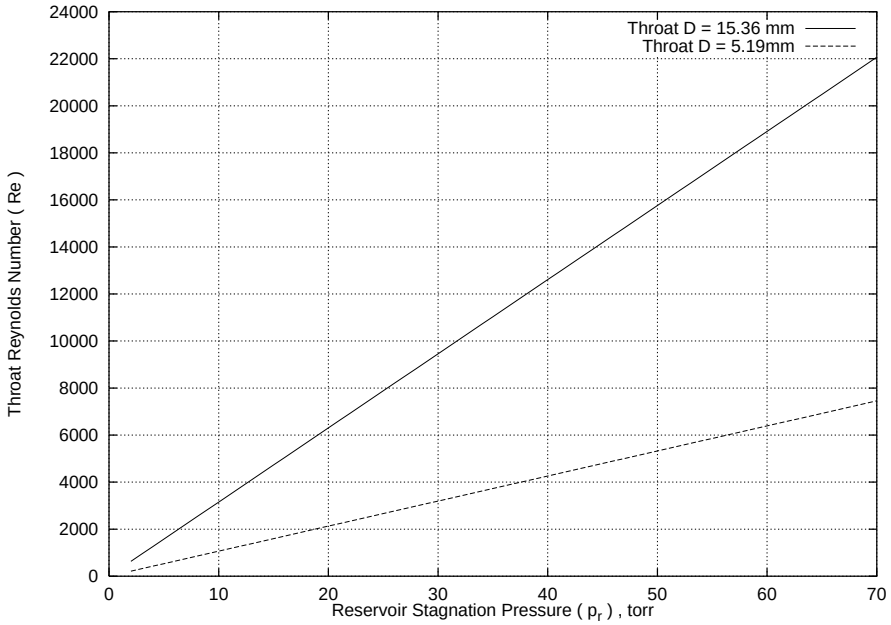


Figure 4.17: *Reynolds number variation with stagnation pressure*

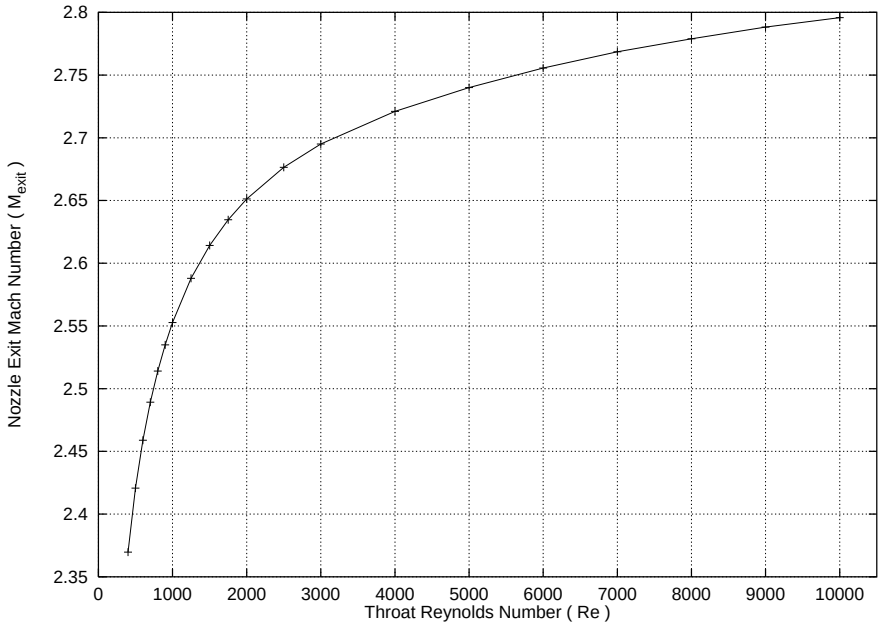


Figure 4.18: *Effect of throat Reynolds number on maximum nozzle exit Mach number*

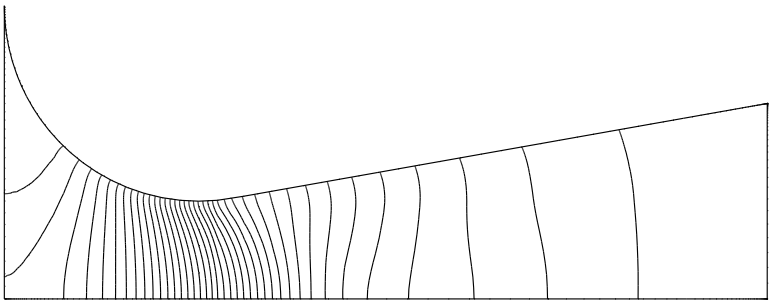


Figure 4.19: *Pressure contours, nozzle calculation, Re = 800.0*

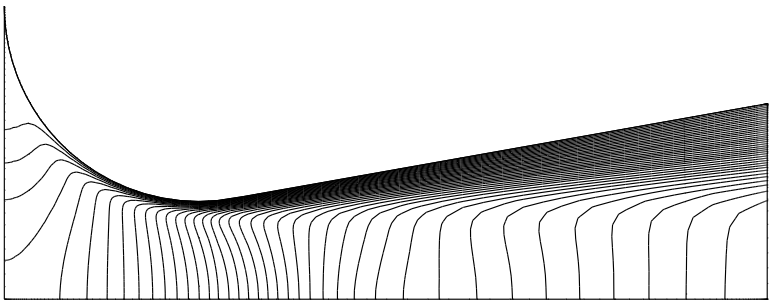


Figure 4.20: *Mach number contours, nozzle calculation, Re = 800.0*

4.4.2 Hysteresis Loop

The calculated shock reflection type and distance variation with pressure ratio is shown in Figure 4.21. The Reynolds number was kept constant, as discussed below. The shock reflection distance is the axial distance (X_{sr}) from the nozzle exit to the centre of the first centre-line regular reflection or Mach disc, non-dimensionalised with respect to the nozzle throat diameter (D). The figure shows how for a small range of pressure ratios either regular reflection (RR) or Mach reflection (MR) may occur. Which condition prevails depends on the time history of the plume development, in accordance with experimental observation. Selecting, for example, the pressure ratio value of 300 in Figure 4.21, the corresponding point on the RR(MR) curve will be reached if the condition immediately prior was also on the RR(MR) curve. The arrows on the figure indicate the "flip" in reflection type which occurs at the limits of the hysteresis loop. From this figure it can be concluded that the quasi-steady approach (section 4.3.3) has been successful, at least qualitatively, in modelling the shock reflection hysteresis phenomenon. A description of the plume structures associated with RR and MR is included in section 4.4.3.

Figure 4.22 shows the extent of the calculated hysteresis loop compared with the data from experiments[71]. The scatter in the experimental results should be explained. Each experiment was carried out with a constant background pressure and varying reservoir stagnation pressure, with the result that the nozzle Reynolds number is not constant. This is illustrated in Figure 4.17 which shows the effect of the varying stagnation pressure on the throat Reynolds number for the nozzles used in the experiment. The effect of a varying nozzle Reynolds number on the maximum exit Mach number for these nozzles is shown in Figure 4.18. Thus for each of the experiments it is difficult to examine the effect of a varying pressure ratio when the Reynolds number and nozzle exit conditions are not constant.

With this is mind, the present calculations were performed with a constant throat Reynolds number of 4000 and a varying pressure ratio, which in effect models varying background pressure and constant nozzle conditions, thus enabling examination of the pressure ratio influence independently. Although we cannot expect close agreement with the experiments for this reason, we can at least conclude from Figure

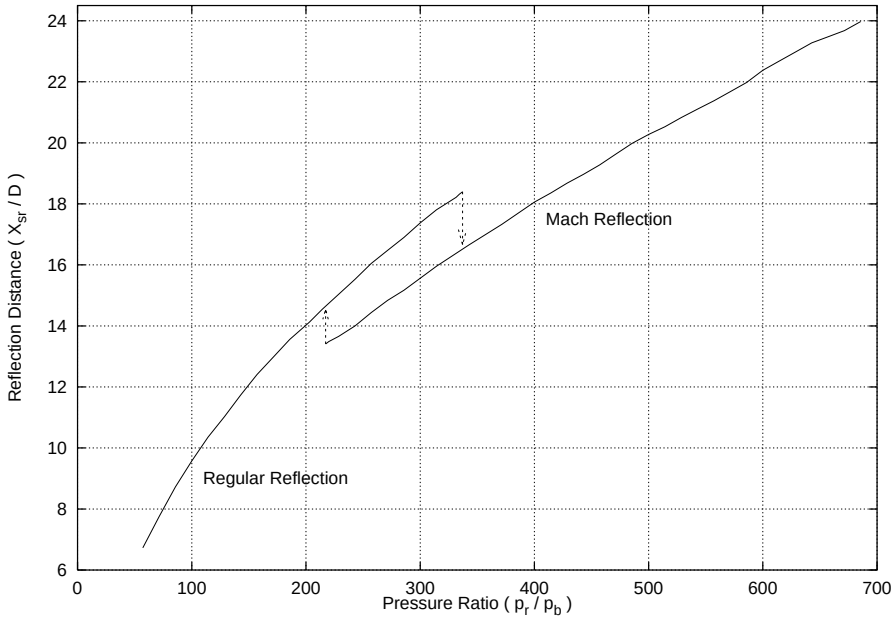


Figure 4.21: *Distance to reflection for range of pressure ratios, Re = 4000*

4.22 that our present calculation of the location of the hysteresis loop is reasonable in terms of both pressure ratio and reflection distance. Comparison will be made in the remainder of this chapter with the experimental conditions where the value of Re in the reported dual solution domain is closest to our constant value, namely the case with D=15mm, p_b=35mtorr where the value of Re varies between approximately 3500 and 4500 in the dual solution domain. The calculated Mach number on the axis immediately upstream of the first shock reflection is plotted in Figure 4.23 for a number of representative points in the pressure ratio range. Within the dual solution domain, a higher Mach number is reached before the shock reflection in the regular reflection cases. This trend is discussed in section 4.4.3.

Across most of the pressure ratio range the predicted reflection type matches the experimentally observed type. Very good agreement between calculated and experimental temperature profiles was achieved in these cases. Figures 4.24 to 4.43 show several comparisons. Note that absolute temperatures are shown here, the ambient temperature being 288K. The experimental data[71] was obtained using a non-intrusive measurement technique, with expected accuracy of $\pm 5\%$. Figures 4.24 to 4.34 compare temperature results on the plume centre-line and across several radial sections respectively for a regular reflection at $p_r/p_b = 228.6$. Figure 4.24

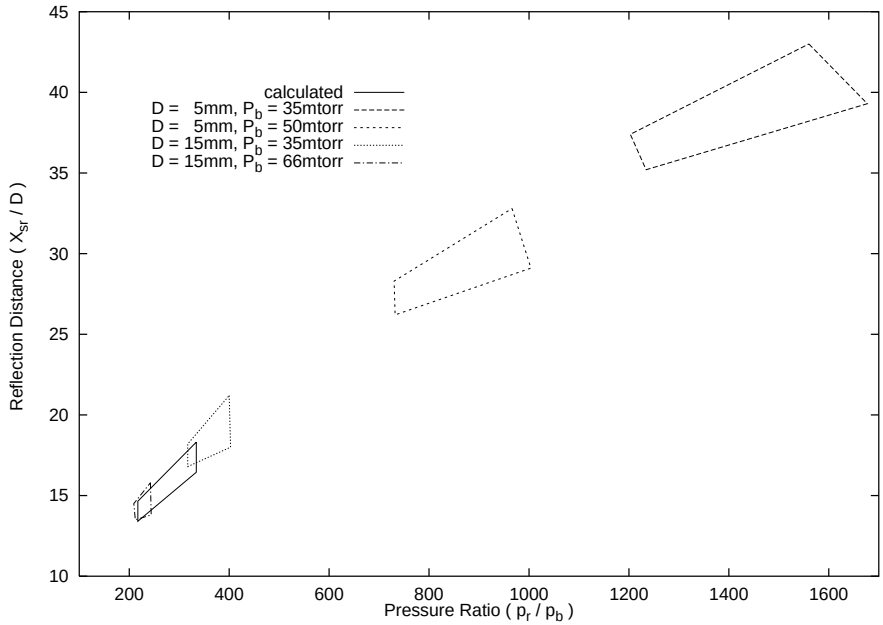


Figure 4.22: *Location of hysteresis loops, experiments and computation*

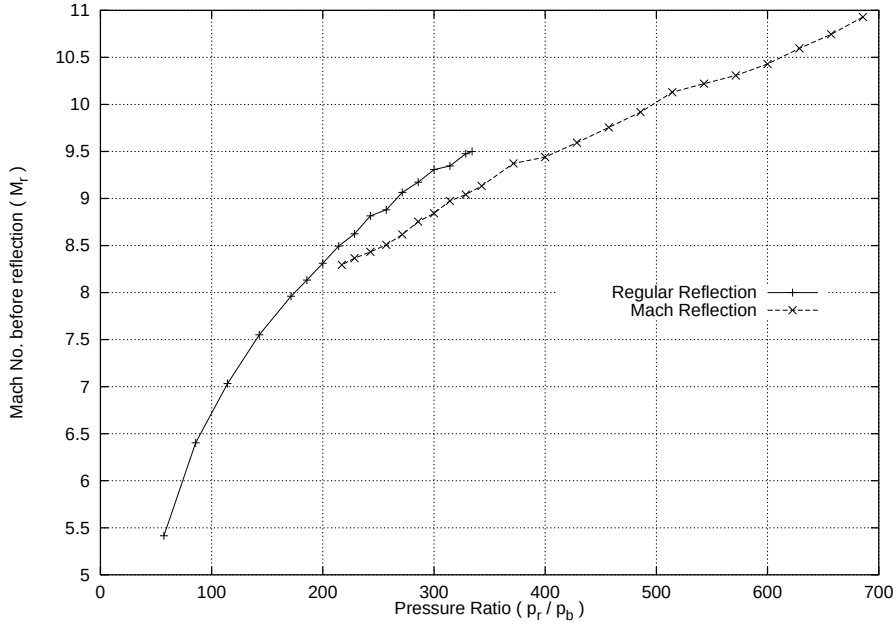


Figure 4.23: *Mach no. before reflection for range of pressure ratios, Re = 4000*

shows a good prediction of the regular reflection location, indicated by the sharp rise in temperature, and downstream of the reflection agreement is also good although temperature is slightly over-predicted. The radial temperature profile comparisons for regular reflection show good agreement. Figure 4.28 for example shows good agreement in the temperature profiles at an axial position downstream of the first regular reflection. At the centre-line the temperature is high since the gas has been compressed by the incident and reflected shock and has yet to re-expand. Moving across the plume, there is a sharp decrease in temperature as the reflected shock is traversed. The fast moving gas in the shock layer behind the incident shock, which has yet to be processed by the reflected shock, is shown by the temperature trough. The temperature recovers through the shock layer to the ambient value. The differing behaviour at the centre-line in Figure 4.25 is because $X/D = 14.84$ is just upstream of the predicted shock reflection distance, but is at the experimentally observed shock reflection distance. Figures 4.35 to 4.43 compare temperature results on the plume centre-line and across several radial sections for a Mach reflection at $p_r/p_b = 328.6$. Good agreement is also demonstrated here. The differing behaviour near the centre-line in Figure 4.36 is because $X/D = 16.28$ is just downstream of the predicted shock reflection distance, but just upstream of the experimentally observed shock reflection distance. Note that in Figure 4.37 the calculated temperature reaches the stagnation value of 288K after the Mach disc, implying that the flow has stagnated. This will be discussed in section 4.4.3.

In section 4.2.4 it is described how in the case of two dimensional shock reflection hysteresis the limits of the dual solution domain can be calculated from knowledge of the Mach number upstream of the reflection and the incident shock angle. In principle a similar analysis is possible here; the Mach number and local shock angle can be obtained from the CFD results, and the theoretical limits to the dual solution domain calculated and compared with the numerical results. However, this approach was not successful since the shock angles are difficult to measure accurately from field plots due to curvature of the shock and shock smearing. Other aspects of the analysis of the computational results are also hampered by this problem, as discussed in section 4.4.3.

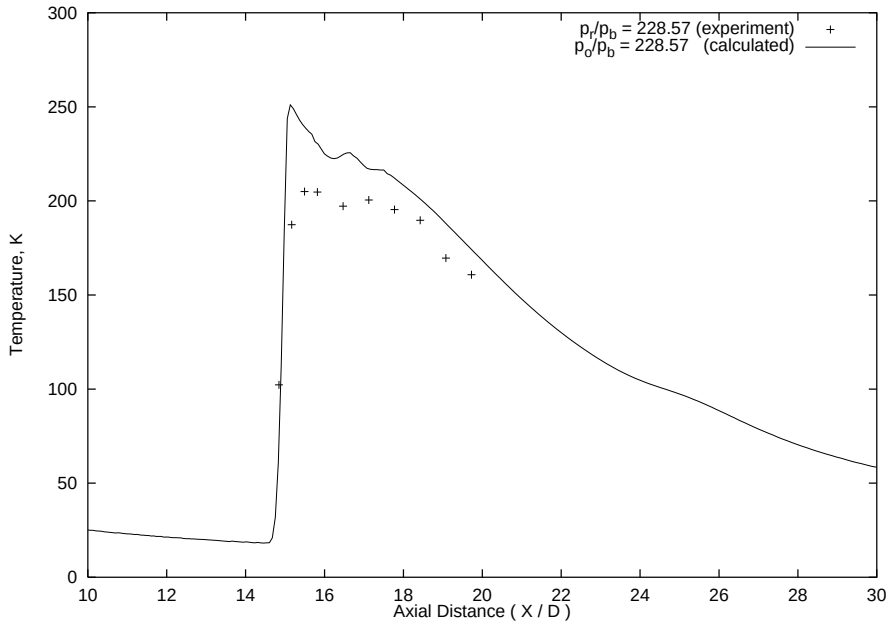


Figure 4.24: *Centre-line temperature, regular reflection*

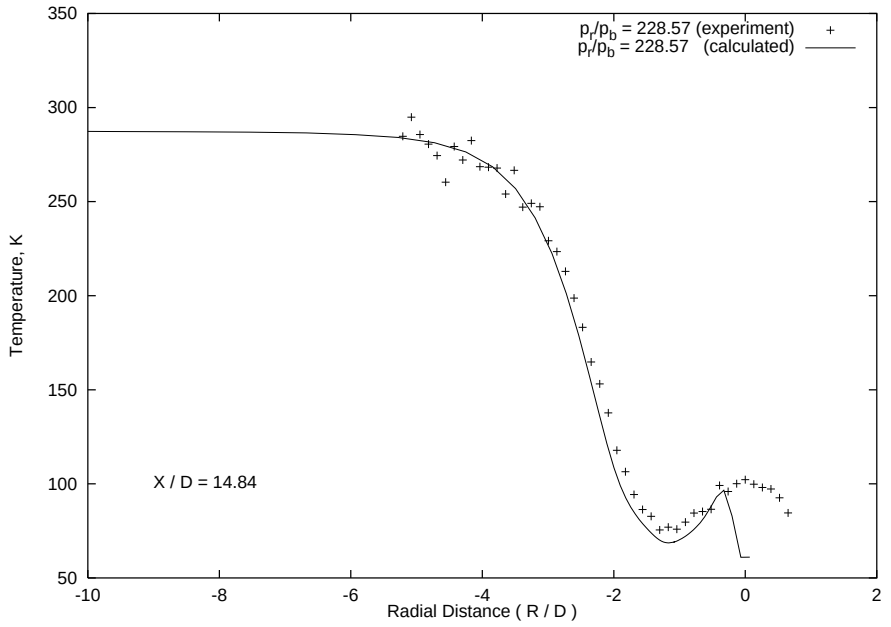


Figure 4.25: *Temperature profile at* $X/D = 14.84$*, regular reflection*

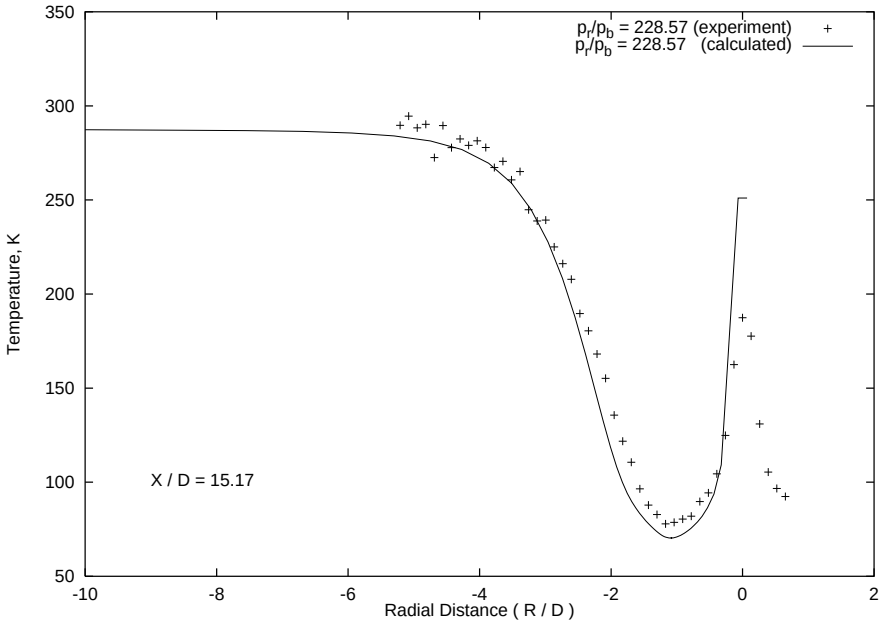


Figure 4.26: *Temperature profile at $X/D = 15.17$, regular reflection*

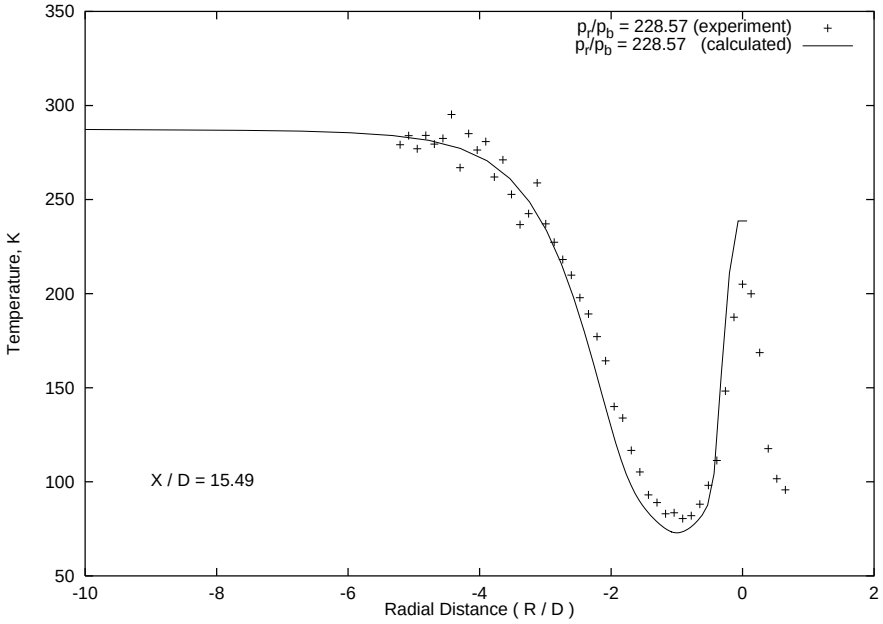


Figure 4.27: *Temperature profile at $X/D = 15.49$, regular reflection*

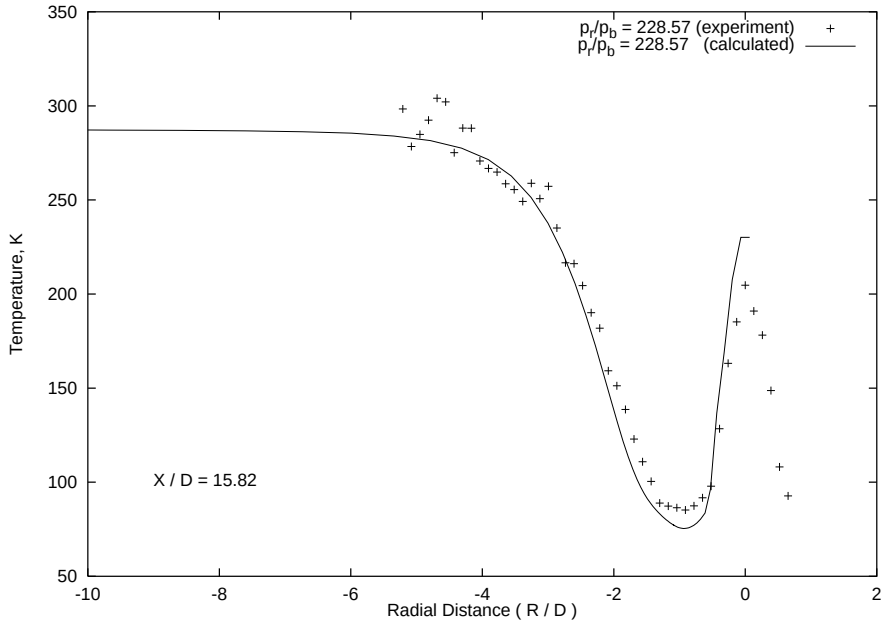


Figure 4.28: *Temperature profile at* $X/D = 15.82$*, regular reflection*

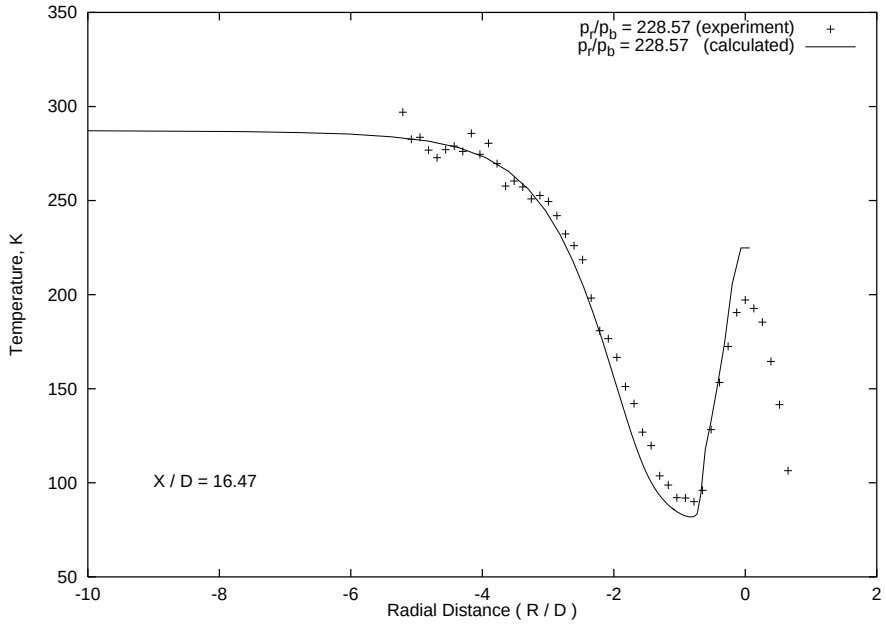


Figure 4.29: *Temperature profile at* $X/D = 16.47$*, regular reflection*

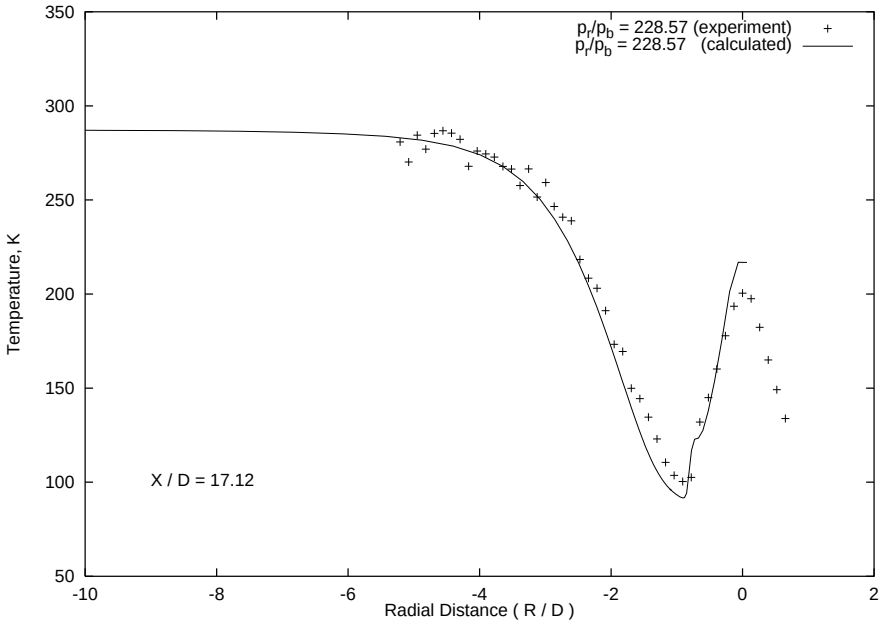


Figure 4.30: *Temperature profile at $X/D = 17.12$, regular reflection*

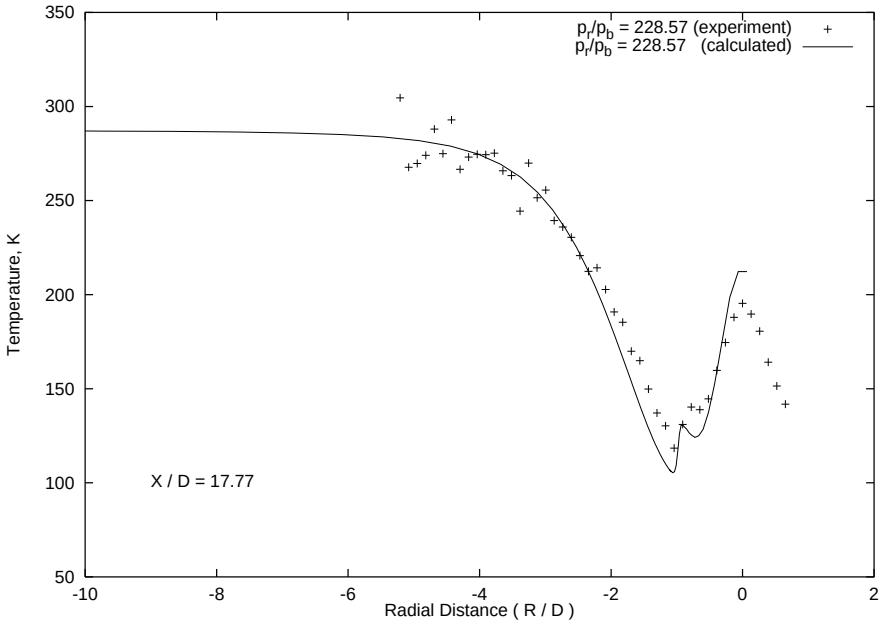


Figure 4.31: *Temperature profile at $X/D = 17.77$, regular reflection*

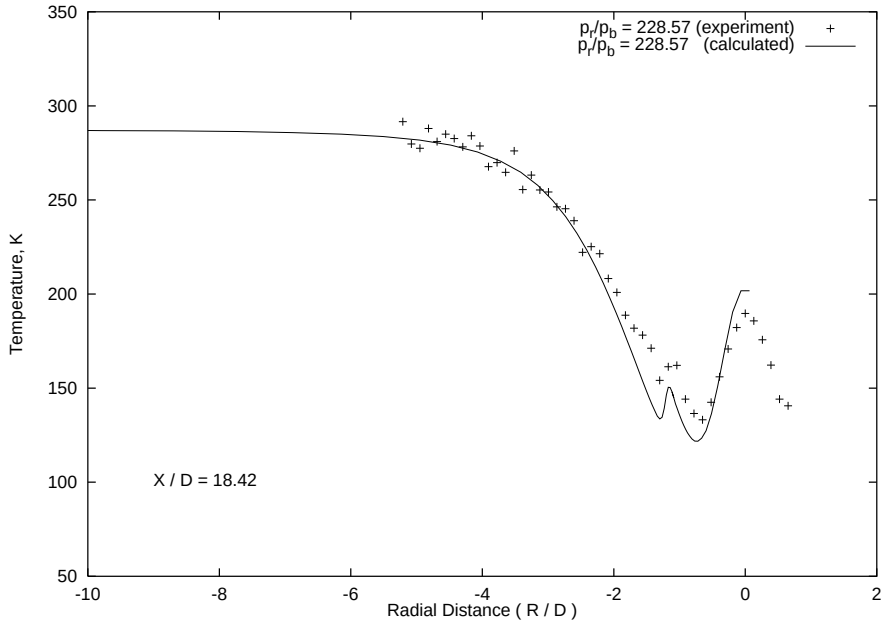


Figure 4.32: *Temperature profile at* $X/D = 18.42$*, regular reflection*

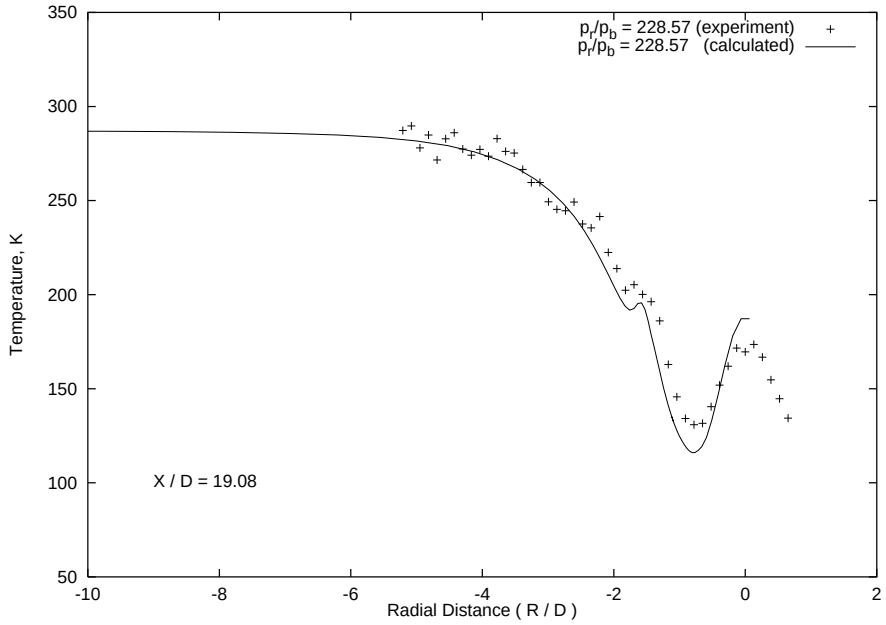


Figure 4.33: *Temperature profile at* $X/D = 19.08$*, regular reflection*

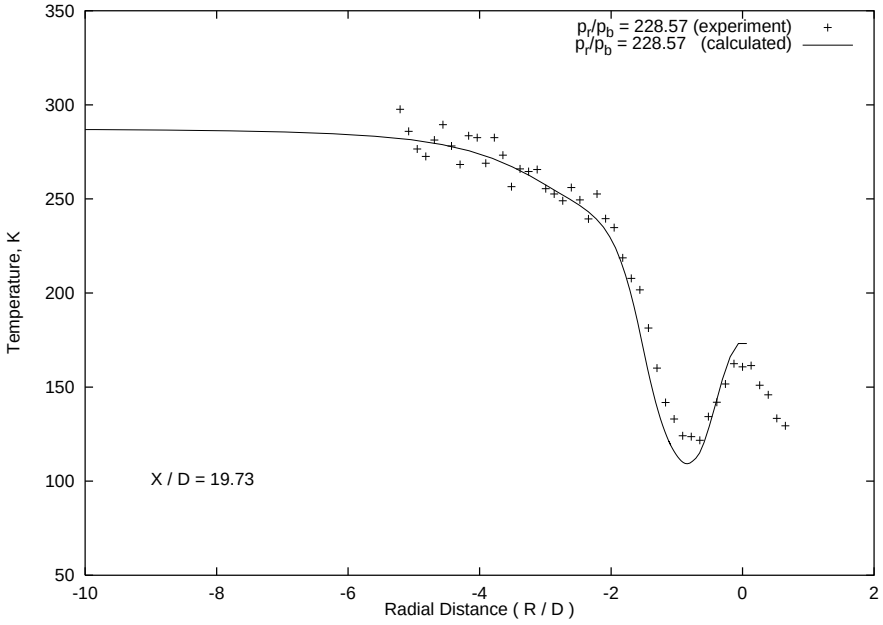


Figure 4.34: *Temperature profile at $X/D = 19.73$, regular reflection*

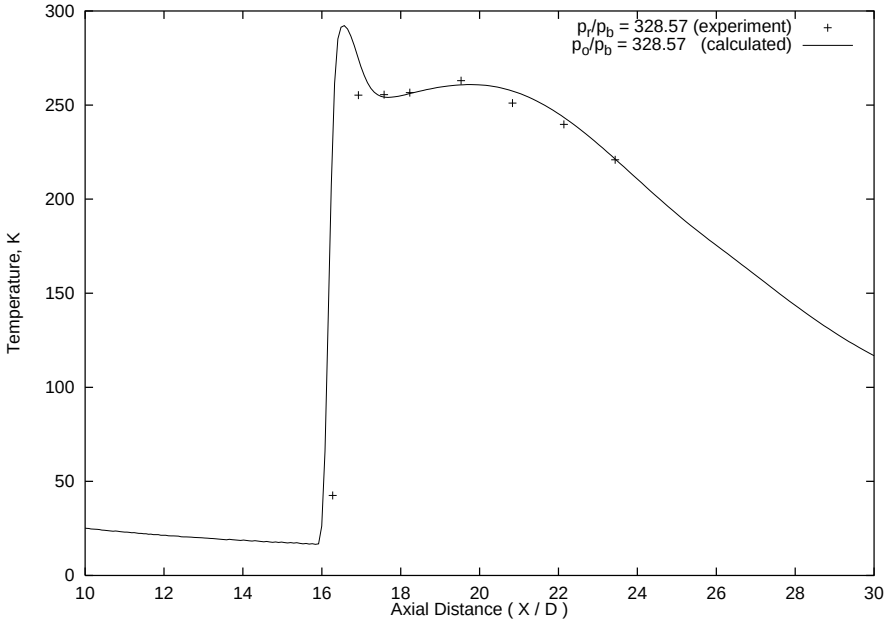


Figure 4.35: *Centre-line temperature, Mach reflection*

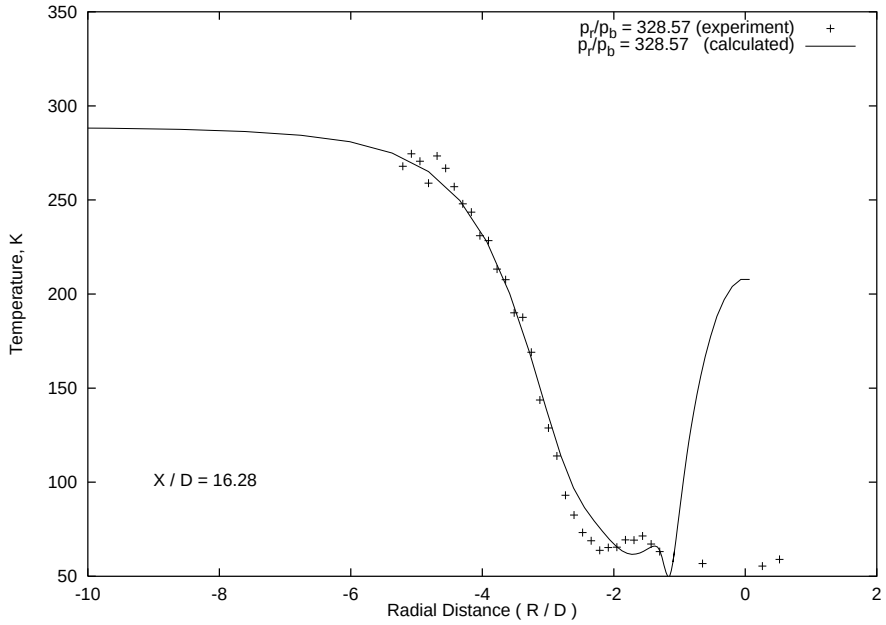


Figure 4.36: *Temperature profile at* $X/D = 16.28$*, Mach reflection*

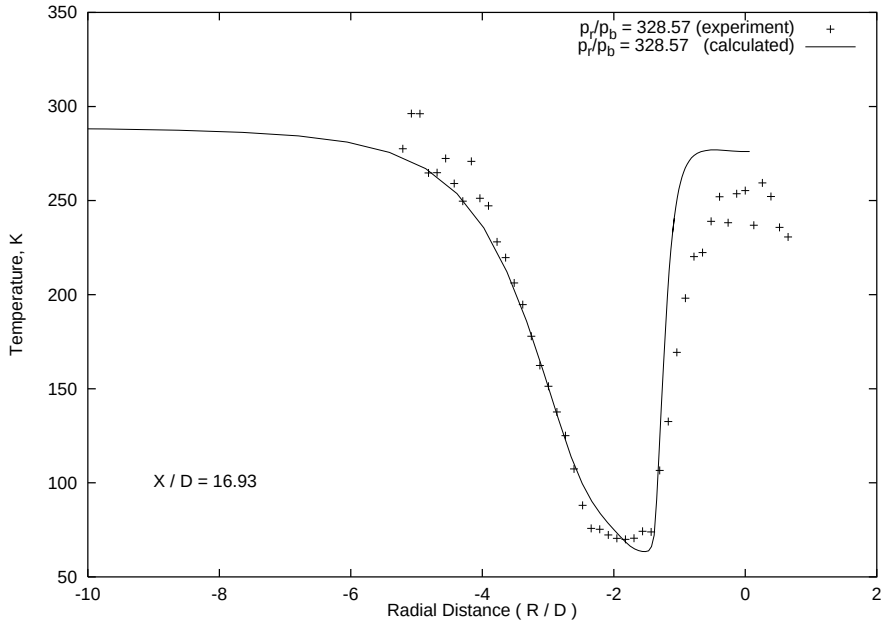


Figure 4.37: *Temperature profile at* $X/D = 16.93$*, Mach reflection*

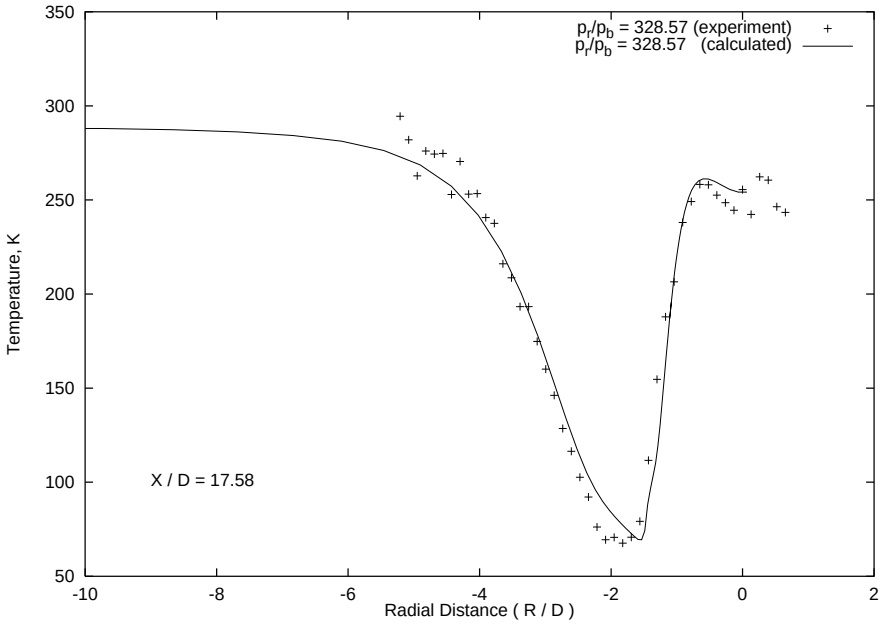


Figure 4.38: *Temperature profile at* $X/D = 17.58$*, Mach reflection*

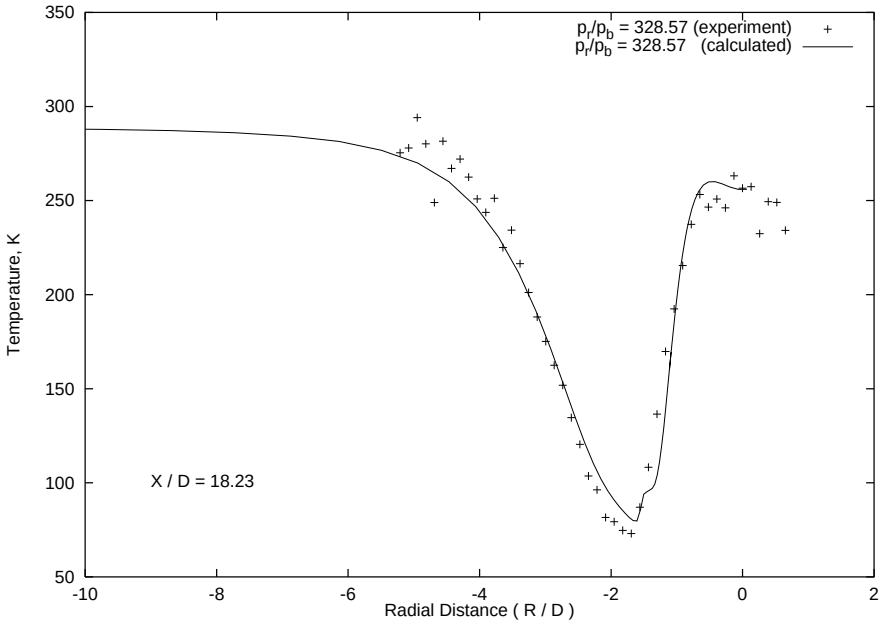


Figure 4.39: *Temperature profile at* $X/D = 18.23$*, Mach reflection*

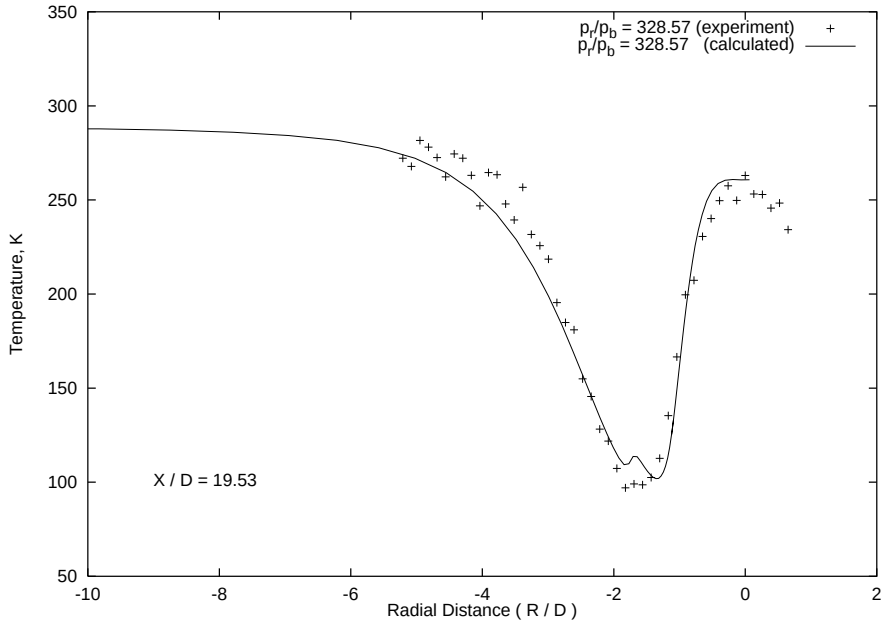


Figure 4.40: *Temperature profile at* $X/D = 19.53$*, Mach reflection*

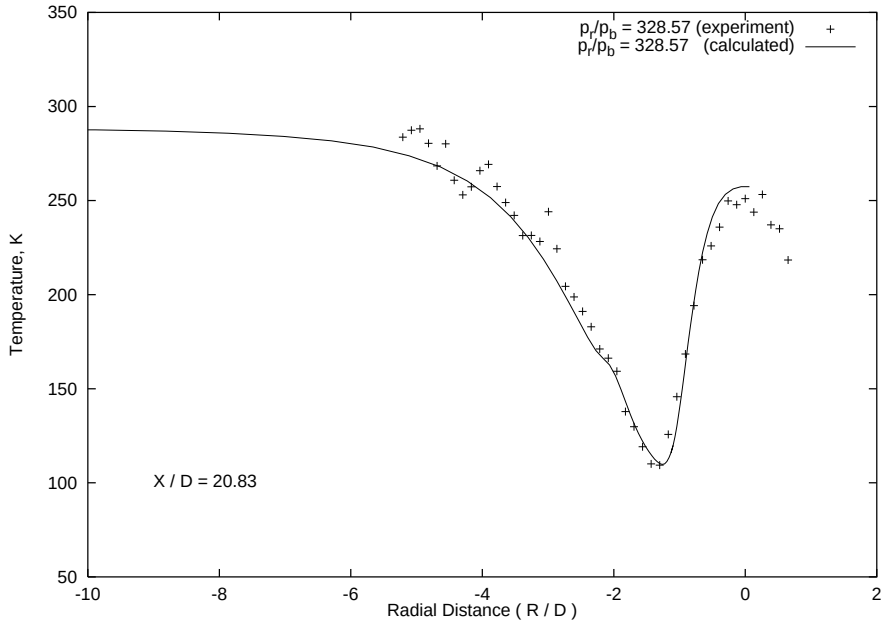


Figure 4.41: *Temperature profile at* $X/D = 20.83$*, Mach reflection*

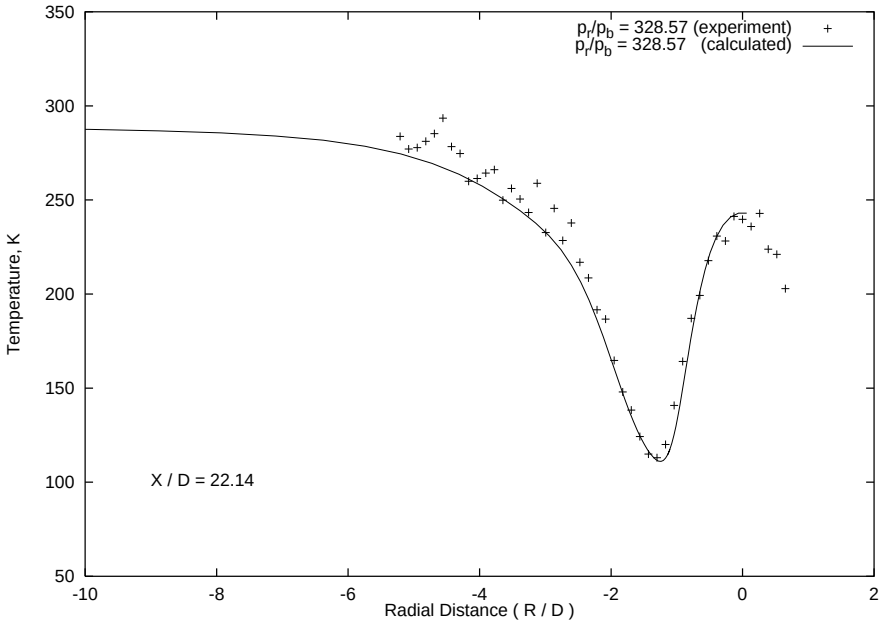


Figure 4.42: *Temperature profile at $X/D = 22.14$, Mach reflection*

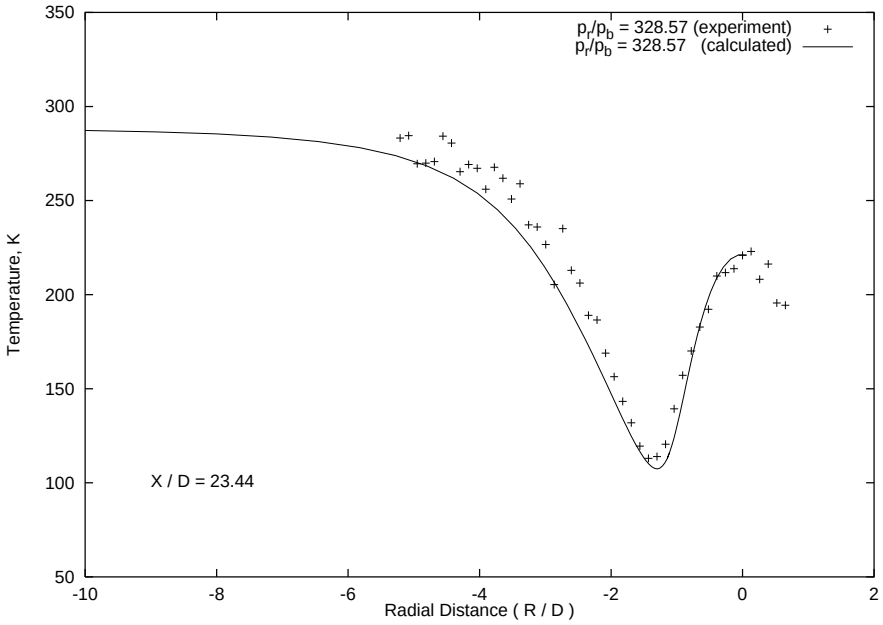


Figure 4.43: *Temperature profile at $X/D = 23.44$, Mach reflection*

4.4.3 Plume Structure

Presentation of Results

It is useful here to include a note on the sequence in which the results are presented. The figures appear out of sequence when referenced from the text; this is because all of the field plots (whether for RR or MR) are grouped together, followed by the centre-line plots, in this way aiding comparison of figures. It should be noted that the field plots of the CFD results presented in this section agree well with the excellent flow visualisation photographs included in the report on the experiments[71].

Regular Reflection

Figure 4.44 shows calculated density contours for a pressure ratio $p_r/p_b = 185.7$ which lies in the regular reflection range. The figure clearly shows the repeated shock cell pattern typical of this regime. Figures 4.45 to 4.47 show a detail of the second shock cell including the incident shock from the first cell and reflected shock at the beginning of the third. Pressure contours, Mach contours, velocity vectors and streamlines are shown. For clarity velocity vectors are shown for only every fifth grid cell in the axial direction and every fourth in the radial direction. From these field plots the important elements of the plume structure can be visualised. On exiting from the nozzle (on the left hand side of Figure 4.44) the air is at a higher pressure than the ambient air and expands sharply, increasing the cross-sectional area of the plume. Expansion waves reflect from the free jet boundary as compression waves, and in so doing turn the jet boundary towards the axis. The curved nature of the jet boundary causes the compression waves to coalesce and form an oblique shock wave, the incident shock labelled. Air passing through this shock is turned back towards the axis and collects in a shock layer of increasing density, causing the shock itself to turn further towards the axis. This is also encouraged by the increasing Mach number of the air before the shock in the still expanding core flow, whose pressure now lies below the background pressure. The axisymmetric shock intersects the axis and is reflected as another oblique shock. This Regular Reflection is analogous to the Regular Reflection in two dimensional uniform flow discussed in Chapter 4.2. The shock reflection is the mechanism through which the

condition of axial flow on the centre-line is achieved; after the incident shock the flow is converging on the axis and is then turned away by the reflected shock. The flow direction can be clearly seen from the streamlines in Figure 4.47. Immediately downstream of the reflection point the air being processed by the reflected shock is of increasing density due to the accumulated shock layer, turning the shock towards the axis. This tendency is quickly overtaken by the rapidly re-expanding core flow which causes the shock to turn outwards again towards the jet boundary. The change in curvature of the reflected shock is best seen in Figure 4.45. The shock is reflected as expansion waves by the free jet boundary which is turned back away from the axis. This expansion reinforces the expansion of the under-expanded core flow, and initiates another shock cell when the expansion waves again reflect from the jet boundary. The pattern is repeated, its strength gradually lessening, until the structure is destroyed by diminishing pressure ratio and mixing.

Figures 4.55 to 4.58 show calculated centre-line distributions of pressure, density, Mach number and axial velocity. Regular reflections arising for three pressure ratios are shown; $p_r/p_b = 334.3$ is the highest pressure ratio for which regular reflection occurs (a limit of the hysteresis loop), $p_r/p_b = 57.1$ was the lowest pressure ratio considered and $p_r/p_b = 185.7$ was selected as an intermediate point. The ragged peaks to the pressure and density curves for the lower pressure ratio cases are possibly explained by the interaction of the shock layer behind the incident shock with the reflected shock. It is interesting to note that upstream of the interactions all of the curves are coincident. Despite the fact that the cross-sectional area of the plume increases with pressure ratio, the core expansion along the axis appears to be independent of pressure ratio until the reflection occurs. From Figure 4.57 it can be seen that the flow behind the first regular shock reflection may become subsonic. From the present calculations, the lowest pressure ratio at which this occurs is $p_r/p_b = 171.4$ and as the pressure ratio increases in the regular reflection range the subsonic region becomes larger and the minimum Mach number smaller. At the upper limit of the RR range ($p_r/p_b = 334.3$) the subsonic region is 2.65 throat diameters in length with a minimum Mach number of 0.26 . More will be said about the region of subsonic flow in section 4.4.3.

Mach Reflection

As the ratio of nozzle exit to background pressure increases, the amount of expansion at the nozzle exit increases and the shock cell grows in size. As the pressure ratio increases, the angle at which the incident shock intersects the axis increases. This is a trend which is evident from comparison of contour plots for various pressure ratios, but the actual angle is difficult to measure precisely due to the curvature of the shock and its apparent thickness in the CFD results due to shock smearing. The increasing angle has the result that the flow behind the incident shock is deflected more towards the centre-line. The stronger incident shock also results in a greater decrease in Mach number across the shock, an effect which is countered by a greater Mach number in the core flow upstream of the incident shock, as shown in Figure 4.23. The shock deflection angle necessary for a reflected shock to re-align the flow is thus increasing, and at the same time the Mach number between shocks may be decreasing. A point is reached where an oblique shock solution for the required θ given M_1 is not possible. The re-alignment is in this case achieved via a Mach reflection, which consists of a normal shock called a Mach disc and a curved oblique shock, see Figures 4.48 to 4.51. The flow is subsonic behind the Mach disc, but is supersonic behind the oblique shock. These areas are separated by a slip line which emanates from the triple point where all three shocks meet. Downstream of the Mach disc, the flow re-expands to become supersonic and initiates a second shock cell in a similar fashion to the case of regular reflection. As the pressure ratio is increased further, the shock cell grows in size, and the incident shock angle upstream of the triple point continues to steepen. This Mach Reflection is analogous to the Mach Reflection in two dimensional uniform flow discussed in Chapter 4.2.

A recirculation zone was predicted behind the Mach disc, see for example Figures 4.51 and 4.52. This surprising result was first reported by Martin Gilmore at DERA Farnborough for an as yet unpublished single calculation in the MR region. This feature is predicted in the present results for all the pressure ratios examined in the MR range. As can be seen from Figures 4.49 and 4.59, immediately downstream of the Mach disc the pressure is still increasing; this pressure gradient appears to be driving the recirculation. An explanation for the continuing increase in pressure

is that immediately downstream of the Mach disc the gas being processed by the reflected oblique shock is of relatively high density due to the accumulation in the incident shock layer.

The calculated Mach disc is curved, convex if viewed from upstream, for each of the pressure ratio values examined. The amount of curvature increases slowly with pressure ratio. The curvature is apparent in Figures 4.49 and 4.50. This curvature implies that the flow is being turned away from the axis at the triple point. This corresponds to an Inverted Mach Reflection following Hornung's classification [90]. However, due to the curvature of all three shocks and their apparent thickness in the present results it is difficult to precisely identify the location of the triple point and verify the Mach Reflection type. The flow direction changes significantly in the locality of the triple point, see Figure 4.52.

Figures 4.59 to 4.62 show calculated centre-line distributions of pressure, density, Mach number and axial velocity. Mach reflection results for three pressure ratios are shown; $p_r/p_b = 217.1$ is the lowest pressure ratio for which Mach reflection occurs (the lower limit of the hysteresis loop), $p_r/p_b = 685.7$ was the highest pressure ratio considered and $p_r/p_b = 342.9$ was selected as an intermediate point. As also shown in the regular reflection results, upstream of the interactions all of the curves are coincident. From Figure 4.62 it can be seen that the flow behind the Mach disc reverses. At the lower limit of the hysteresis loop ($p_r/p_b = 217.1$) the subsonic region is 5.58 throat diameters in length. At the highest pressure ratio considered ($p_r/p_b = 685.7$) the subsonic region is 8.82 throat diameters in length.

The shock reflection type in the subsequent shock cells downstream of the first was calculated to be regular in all cases, as shown in Figure 4.61 where the flow is supersonic following the second (and third) sudden compressions. However, this study has concentrated on the first shock cell and no grid independence study was carried out for the other cells.

Dual Solution Domain

Figures 4.53 and 4.54 show calculated density contours for both MR and RR for the same pressure ratio ($p_r/p_b = 285.7$), a condition which lies in the dual solution domain. Note that upstream of the first shock reflection no difference in the

flow behaviour can be detected. This point is supported by Figures 4.63 to 4.67 where centre-line distributions of pressure, density, Mach number, axial velocity and temperature are compared for the same calculations. Upstream of the first shock reflection the curves are coincident. It is also clearly shown in these figures that the MR occurs slightly upstream of the corresponding RR, allowing a greater initial expansion in the RR case.

Pseudo-Mach Reflection

There is some evidence to suggest that the regular reflections discussed above are in fact Mach reflections with a Mach disc of small diameter. In Figure 4.46 there appears to be a slip line behind the 'regular' reflection; compare with the stream line behind the Mach reflection in Figure 4.50. As already noted in section 4.4.3 there is a significant subsonic region behind the 'regular' reflections at the higher pressure ratios. On close examination of the pressure contours in the region around the reflection (see Figure 4.45) there is an apparent Mach disc of approximately three grid cells in radius. As discussed in section 4.3.4 the criterion used in the grid independence study is the calculated hysteresis loop limits and not the resolution of any particular flow feature. However, refinement of the grid in this area by a factor of ten had no impact on this feature. It appears that two different levels of Mach reflection are occurring. Referring to the pressure-deflection diagrams introduced in section 4.2.3 the situation is complicated because the incident shocks are in this case curved. The condition (1) on the I polar could be in several different locations for the same incident shock because the deflection θ varies along the shock and conditions upstream of the shock are varying along its length. Consider an inverted Mach reflection, see Figure 4.2(f), in the underexpanded jet plume. As the pressure ratio is decreased from this condition the reflection type may suddenly change not to regular reflection but to an entirely different Mach reflection with a much smaller Mach disc. A possible location for the point (1) relative to its IMR location is at a greater θ value, corresponding to a longer incident shock which is steeper at the triple point and a smaller Mach disc. Such a reflection is represented in the (p, θ) plane by Figure 4.2(c). As discussed in section 4.2.3, a dual solution domain can exist in the simpler case of the reflection of a straight, planar shock wave. Examination of

the problem via (p, θ) diagrams does not provide information about the Mach stem length. Applied to the case of a curved incident shock this implies that multiple Mach reflection solutions are possible.

In the initial test programme it was assumed that a regular reflection was occurring, so the present results showing a 'pseudo-Mach reflection' were surprising. However, subsequent tests[107] have indicated that what had been previously accepted as a regular reflection is in fact a very small diameter Mach reflection. In addition, it has been argued that a true axisymmetric regular reflection cannot exist, with the results of numerical experiments presented as evidence[108]. This very small diameter Mach reflection has been termed an apparent regular reflection.

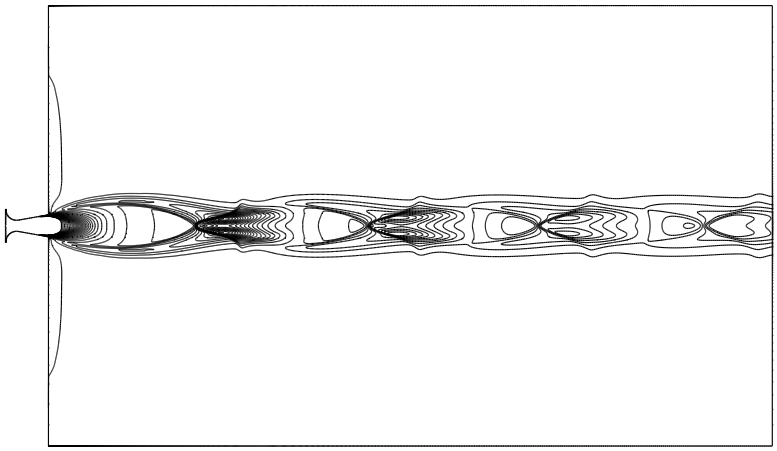


Figure 4.44: *Density contours showing regular reflection,* $p_r/p_b = 185.7$

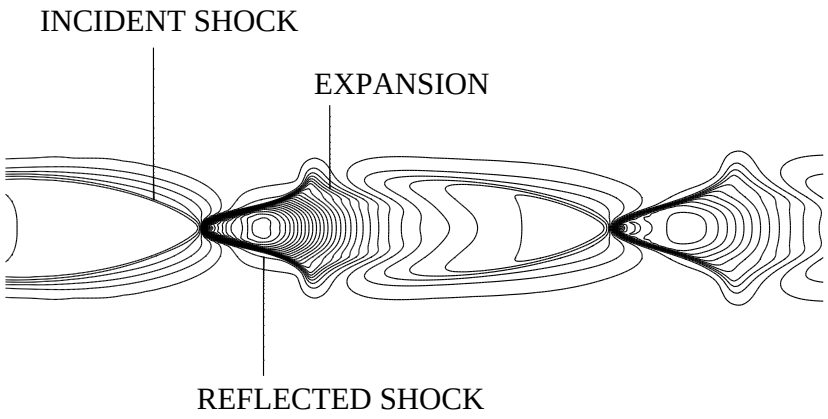


Figure 4.45: *Pressure contours showing regular reflection, $p_r/p_b = 185.7$*

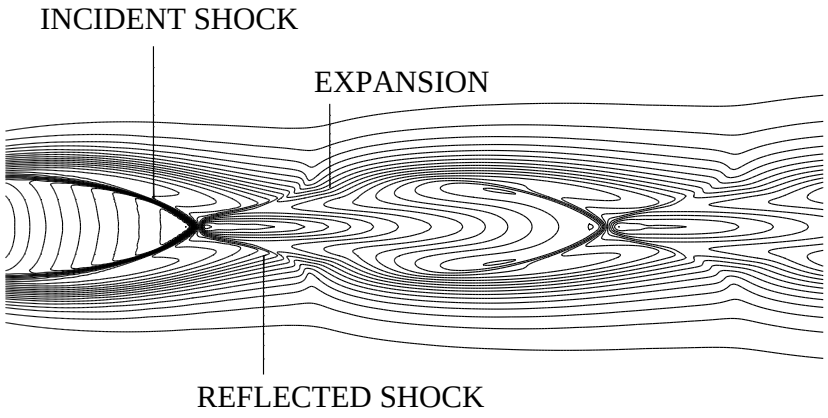


Figure 4.46: *Mach number contours showing regular reflection, $p_r/p_b = 185.7$*

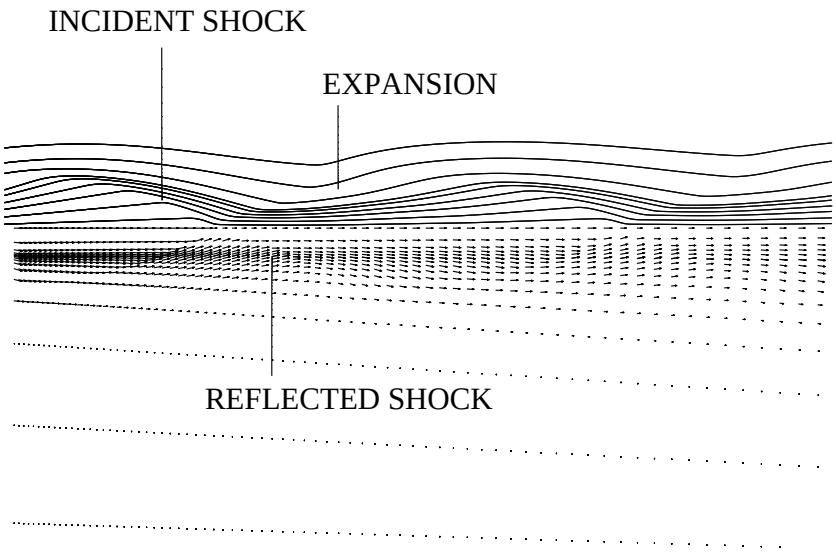


Figure 4.47: *Velocity vectors and streamlines showing regular reflection,* p_r/p_b = *185.7*

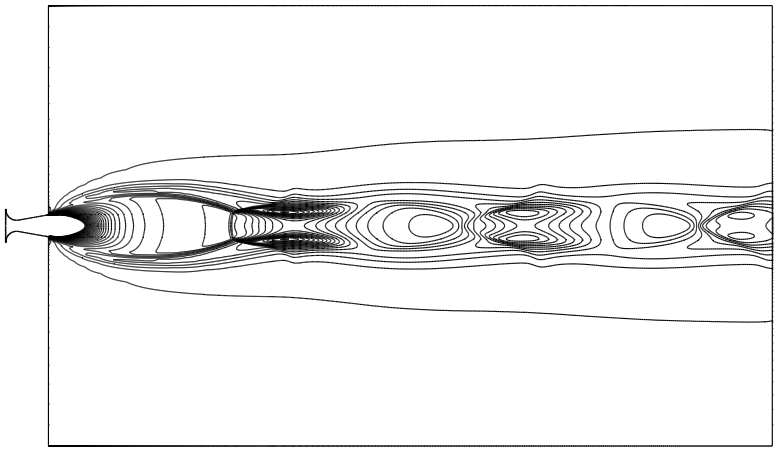


Figure 4.48: *Density contours showing Mach reflection,* p_r/p_b = *342.9*

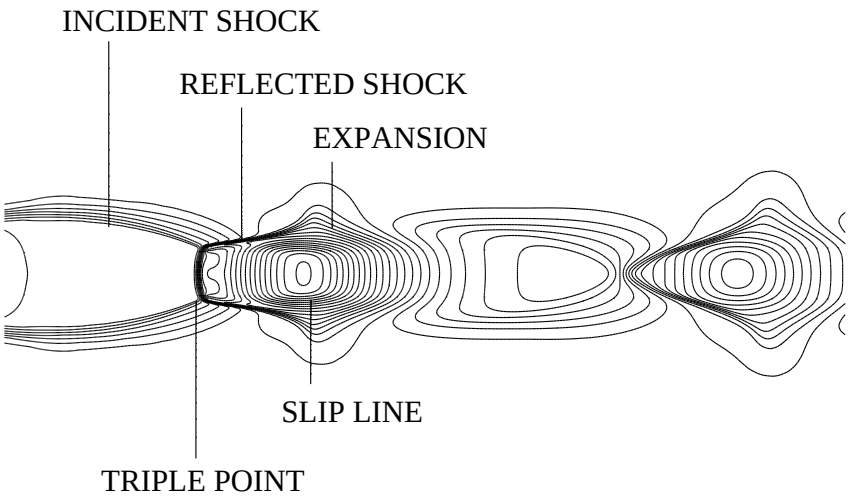


Figure 4.49: *Pressure contours showing Mach reflection, $p_r/p_b = 342.9$*

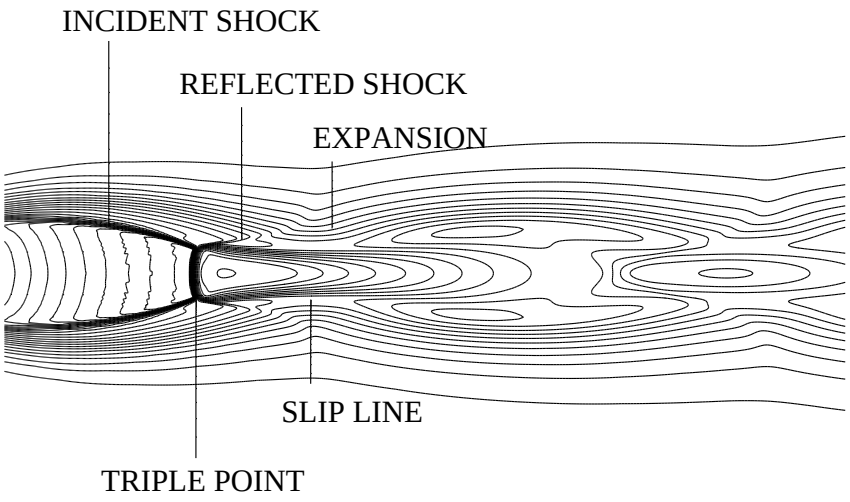


Figure 4.50: *Mach number contours showing Mach reflection, $p_r/p_b = 342.9$*

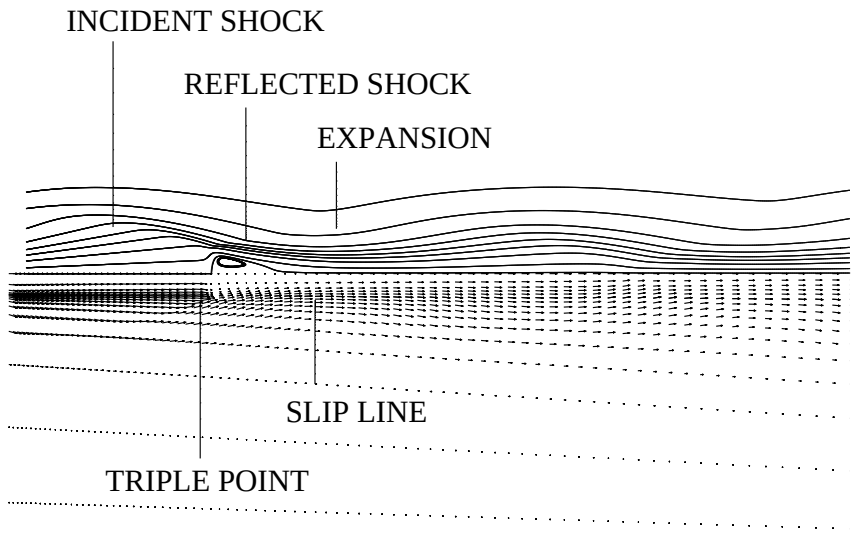


Figure 4.51: *Velocity vectors and streamlines showing Mach reflection, $p_r/p_b = 342.9$*

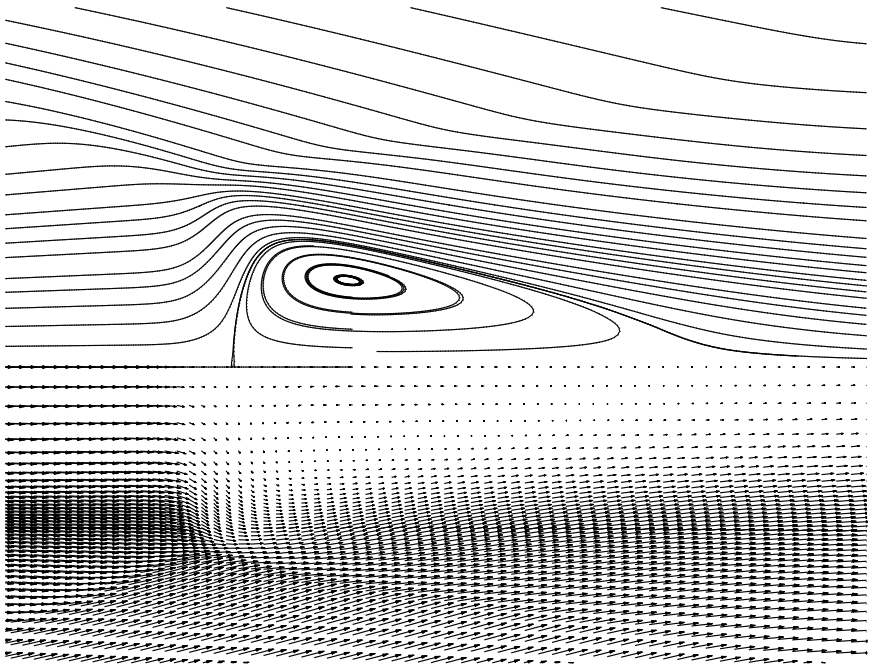


Figure 4.52: *Detail of velocity vectors and streamlines showing Mach reflection, $p_r/p_b = 342.9$*

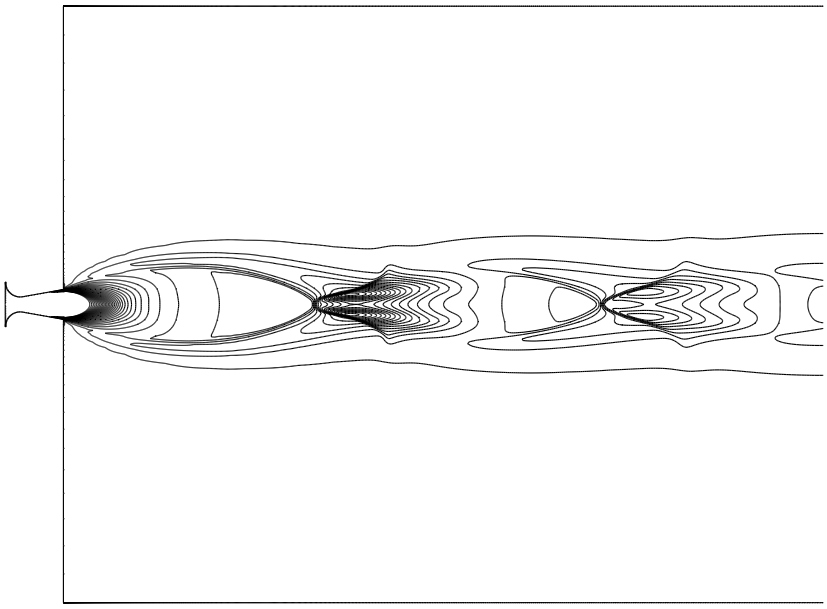


Figure 4.53: *Density contours showing regular reflection, $p_r/p_b = 285.7$*

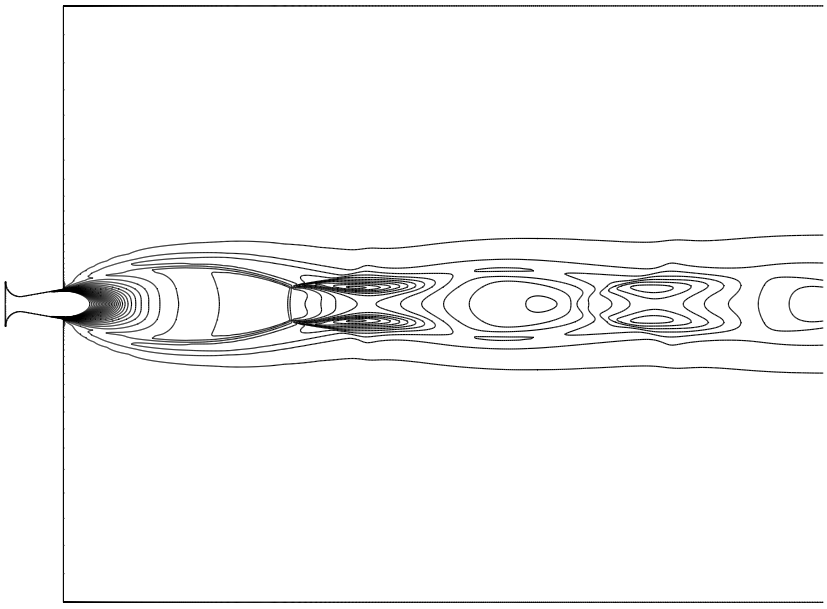


Figure 4.54: *Density contours showing Mach reflection, $p_r/p_b = 285.7$*

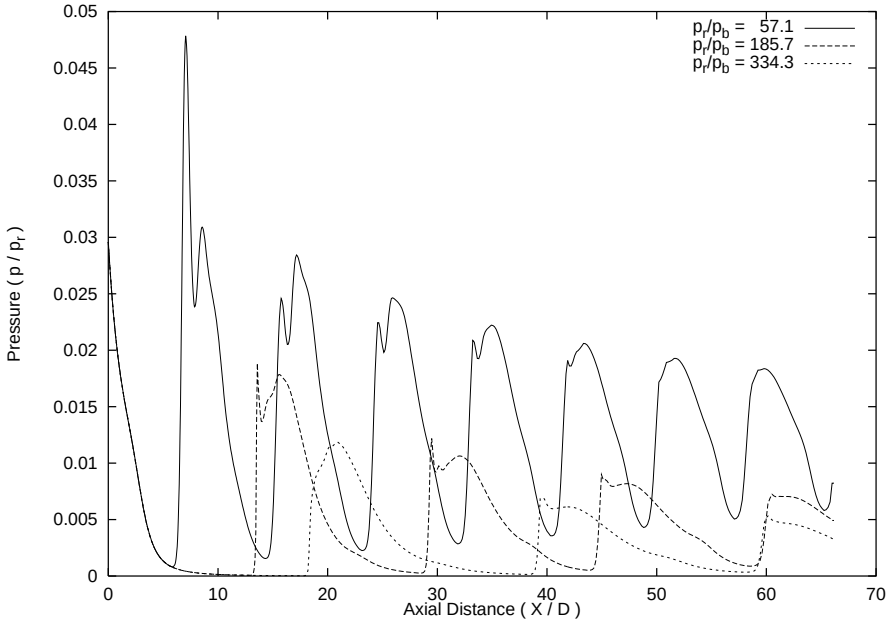


Figure 4.55: *Centre-line pressure distribution, regular reflection*

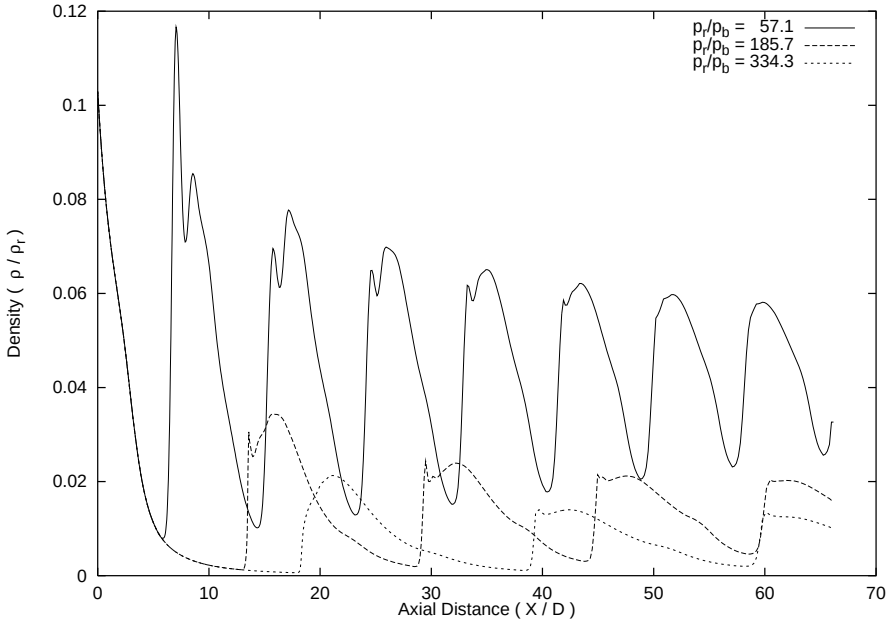


Figure 4.56: *Centre-line density distribution, regular reflection*

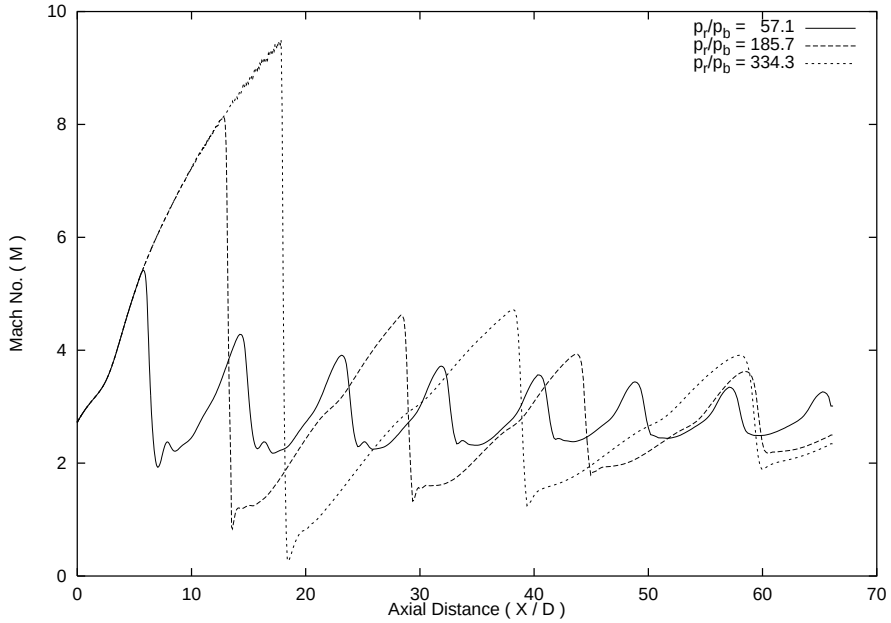


Figure 4.57: *Centre-line Mach no. distribution, regular reflection*

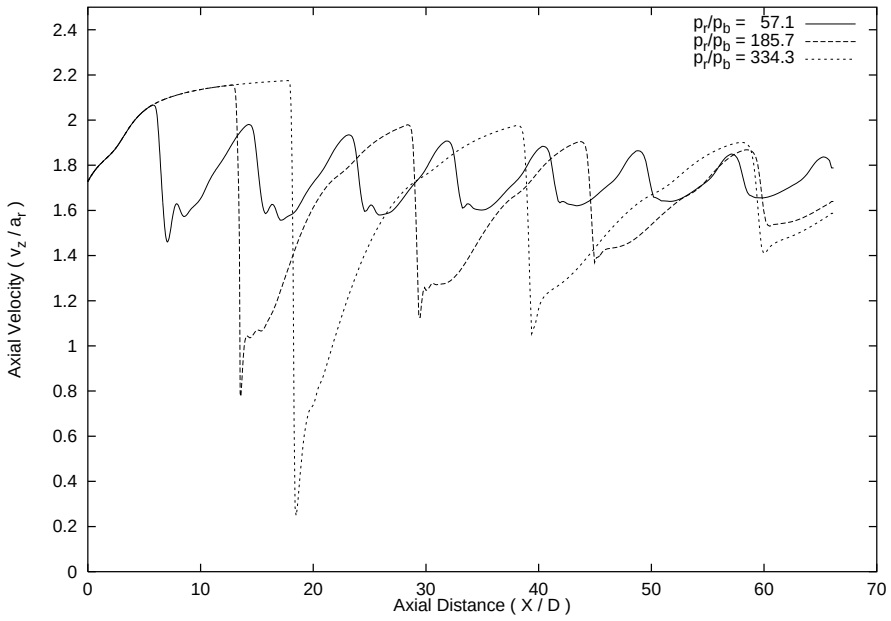


Figure 4.58: *Centre-line axial velocity distribution, regular reflection*

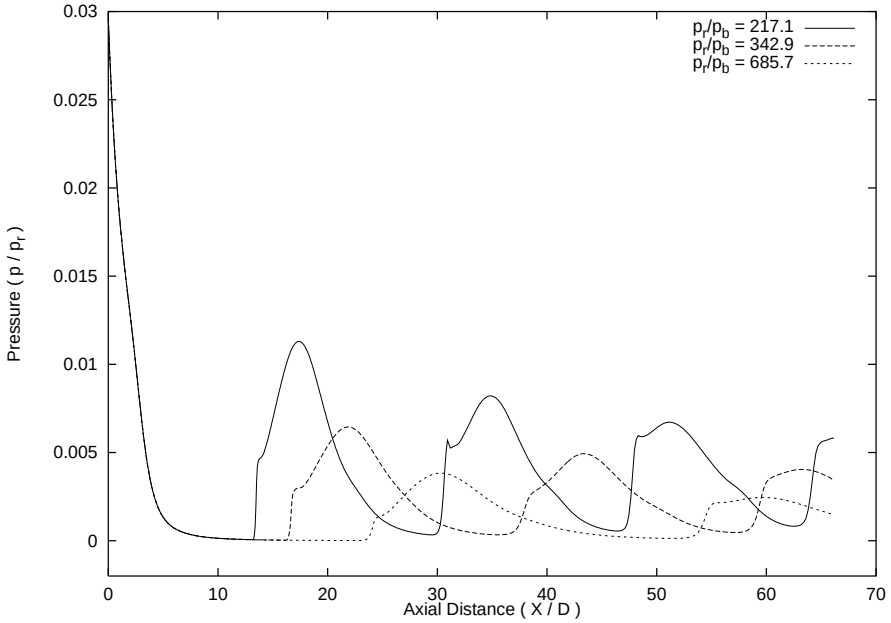


Figure 4.59: *Centre-line pressure distribution, Mach reflection*

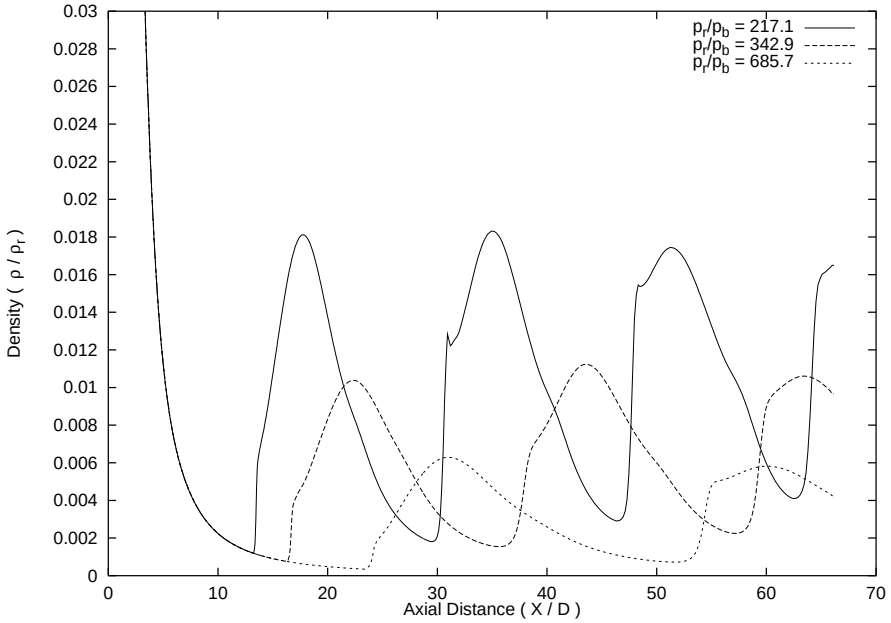


Figure 4.60: *Centre-line density distribution, Mach reflection*

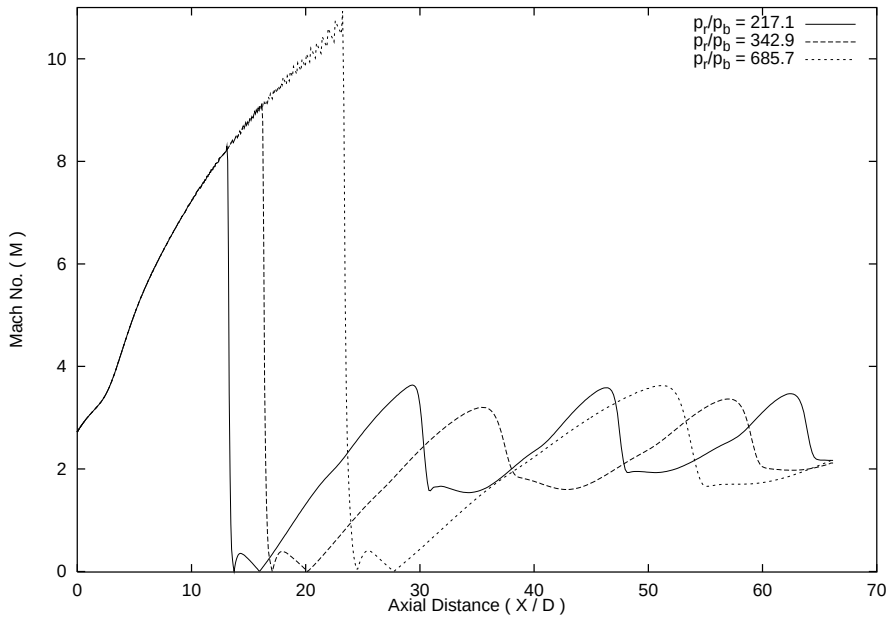


Figure 4.61: *Centre-line Mach no. distribution, Mach reflection*

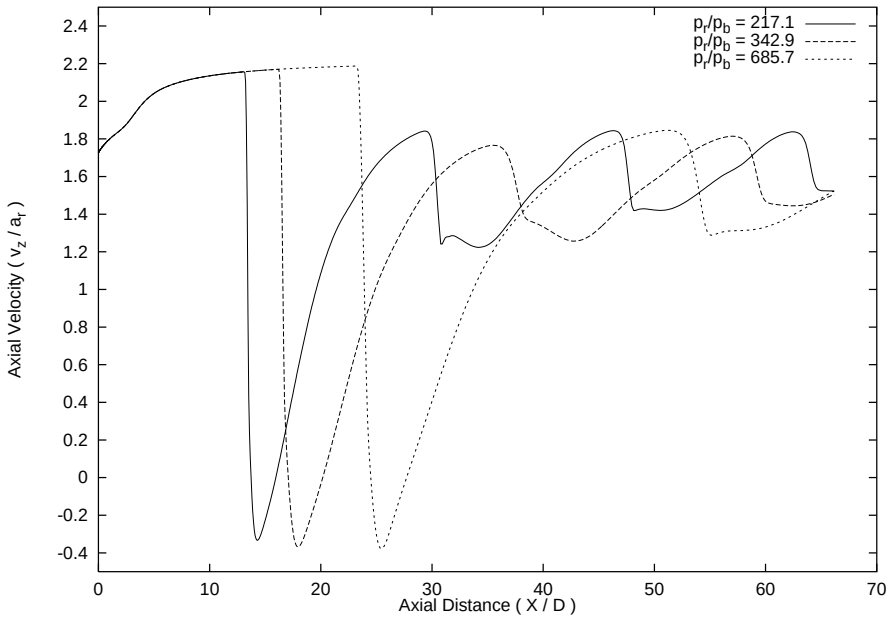


Figure 4.62: *Centre-line axial velocity distribution, Mach reflection*

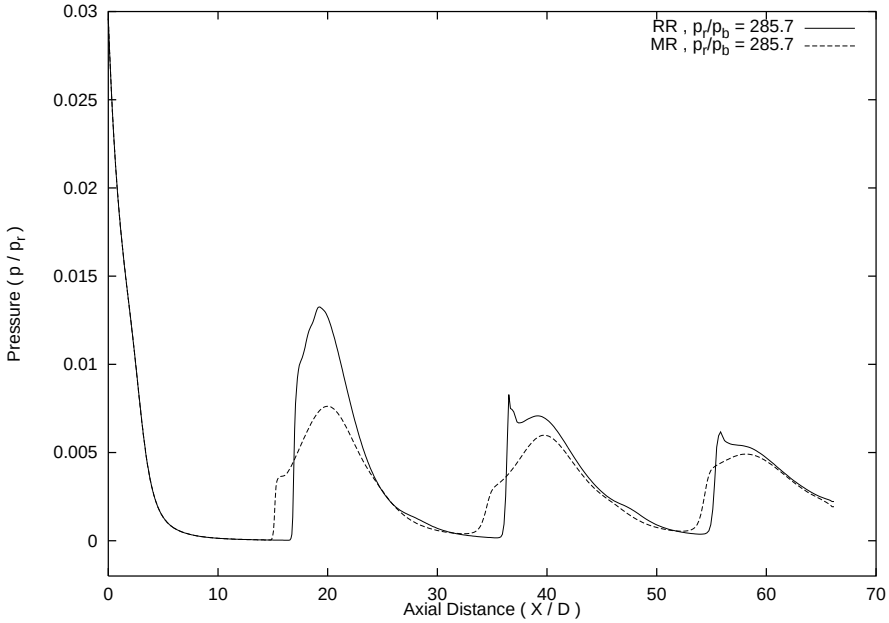


Figure 4.63: *Centre-line pressure comparison, regular and Mach reflection*

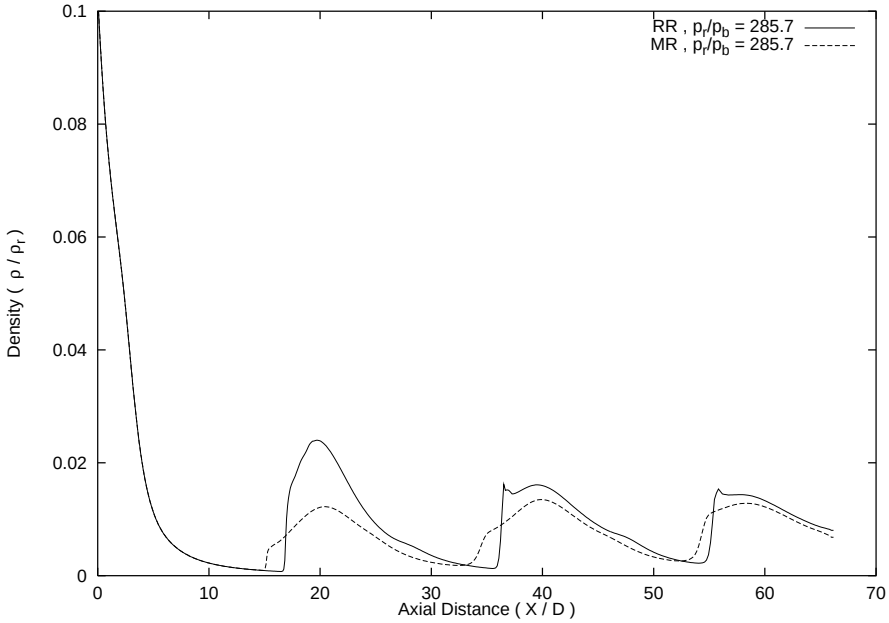


Figure 4.64: *Centre-line density comparison, regular and Mach reflection*

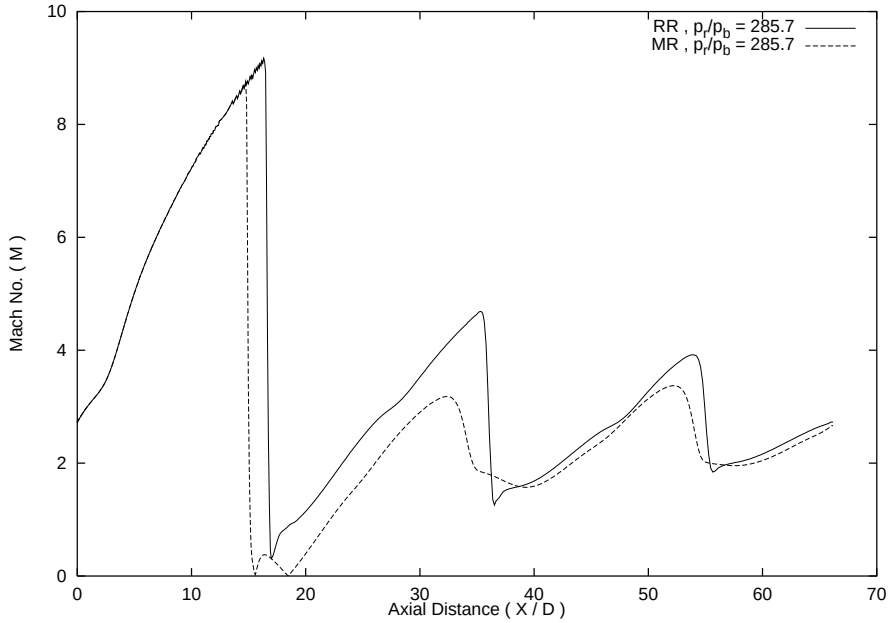


Figure 4.65: *Centre-line Mach no. comparison, regular and Mach reflection*

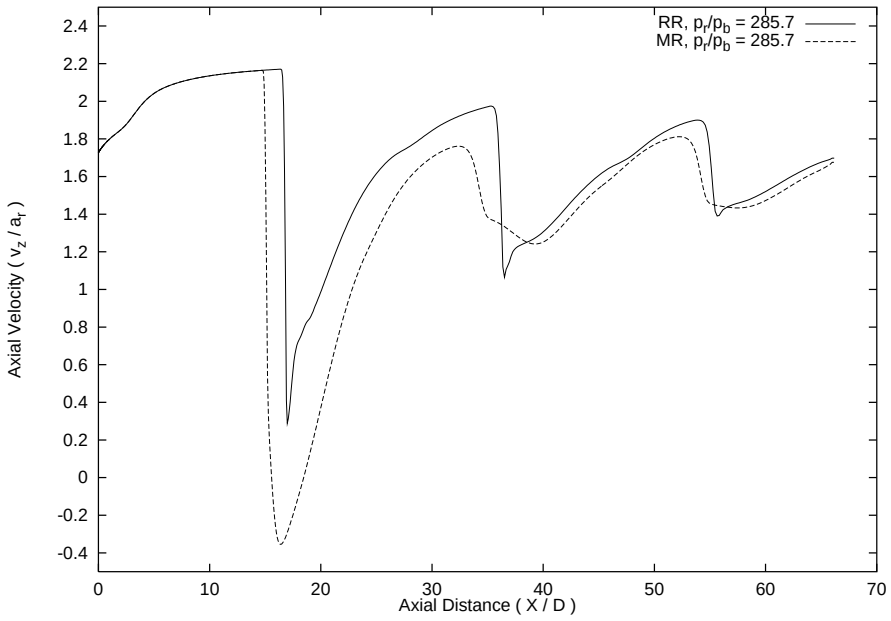


Figure 4.66: *Centre-line axial velocity comparison, regular and Mach reflection*

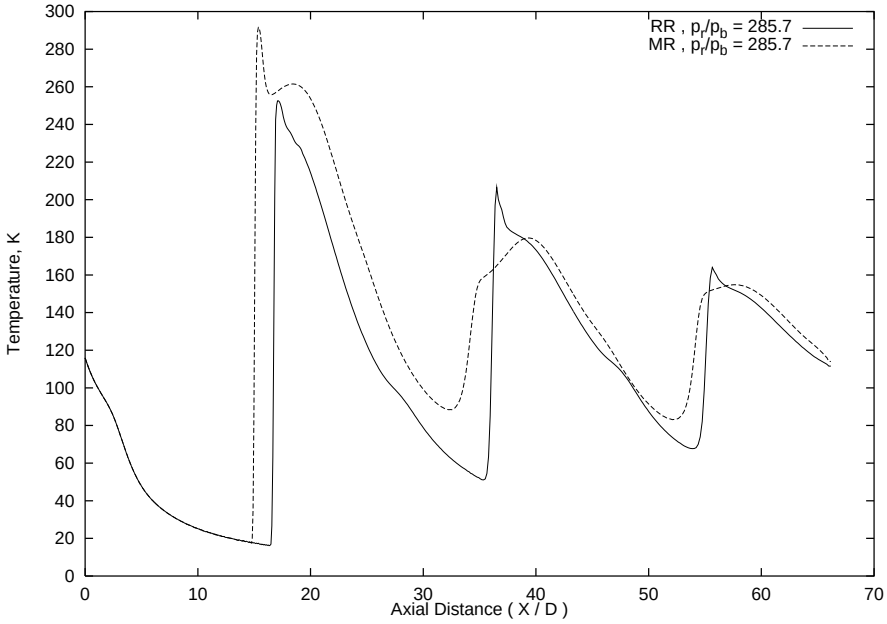


Figure 4.67: *Centre-line temperature comparison, regular and Mach reflection* T_o = *288.0 K*

4.4.4 Discussion

A hysteresis effect has been predicted in the shock reflection type of an axisymmetric jet plume for a range of pressure ratios and a constant Reynolds number. The predicted hysteresis loop agrees well with experimental results where the Reynolds number was also varying. The predicted temperatures in the jet plume agree well with experimentally recorded values. Field plots of the results which enable visualisation of the plume structure also agree well with the excellent flow visualisation images included in the original report. Having gained confidence in the accuracy of the simulation, the detail obtained from the CFD analysis was used to examine a number of flow features not recognised in the original experimental study. These features are Mach disc curvature, recirculation and continuing compression behind the Mach disc and the presence of a small diameter Mach disc in the apparent regular reflection. The possible existence of multiple Mach reflection solutions agrees with the shock reflection theory discussed in section 4.2. Further use of shock reflection theory in comparing the present axisymmetric results with established two-dimensional theory was hampered by difficulties measuring shock angles due to shock curvature

and smearing. A study of the shock reflection hysteresis in isolation rather than in the context of a jet plume would be much simpler to perform and could lead to a clearer understanding of the mechanisms involved, without the complications of curved shocks, shear layers etc. A full analysis of the present computational results using the methods used for the problem of two dimensional shock reflection hysteresis in uniform flow would require more accurate shock wave resolution than achieved in the present study. The success of the present study gives confidence that the same methods could be applied to promote the understanding of other shock interaction phenomena in the same series of experimental studies[107].

4.5 Conclusion

In this chapter the application of a CFD method to examine shock reflection hysteresis in an underexpanded jet plume has been described. Included in the study is a review of two-dimensional shock reflection hysteresis which has been useful in establishing theoretical background and terminology, and has also highlighted some remaining gaps in the understanding of this phenomenon. The interpretation of the CFD results against this theoretical background and combined with experimental data has contributed to the understanding of the plume structure and hysteresis phenomenon. The value of CFD as a scientific investigation tool for this type of problem has been clearly demonstrated, along with the wider potential of applying CFD to other problems where the role of experiments is limited.

Part II

Pre-Processing and Calculation Issues

Chapter 5

Towards Automatic Multiblock Topology Generation

5.1 The need for automation

5.1.1 Introduction

Multiblock or zonal structured grids remain a popular choice in CFD. This approach involves an unstructured arrangement of blocks with structured grids which conform with the problem geometry. The alternatives of unstructured, Cartesian, hybrid structured-unstructured and overset (Chimera) grids each have their own advantages and disadvantages. The choice of which one to use is difficult, an essential element of which is a compromise between the relative complexity of grid generation and flow solution. Multiblock grids afford the advantage of easier calculation management and lower operation counts and memory requirements due to their inherent structure, but grid generation for complex configurations is problematic and time-consuming. The subject of which method to choose is not discussed further here, for an introduction to the issue see [109],[24]. Here we are interested in the multiblock grid generation procedure, and note that to address its particular problems is relevant and useful since simulation using multiblock grids is popular. For some recent examples of its application see [110],[111],[112],[113].

5.1.2 Elements of the analysis process

Figure 5.1 shows a schematic diagram of the elements of a CFD analysis using multiblock grids. With modern CAD and graphical plotting software, the geometry definition and results analysis stages present few problems. Numerous satisfactory commercial packages exist for these tasks, with present work concentrating on improving speed and extending capability although the basic tools are well established. The flow solution stage is the subject of much ongoing research, but with modern computing power even large scale simulations can be achieved with reasonable turnaround times. The primary obstacle to obtaining accurate flow solutions is the lack of a practical, accurate and general turbulence model.

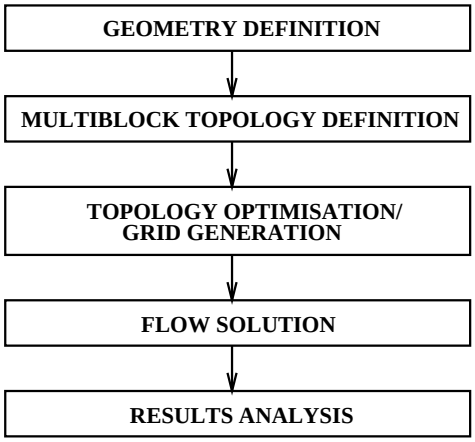


Figure 5.1: *A CFD analysis process using the multiblock method*

The bottleneck in the process occurs at the second and third stages. Even for fairly simple geometries in two dimensions, the task of designing a suitable arrangement for the grid blocks can be a demanding one. Each part of the problem geometry requires a body conforming local arrangement of the blocks, for example a 'C'-shaped arrangement around an aerofoil, but these local patterns are often difficult to match as a coherent whole. The task of defining an appropriate block pattern is known as

'topology generation'. In three dimensions the task can be daunting and requires considerable skill. Fine adjustments of the curves and block faces making up the topology and actually generating the grid to satisfaction can take man-months of effort for complicated configurations like full aircraft. The need to facilitate the topology and grid generation process by providing interactive graphical environments specifically designed for the task was recognised nearly ten years ago [114],[115]. A few years later Thompson and Weatherill [24] were able to list several commercial packages providing this capability and subsequent development has continued apace. Although these tools undoubtedly accelerate the process, the amount of time and effort required for grid generation still impedes routine analyses for multiple geometries, especially for complex configurations. Progress towards the alternative goal of fully or mostly automatic grid generation for arbitrary geometries [24],[115] has not been as impressive. In Thompson's recent review paper [25] the need for automation is particularly stressed. Real progress has been made by several authors but all of the diverse approaches suggested to date require a degree of skilled user input. The main problem is the difficulty in encapsulating the 'art' of topology generation in a programmable method. The approach of Dannenhoffer[116],[117], which is an integral part of the National Grid Project[118], is probably the most advanced method to date in terms of automating as much of the grid generation process as possible. An abstract "topology plane" is employed initially to interactively design the topology, and block faces are automatically set up by the code. A stochastic process is then employed to reduce the number of blocks. The user then proceeds to edit the topology and construct the grid using a state-of-the-art GUI. Stewart[119],[120] employs a search algorithm with a directional probe to build a two-dimensional block decomposition. This promising approach has proven difficult to apply generally, and it is unclear how well it could extend to three dimensions. The SAUNA[121] system employs a library of known topologies; to generate a new grid with a known topology is therefore straightforward, but for a new topology considerable effort is required to add to the library. The ICEM-CFD system[122] can automatically generate local topologies around recognisable components, after which the user must create the remainder of the topology. Unstructured quadrilateral and hexahedral mesh generation techniques have also been employed to create block topologies[123],[124]. Note

that the methods used are not fully automatic and appear to suffer from generality problems. However, this type of approach appears very promising since a number of general, fully automatic methods have been established in the parallel field of structural mechanics. In Section 5.1.3 it will be suggested that a particular method for initial topology generation possessing the characteristics required already exists and has been well developed. There is therefore the potential to fully automate one of the troublesome elements of Figure 5.1. Section 5.2 is the main part of the present study. Having defined an initial multiblock topology, the actual shape and location of the blocks must be defined. A new, straightforward method is proposed for automatically adjusting both the relative placement of blocks and the shape of the curves making up their edges. In this way the subsequent generation of the block interior grids can take place with little or no recourse to further manual block placement or edge shape modification. This process is here called 'topology processing'. Together with established algebraic grid generation and elliptic smoothing techniques this provides the potential for automation of the third stage in Figure 5.1. After a topology of good quality has been obtained, the task of generating the grid proper in the interior of the blocks becomes straightforward using conventional algebraic grid generation tools. Any remaining grid smoothness problems across block boundaries can be treated using elliptic smoothing.

5.1.3 Automatic topology generation

A multiblock grid consists of an unstructured arrangement of structured grid blocks. Traditionally the definition of this block arrangement is conceived by the expert user who views the domain in question and imagines the best way to fill it with blocks. This is a skilled task, especially in three dimensions. An attempt to replicate the expert's thought processes in code to produce an automatic tool would necessarily involve shape recognition and trial and error as well as an appreciation of the target flow solver's requirements for the grid. Rather than starting from scratch in an attempt to create such a tool, a simpler alternative is possible. Since the topology consists essentially of unstructured quadrilateral blocks in 2D or unstructured hexahedral blocks in 3D, it is possible that one or more automatic mesh genera-

tion procedures developed for structural analysis could be suitable for generating initial multiblock topologies. In this way the expert task of generating multiblock topologies for each individual case can potentially be reduced to the expert task of choosing an existing automatic mesh generation method which produces multiblock topology-like results. Several automatic unstructured quadrilateral and hexahedral mesh generation methods exist, see for example [125], the resulting meshes each having there own attributes. An approach which generates results consistently striking in their similarity to good multiblock topologies is the Medial Axis approach of Armstrong et al., see [126]-[130]. The method is based on a skeletonization technique (where for example a 2D shape is encoded in an essentially 1D manner) which is well known for its high quality of shape description. Intriguingly, the method was initially proposed as a model for human shape perception, which perhaps explains its ability to generate domain decompositions which fit geometries well, the main requirement of a multiblock topology. This speculation aside, in the Medial Axis approach there is an established automatic domain decomposition technique which results in good quality unstructured quadrilateral or hexahedral meshes which appear to meet the requirements of multiblock decompositions. Of course an initial topology formed in this way would consist of blocks with straight-sided faces. The initial topology may also have other unwanted features such as poor orthogonality at block corners and poorly shaped blocks which do not conform well with other blocks and the problem geometry. The re-shaping of the initial topology for our purpose is the subject of the next section.

5.2 Automatic topology processing

5.2.1 Rationale

Once the initial topology has been constructed, it is necessary to form the detailed shape of the curves making up the edges of the blocks and to decide on the placement of important points such as where a number of block corners meet. This process is referred to here as topology processing. There is no generally applicable definition of an optimal multiblock grid or topology. Different grids and different topologies

can be used to obtain good results, see for example [131] where various grids and topologies were employed to good effect on the same two-element aerofoil problem. In the absence of a definite objective in optimising the topology, to achieve our goal of obtaining an automatic procedure we instead attempt to model the actions of an experienced grid generation engineer. Topology processing is achieved with modern grid generation packages using an interactive Graphical User Interface (GUI). The GUI enables simultaneous design and assessment of the topology but is very labour intensive. The skill involved is to shape the topology in a manner which will allow the generation of a grid with good characteristics such as orthogonality and smoothness. These qualities are in themselves difficult to define as well as to achieve, which is one reason why grid generation is often referred to as an art as well as a science. An engineer experienced in multiblock grid generation soon recognises certain simple elements to this process however; in this section it is argued that these elements can be formulated in a cost function which can be used to quantify the quality of the topology. To simulate the interactive operations of an engineer the cost function can then be minimised to achieve a topology of good quality. The cost function will be constructed using geometric considerations only. In some cases another factor in grid generation, including the topology design, is the expected behaviour of the flow itself; notably grid lines can be deliberately aligned with streamlines and shock waves. Topology design based purely on geometry will in many cases be sufficient, and at the least will provide an advanced starting point for further modification based on the actual flow.

In Section 5.1.3 it was discussed how an unstructured mesh generation method can provide an initial topology definition. An ideal initial topology generator would produce topologies which would require no processing, this stage could be by-passed and grid generation could proceed directly. Even if the topology generator produced straight-sided blocks, elliptic smoothing could be sufficient to provide a smooth grid especially if a large number of small blocks were used. However, although it is difficult to quantify how much poor quality in a block topology elliptic smoothing can cure, there does not appear to be at present an automatic, unstructured quadrilateral/hexahedral mesh generation method which can deliver the ideal level of topology quality. Even the most promising method available for this application,

the Medial Axis approach discussed in Section 5.1.3, would require significant additional refinement of block edge shape and singular point location, which is too much to demand of elliptic smoothing in the general case.

5.2.2 Curve definitions

In the present study each curve or block edge is defined simply as consisting of straight line segments joining p equally spaced points with index j, see Figure 5.2. To simplify programming, all curves have p points irrespective of the actual curve length or shape. The initial location of the points is obtained by interpolation from the initial topology definition. A straight line segment approach cannot provide a high quality of shape description without using an excessive number of data points. However, since these curves are used here only to define internal block boundaries, onto which a spline can be fitted for algebraic grid generation and across which elliptic smoothing may be employed, such a definition is adequate. Note that although the problem geometry is also represented by straight line segments during the topology processing, the problem geometry is fixed and the original definition can be recalled on proceeding to the grid generation stage.

5.2.3 Cost function

Figure 5.3 shows a multiblock grid for the NLR 7301 wing/flap configuration which has been used in a CFD study where excellent agreement with experiment was obtained[132]. Figure 5.4 shows the topology defined by the block edges, and Figure 5.5 shows a detail of this around the flap. Reference will be made to these figures to help illustrate the objectives of the cost function construction.

In Figure 5.5, there are two points in the vicinity of the flap leading edge where five blocks corners meet. Here the designer must consider how best to set the block corner angles. Structured grid flow solvers give most accurate results when the grid cells are orthogonal since this minimises the truncation error associated with the discretisation. When four blocks meet at a point, as shown in Figure 5.6(a), it is therefore desirable to ensure that the angle θ in the corner of each block is as close as possible to a right-angle. Similarly when three, five or more blocks meet at a

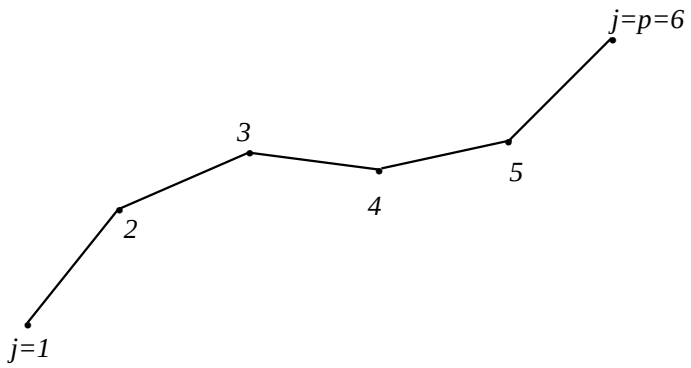


Figure 5.2: *Curve defined as straight line segments*

point, as shown in Figures 5.6(b) and 5.6(c), it is desirable to have the same value for θ in each of the block corners so that no one block corner has cells with a large deviation from orthogonality. In our cost function we therefore penalise deviation of the vertex angles θ_i for each block corner or vertex i. A simple way of achieving this is to write the cost C_v associated with block vertices

$$C_v = \sum_{i=1}^{v} \left(\theta_i - \frac{\pi}{2}\right)^{1.1} \tag{5.1}$$

where v is the total number of vertices. In this way where four blocks meet at a point the cost is zero if all of the block corners form right angles, and the cost increases sharply on deviation from this. Since the exponent is greater than one, when a number of blocks other than four meet at a point the minimum cost is incurred when all the block corner angles are equal. A value of 2 was used initially for the exponent, but the cost incurred when other than four blocks would meet at a point rendered other costs insignificant.

Figure 5.3 shows a grid with good smoothness properties. On examining figure 5.4, it is evident how the shape of the interior block edges follows the shape of the

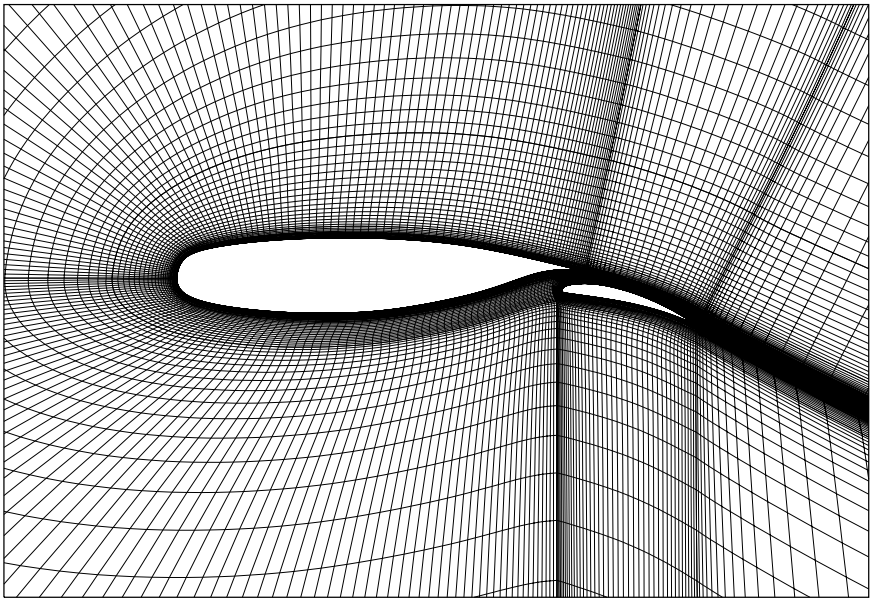


Figure 5.3: *Multiblock grid for NLR 7301 wing/flap configuration*

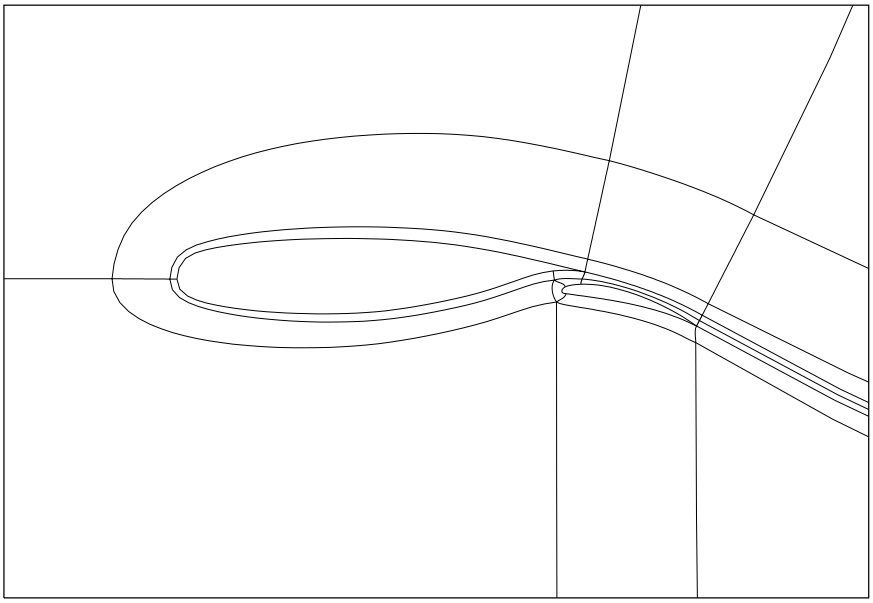


Figure 5.4: *Block topology for NLR 7301 wing/flap configuration*

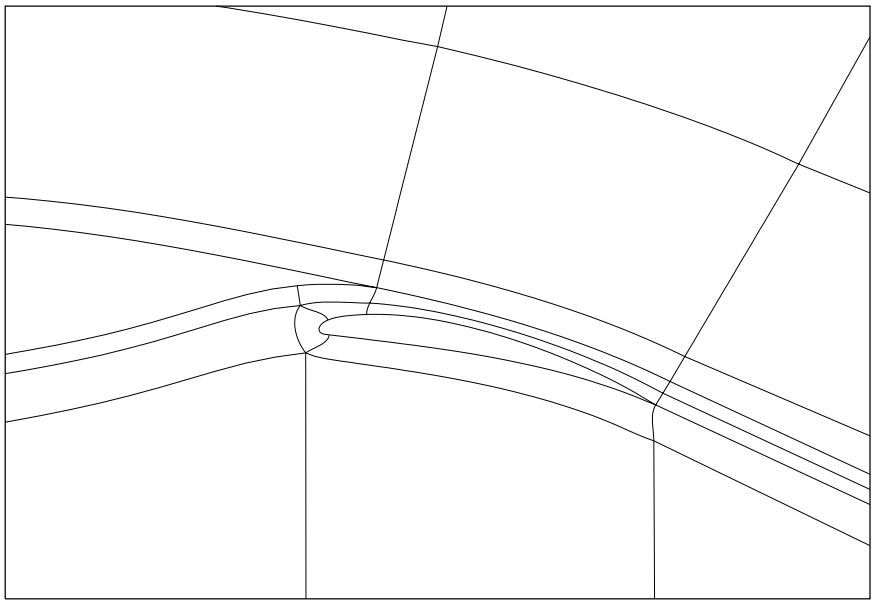


Figure 5.5: *Detail of block topology around flap*

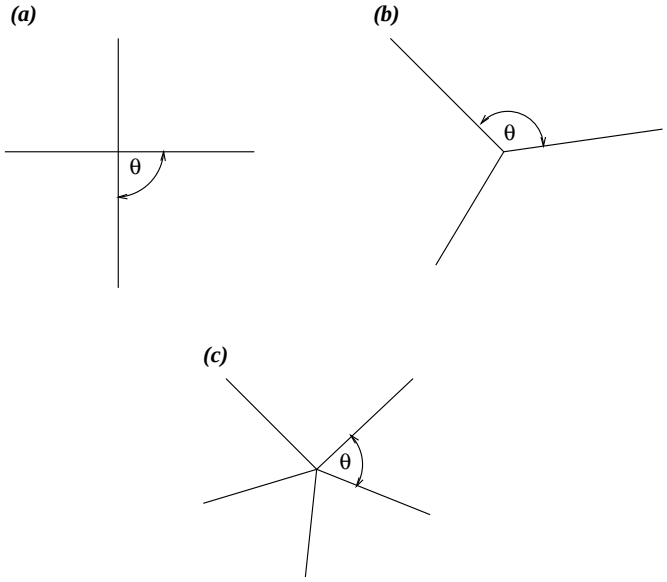


Figure 5.6: *Several block corners meeting at one point*

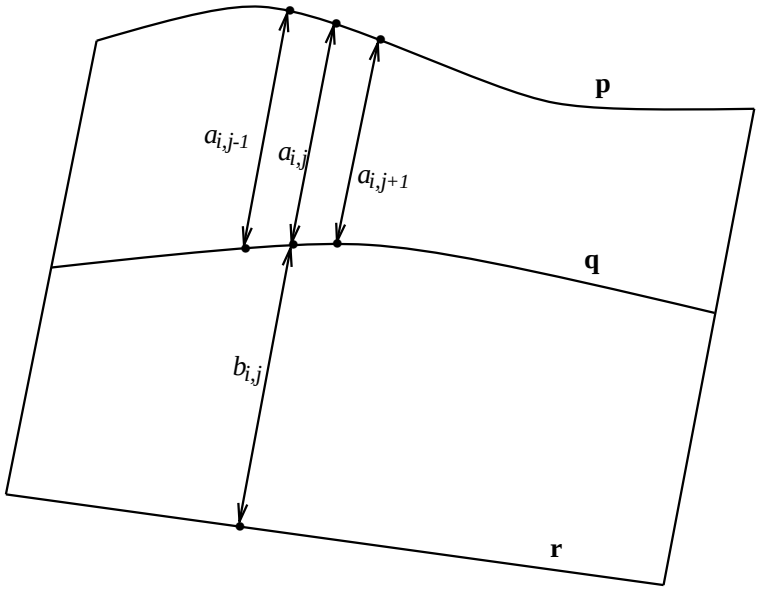


Figure 5.7: *Measurement of shape-following cost element*

aerofoil surfaces to contribute to this smoothness. Grid smoothness is possible only if gradual changes in the curvature of adjacent grid lines are permitted. Consider Figure 5.7 where two blocks are shown which share a common edge **q**. To encourage grid smoothness between the block edges **p** and **r** the shape of **q** will ideally represent a transition from the shape of **p** to **r**. The closer **q** is to **p**, the more closely the shape of **q** should follow that of **p**, and the more the influence of **r** should diminish. A cost element C_s to penalise poor 'shape-following' has been constructed as

$$C_s = \sum_{i=1}^{c} \sum_{j=2}^{p-1} \left\{ \frac{\overline{b}_i}{\overline{a}_i + \overline{b}_i} \left(\frac{A_{i,j}}{\overline{a}_i} \right)^2 + \frac{\overline{a}_i}{\overline{a}_i + \overline{b}_i} \left(\frac{B_{i,j}}{\overline{b}_i} \right)^2 \right\} \tag{5.2}$$

$$A_{i,j} = a_{i,j+1} - 2a_{i,j} + a_{i,j-1}$$

$$B_{i,j} = b_{i,j+1} - 2b_{i,j} + b_{i,j-1}$$

where c is the total number of curves. On each curve i there are p equally spaced points. The distance from the point j on the curve i to the corresponding point on an opposing block face is labelled $a_{i,j}$ as indicated in Figure 5.7. The quantity $A_{i,j}$ is

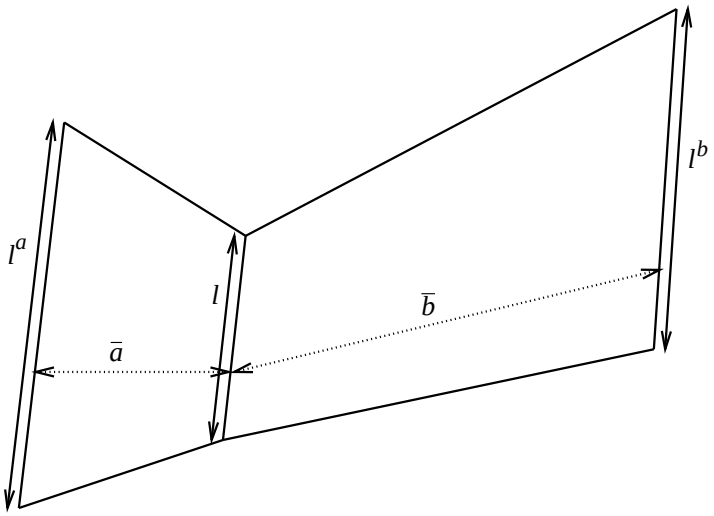


Figure 5.8: *Measurement of block expansion cost element*

therefore a measure of how well the local curvature of $\mathbf{p}$ is following that of $\mathbf{q}$. This is summed over the length of the curve. Each curve in the interior of the domain (i.e. each curve that does not define a fixed geometry) has two opposing counterparts; $a_{i,j}$ and $b_{i,j}$ are the distance measures to each. To ensure greater influence of curves in close proximity, the influence on curve i of each opposing curve is scaled by their average separations $\overline{a}_i$ and $\overline{b}_i$ from i, defined as

$$\overline{a}_i = \frac{1}{p}\sum_{j=1}^{p} a_{i,j} \tag{5.3}$$

$$\overline{b}_i = \frac{1}{p}\sum_{j=1}^{p} b_{i,j} \tag{5.4}$$

The construction of a cost function element to model shape-following is not straightforward. The engineer with experience of multiblock grid generation can readily recognise when blocks are well shaped, but how to define what this means in terms of gradients, curvatures etc. is not obvious. The cost element (5.2) tries to match local curvature. The definition of the gradients at the curve ends is then important

to the success of the method. A previous attempt at constructing a cost element was based on local gradient rather than curvature. Referring to Figure 5.7, this worked very well for cases where the curves **p** and **r** have approximately the same orientation as **q**, but becomes a poor measure of shape-following otherwise.

In Figure 5.4 the blocks are fairly regularly shaped in that none of the blocks expand in size very sharply. If a block expands too sharply, then cell orthogonality and grid smoothness can be adversely affected in the block interior. Figure 5.8 shows two blocks sharing a common edge of length l. The opposing edges have lengths l^a and l^b. A cost element C_e to penalise sharp block expansions has been constructed as

$$C_e = \sum_{i=1}^{c} \left[\frac{(l_i - l_i^a)^2}{l_i \overline{a}_i} + \frac{(l_i - l_i^b)^2}{l_i \overline{b}_i} \right] \tag{5.5}$$

The total cost C_t associated with the quality of the topology can then be written as

$$C_t = k_v C_v + k_s C_s + k_e C_e \tag{5.6}$$

where k_v, k_s, k_e are positive constants which define the relative importance of the cost elements. Appropriate values for these constants were found by experimentation using simple model cases and verification on more complex cases, see Section 5.2.4.

5.2.4 Cost function minimisation

Equation (5.6) defines a cost function which measures the quality of a multiblock topology. This cost function is minimised in order to obtain a topology of good quality. The resulting topology is referred to as the 'processed' topology. To do this, a straightforward iterative improvement technique is employed. The number of points p defining each curve is chosen as the minimum number which give a suitable definition of the problem geometries, typically between 8 and 40. A point on one of the c curves is chosen at random. Two random numbers between -1 and 1 are multiplied by the pre-defined maximum displacement distance d_{max}, and the selected point is displaced in the x and y directions by each result respectively, remembering that a point may belong to more than one curve. Points on curves defining the domain boundaries, i.e. on "exterior" edges, are not permitted to

move. If the total cost of the modified topology has been reduced then the move is accepted. Otherwise the move is rejected and the former position of the displaced point is recalled. A large number Nc of trial moves are attempted, N being some large integer.

In the cost elements (5.2) and (5.5) the quantities $\overline{a}_i$ and $\overline{b}_i$ are used as scaling factors. In implementing the cost function minimisation procedure, care must be taken to ensure that the block shapes are not being inadvertently altered to maximise these quantities in order to minimise the cost (they are on the denominator). To achieve this they are evaluated infrequently, every 100 successful trial moves.

It is well known that simple iterative improvement does not provide a mechanism for avoiding local minima. Careful selection of the trial moves can help avoid this problem. Trial moves of curve sections as well as single points were employed. Although this helps to avoid local minima to some extent, this simple approach to cost function minimisation could be improved upon, as will be discussed in Section 5.3.1. It is considered sufficient however for the task of demonstrating the general method. As will be demonstrated below for a number of test cases, iterative improvement has succeeded in finding a good enough local minimum where the block topology properties have clearly been improved in terms of preparation for the grid generation stage.

5.2.5 Calibration test cases

A simple test case was constructed, consisting of two blocks sharing a common edge, in order to find appropriate relative magnitudes of k_v and k_s in equation (5.6). For these tests k_e was set to zero. Figure 5.9 shows some representative results for a number of cases where $k_s = 1.0$ and the magnitude of k_v was varied. The curve definition $p = 10$ was used. With $k_v = 0.0$ the shape-following cost is the only non-zero part of the cost function. As expected the shape of the resulting curve lies somewhere between the straight line of the left-hand opposing curve and the greater curvature of the right-hand opposing curve. As k_v is increased the tendency for the ends of the curve to form right angles at the block corners increases, eventually to the detriment of the overall shape. A good compromise is found at values around

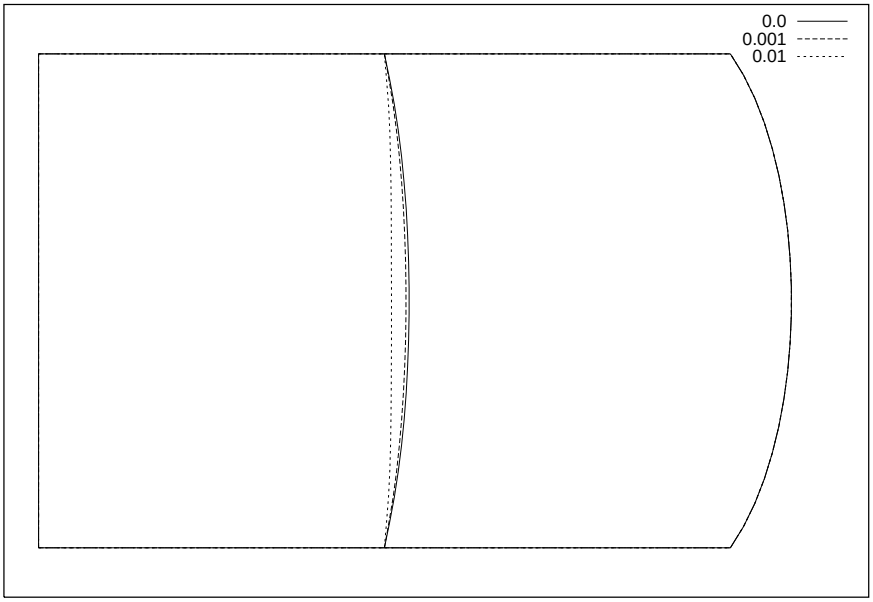


Figure 5.9: *Test case to find value for k_v*

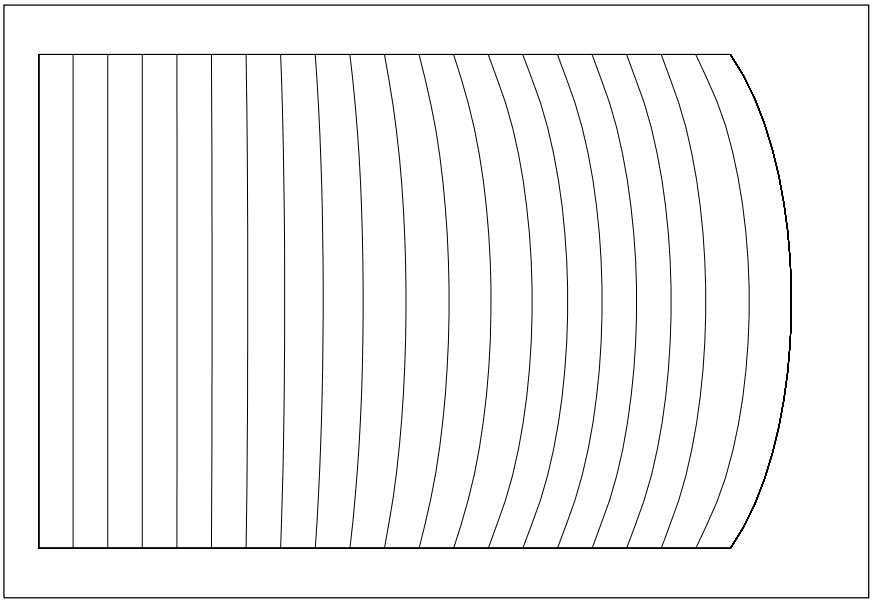


Figure 5.10: *Test case to check shape-following*

$k_v = 0.001$ (with $k_s = 1.0$), at which condition the effect of the cost associated with corner angles becomes noticeable. For this case the final cost becomes converged to three significant figures for $N = 15000$. A set of similar tests was carried out to ensure that the proximity of opposing block edges has the desired influence on the shape-following cost. Keeping $k_v = 0.001$ and $k_s = 1.0$, the location of the common edge was varied; the results are shown in Figure 5.10. Note that the original result with a central common edge is shown with the other results superimposed. There is a smooth transition in curve shape as required.

An eight block grid for a single element aerofoil was used to determine a suitable value for the block expansion cost coefficient k_e. Figure 5.11 shows the initial topology, taken from a grid known to be of good quality which has been used successfully in a CFD study[33]. The figure also shows a processed topology obtained by setting $k_s = 1.0$, $k_v = 0.001$ and $k_e = 0.0$. For this case the curve definition $p = 20$ was used and the final cost becomes converged to three significant figures for $N = 20000$. The result obtained using $k_e = 0.0$ is satisfactory in this case since the initial topology used does not contain blocks with an unacceptable block expansion rate. However, following the same approach as for k_v, gradually increasing the value of k_e should indicate a value where the block expansion cost element begins to have an effect but is not yet dominating the other cost elements. Figure 5.12 shows the effect of varying the value of k_e. The block expansion cost element begins to take effect for values of k_e around 0.001; in the figure for this value the block edge emanating from the aerofoil leading edge has been stretched slightly to match the length of the block edges emanating from the trailing edge. For lower values of k_e there is no effect, and for higher values the block expansion cost begins to swamp the other cost elements, as shown in the figure where the processed topology for $k_e = 0.01$ has poor shape-following and block corner angle charactersistics. These two examples have indicated appropriate values for the coefficients in equation (5.6) and demonstrated that the method works well for simple cases. Encouraged by this, the method will now be applied to other existing multiblock topologies from real problems, in order to examine how the method performs on topologies which are known to be already of good quality and to check that no deleterious effects are experienced, before moving on to more realistic test cases. The same coefficients will be used throughout as

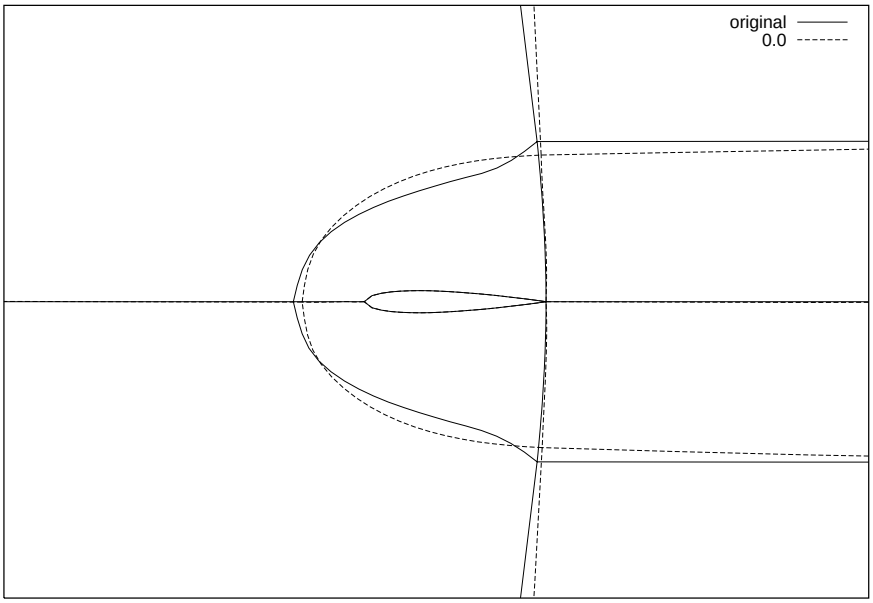


Figure 5.11: *Block expansion test case, initial topology and processed topology with* $k_e = 0.0$

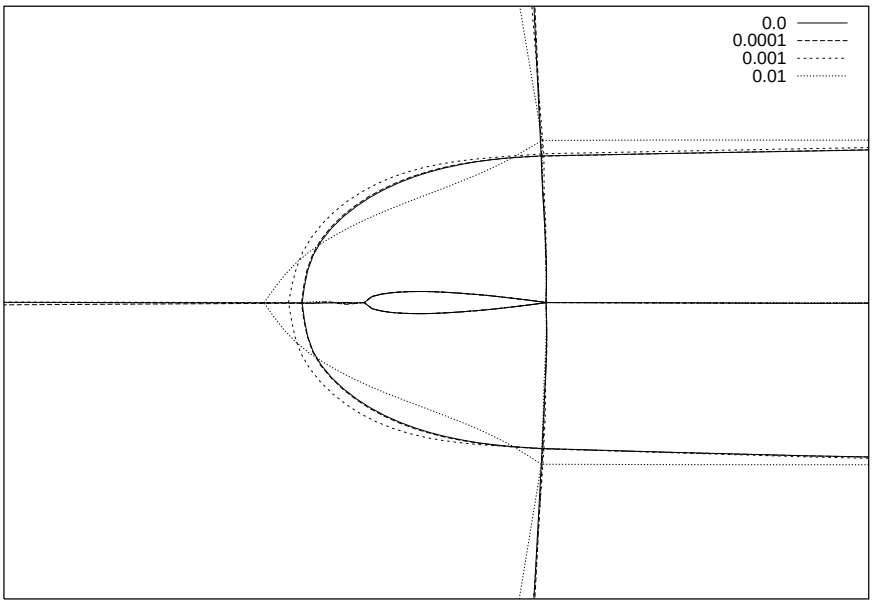


Figure 5.12: *Processed topologies with various values of* k_e

were used for the example test cases ($k_s = 1.0$, $k_v = 0.001$, $k_e = 0.001$) in the hope that their values will be case independent.

5.2.6 Existing topologies

A simple and common multiblock topology is a three block grid around a single element aerofoil. The same grid as used above for the calibration case was also used in three-block form. Figures 5.13 and 5.14 show the original and processed block outlines for this case. The curve definition $p = 40$ was used and the final cost becomes converged to three significant figures for $N = 10000$. The topology processing method has improved the block corner angles at the trailing edge and maintained a satisfactory shape for the interior block edges.

Figures 5.15 shows the original and processed topology for a grid used in a nozzle/plume study[48]. For this case the curve definition $p = 10$ was used and the final cost becomes converged to three significant figures for $N = 15000$. Again the topology processing method has improved the block corner angles, quite significantly changing the shape of one curve, but a satisfactory trade-off between orthogonality and curve smoothness/shape-following has been achieved.

Figures 5.16 shows the original and processed topology for a grid used in a cavity flow study. The cavity has a right-angled leading edge and a radiused trailing edge, the novel topology created for this configuration is a good example of how some imagination can be required to create a topology suitable for even simple configurations. For this case the curve definition $p = 10$ was used and the final cost becomes converged to three significant figures for $N = 20000$. The topology processing method has again significantly altered the shape of one of the curves in order to improve block corner angles.

Figure 5.17 shows the multiblock topology for a multi-element aerofoil grid from British Aerospace which has been used in a CFD study of a high-lift configuration where good agreement with experiment was achieved[133]. The large number of blocks required for even moderately complex configurations (81 in total for this grid) is evident from the figure. The result of the topology processing procedure is shown in Figure 5.18. For this case the curve definition $p = 30$ was required and

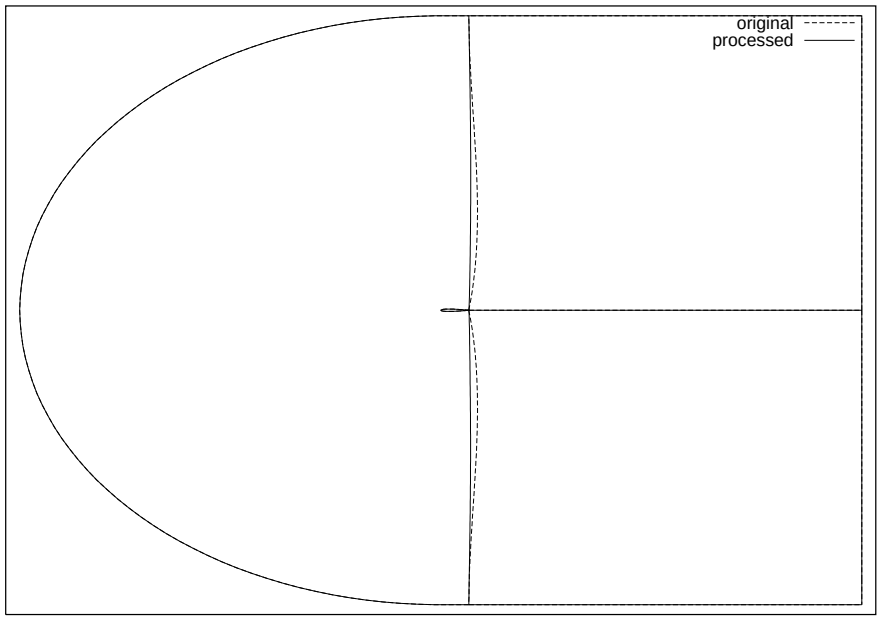


Figure 5.13: *Three block single element aerofoil grid, entire domain*

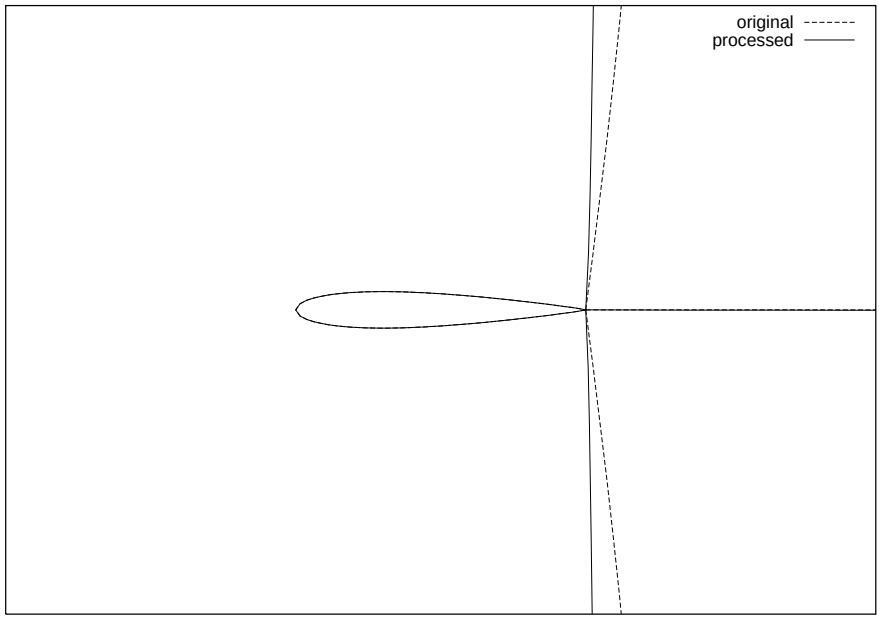


Figure 5.14: *Three block single element aerofoil grid, detail*

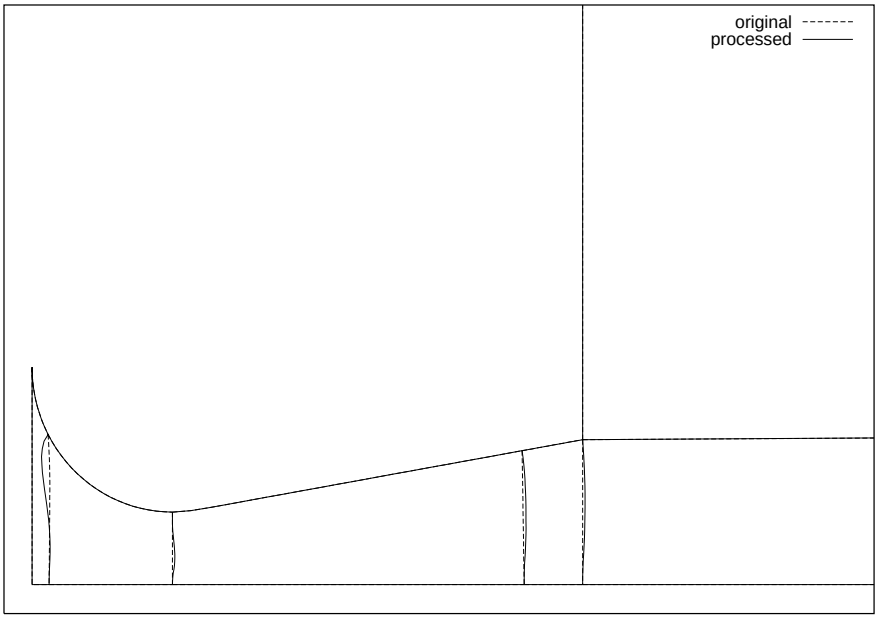


Figure 5.15: *Nozzle/plume grid*

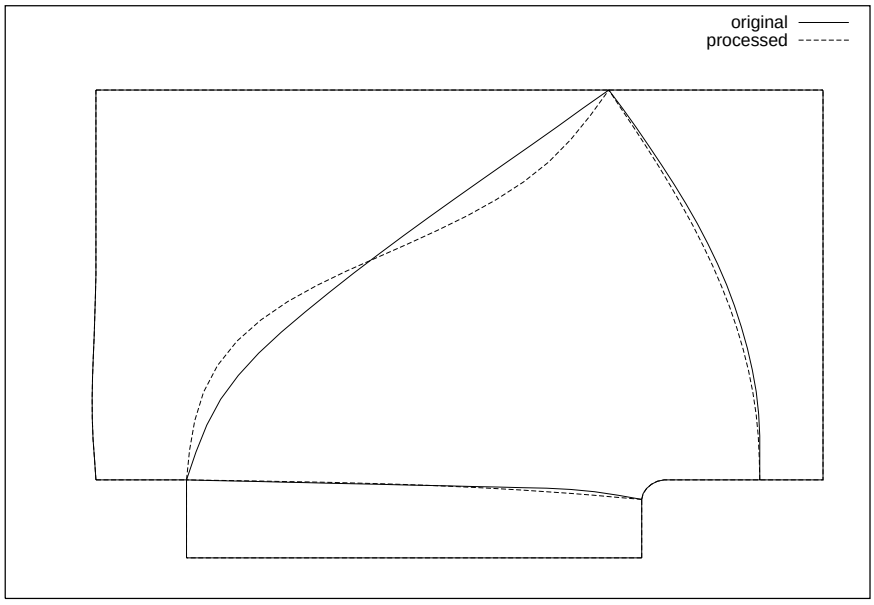


Figure 5.16: *Cavity flow topology*

the final cost becomes converged to three significant figures for $N = 20000$. There is very little room for improvement from the initial excellent configuration, the only real difference is an improvement of the block corner angles, most notably at the point where five blocks meet below the forward part of the main element.

5.2.7 Marine application example

The topology processing method has been applied successfully to simple test cases in Section 5.2.5 and to real problems where the topology is already of good quality in Section 5.2.6. The main aim of this work is to produce a topology processing method applicable to the inevitably unrefined initial topologies which can be generated using unstructured quadrilateral grid generation techniques, see section 5.1.3. A coarse, straight-sided topology has been created manually for a model marine application. This is a demonstration case in order to simulate the result of such an automatic topology generation method, see Figure 5.19. The corresponding processed topology is shown in Figure 5.20. For this case the curve definition $p = 10$ was used and the final cost becomes converged to three significant figures for $N = 40000$. The initial configuration has been improved considerably; the blocks have good orthogonality characteristics, do not expand rapidly and conform well with the geometry and each other.

An unstructured quadrilateral grid for this geometry has been obtained from the Finite Element Modelling Group at the Queen's University of Belfast. The Medial Axis approach, described in section 5.1.3, was used to automatically generate the initial topology shown in Figure 5.22. For this case the curve definition $p = 10$ was also used, and the cost becomes converged to three significant figures for $N = 50000$. Again the initial configuration has been improved considerably. For this example, the assertion that the automatic unstructured mesh generation technique produces suitable initial multiblock topologies has been affirmed.

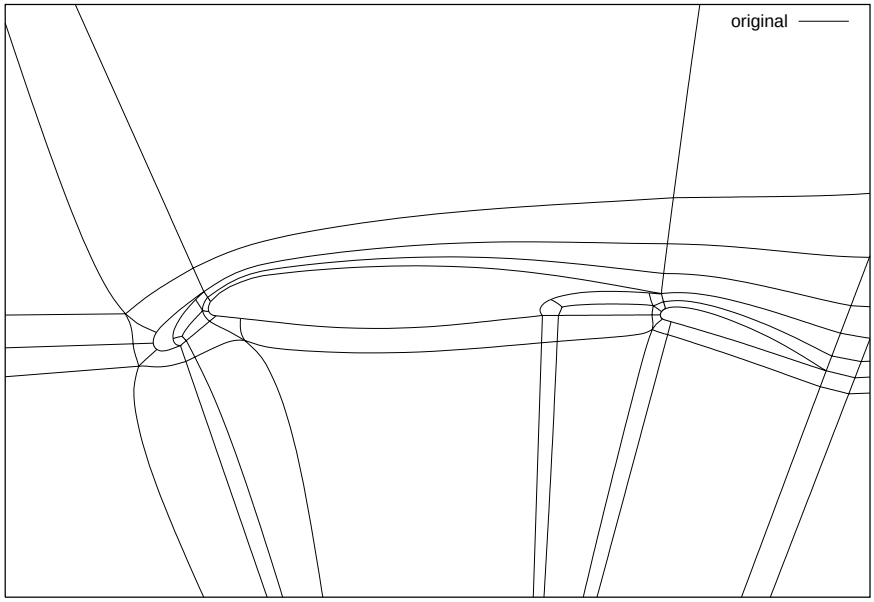


Figure 5.17: *Original multi-element aerofoil topology*

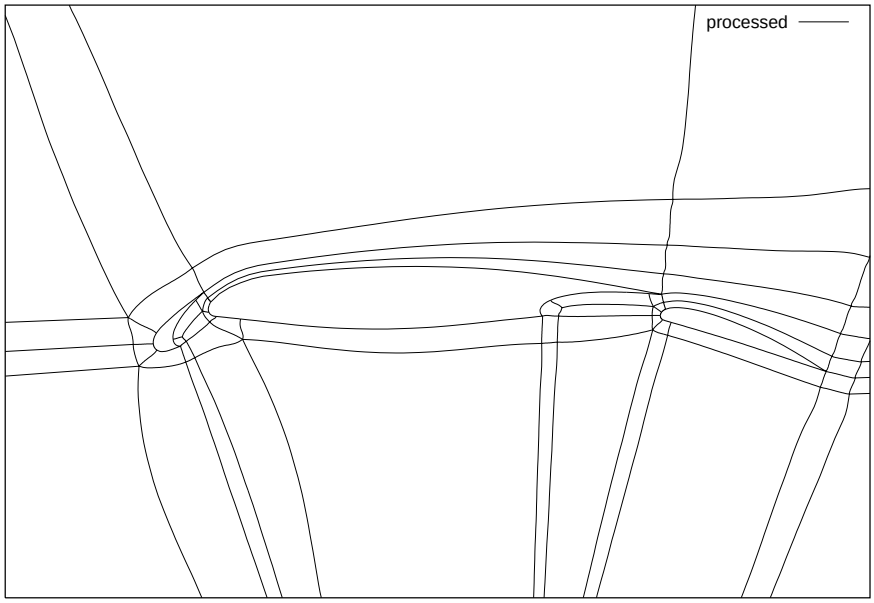


Figure 5.18: *Processed multi-element aerofoil topology*

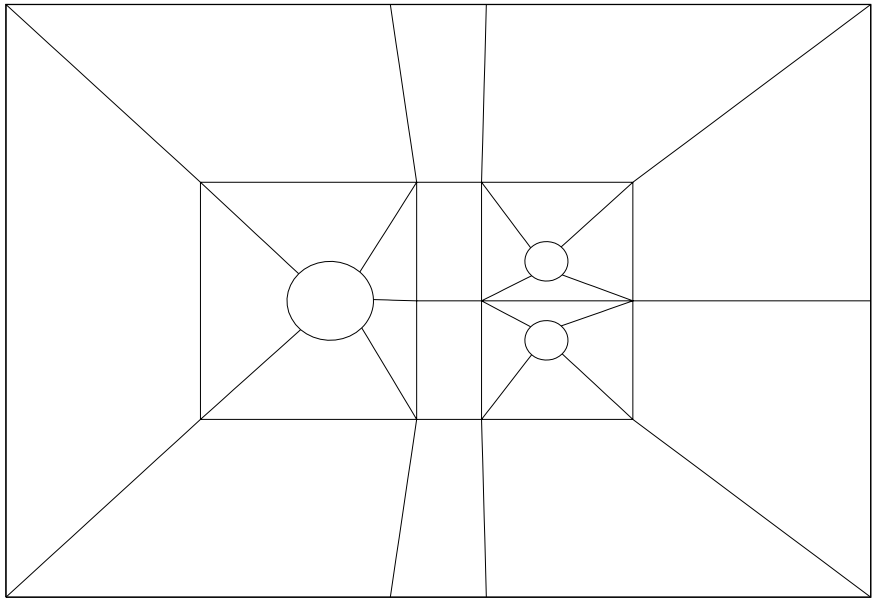


Figure 5.19: *Initial topology A, marine application example*

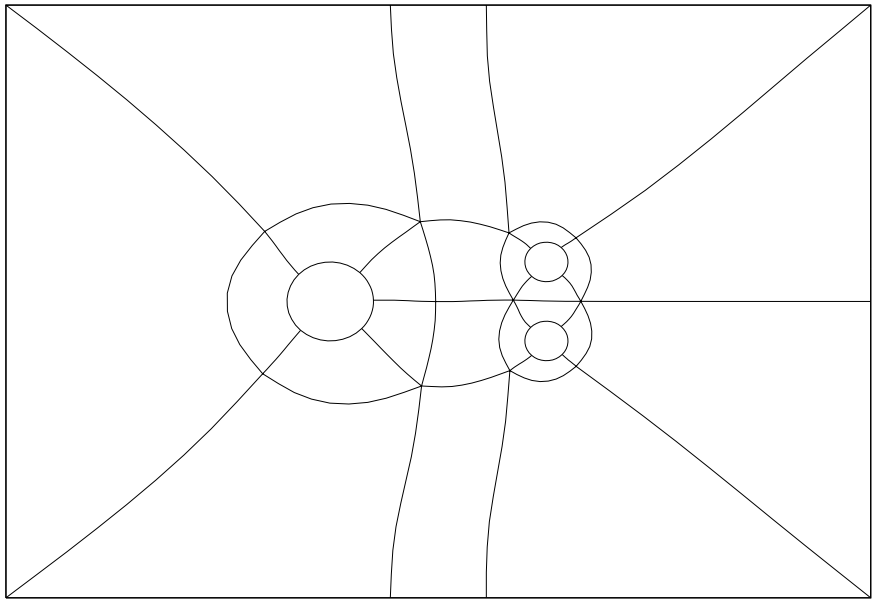


Figure 5.20: *Processed topology A, marine application example*

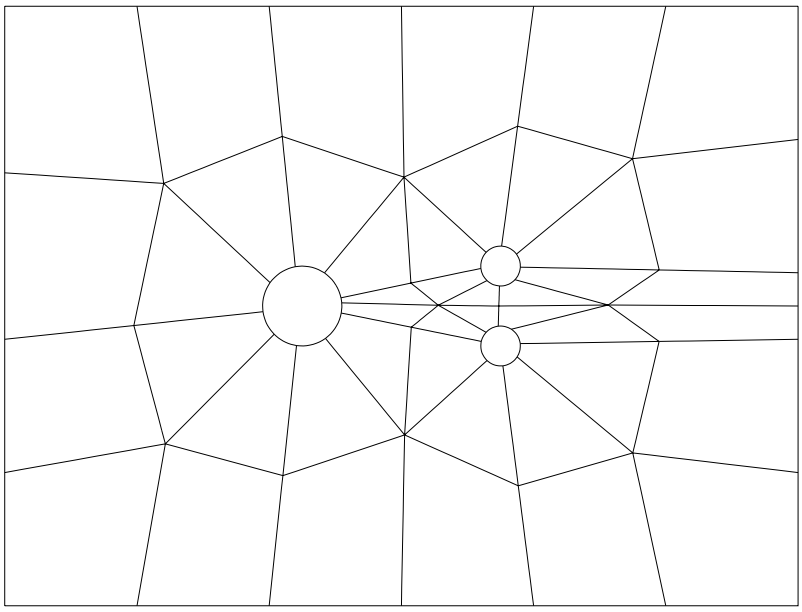


Figure 5.21: *Initial topology B, marine application example*

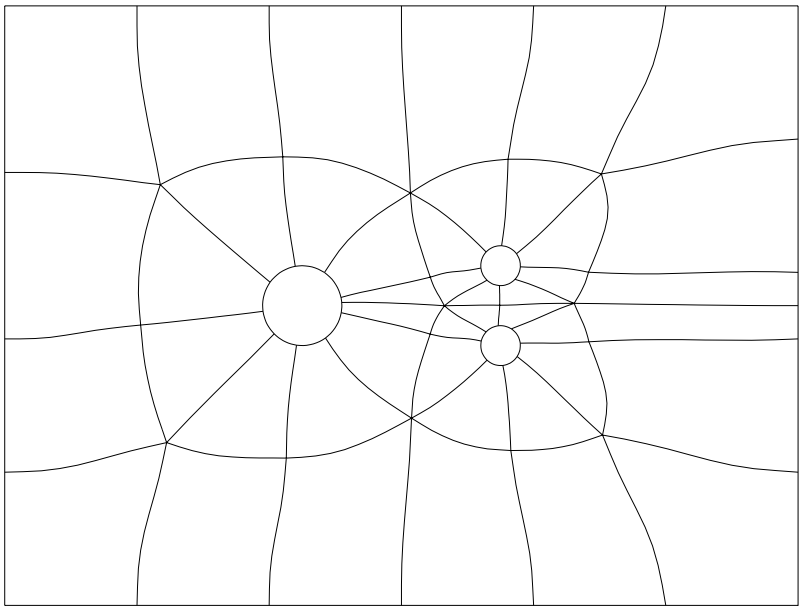


Figure 5.22: *Processed topology B, marine application example*

5.2.8 Two-element aerofoil example

To investigate a further example using initially poor topologies, two coarse straight-sided topologies have been created manually for a two-element aerofoil demonstration case in order to simulate the result of such an automatic topology generation method, see Figures 5.23 and 5.24. The Williams B aerofoils[134] are used in both cases. The corresponding processed topologies are shown in Figures 5.25 and 5.26. The initial topology A (Figure 5.23) has the agreeable feature of well located block corners. To modify this topology to obtain a form suitable as a basis for the actual grid generation phase involves changing the shape of the block edges to a smoother, more geometry conforming pattern. This has been achieved by the present topology processing method, see Figure 5.25. The initial topology B has the additional problem of an irregularly shaped block at the leading edge of the 'flap'. The topology processing method has also coped with this well, see Figure 5.26, by drastically reducing the lengths of the long sides of the block at the nose of the flap. Both results from the topology processing method could be used as inputs to the grid generation proper stage. The method has been successful in finding a compromise between smoothing the initial configuration, maintaining reasonable orthogonality and resizing blocks which expand too sharply. It is noted however that the final configurations are different, so the minimisation method has clearly not found a global minimum. This issue will be discussed in Section 5.3.1.

5.3 Problems encountered and future work

5.3.1 Global Minimum

The present method does not find a global minimum for the multi-element aerofoil case. This is not surprising given the very simple minimisation procedure employed. The local minima obtained for the cases examined here are satisfactory, but there is doubt whether this will be generally true. A straightforward extension of the iterative improvement technique is simulated annealing[135] which is well known to obtain near-optimal results for a broad range of minimisation problems. A drawback to this method is that it necessitates additional computational effort; the present

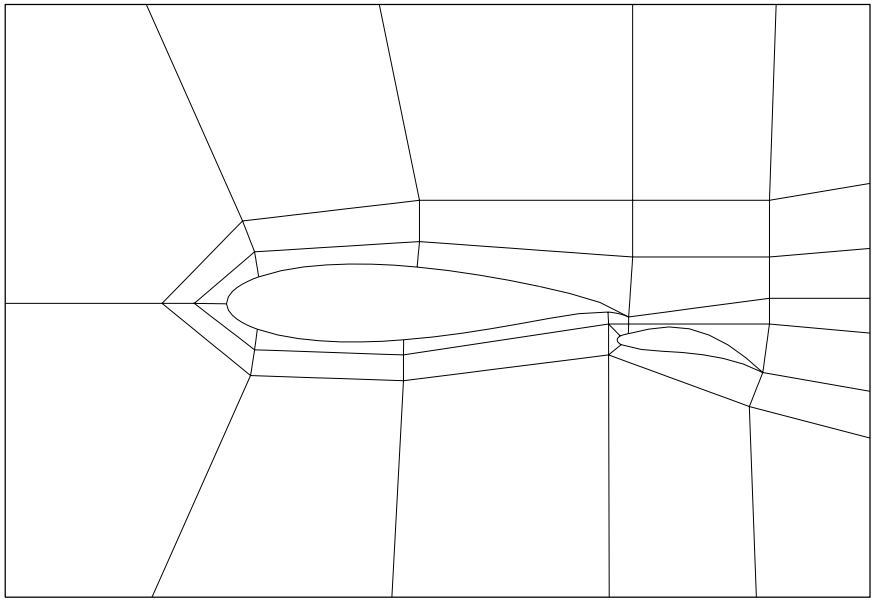


Figure 5.23: *Initial topology A, two-element aerofoil*

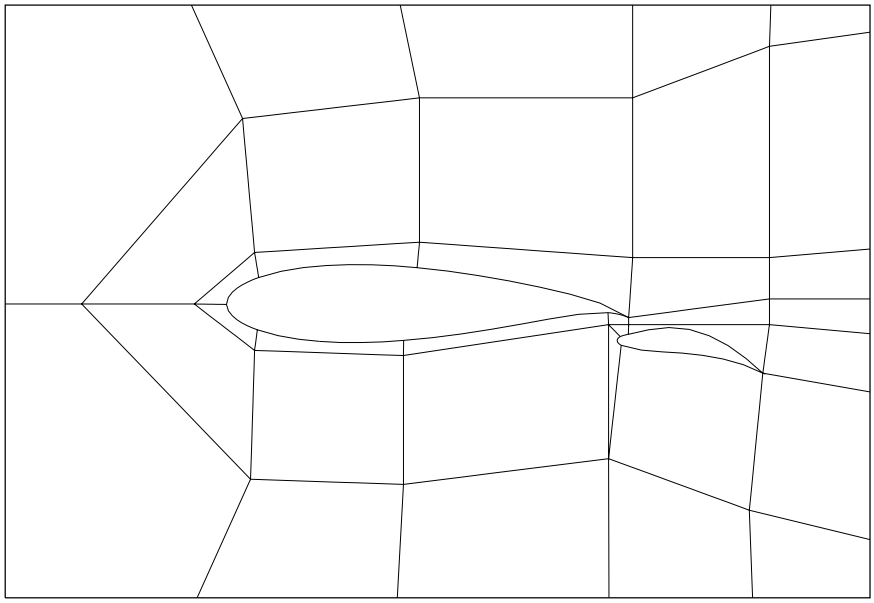


Figure 5.24: *Initial topology B, two-element aerofoil*

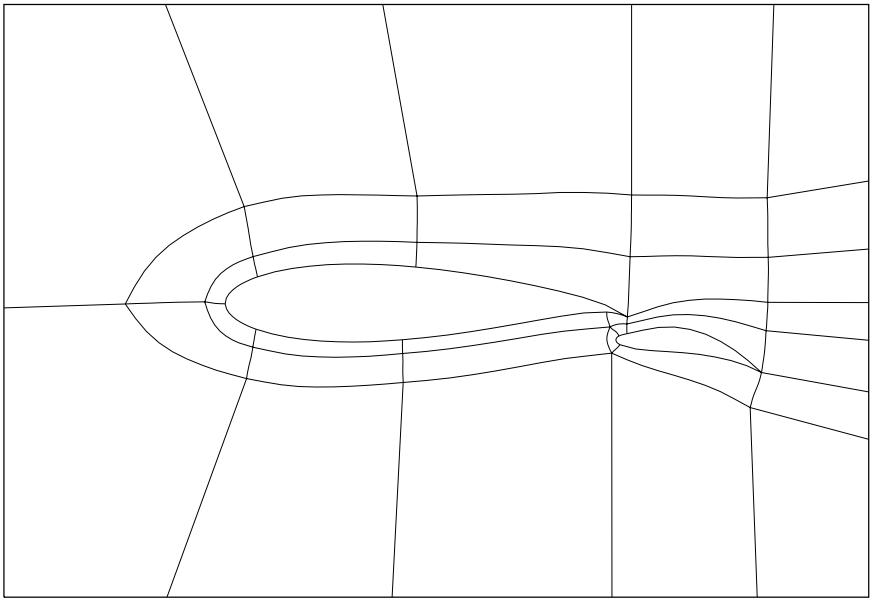


Figure 5.25: *Processed topology A, two-element aerofoil*

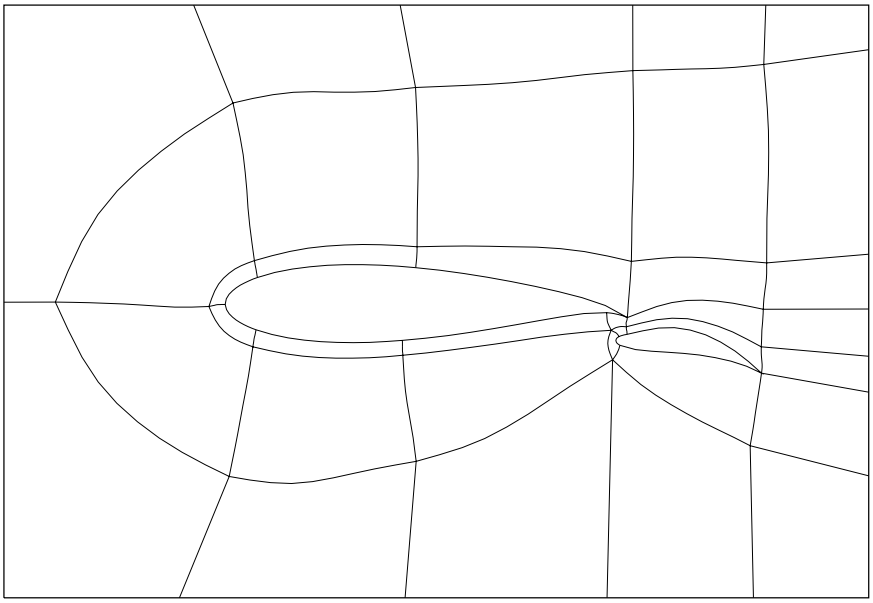


Figure 5.26: *Processed topology B, two-element aerofoil*

method already requires a substantial amount of computing time, the two-element aerofoil example requiring approximately half an hour using a desktop PC. Ultimately the most promising direction is likely to be to begin with a higher fidelity curve description (for example using B-splines) to reduce the number of data points and hence operations, and using a more advanced minimisation procedure, perhaps again borrowing from structural mechanics where there are established techniques for shape optimisation in component design.

5.3.2 Curve definition refinement

As noted directly above, a reduction of the computational time required for the process is desirable. If a small number of points is used in the curve definition, then the number of operations necessary to evaluate the cost function (and hence computational time) is reduced. However, often a finer definition of the curves is required to adequately represent the problem geometry. With this in mind, a curve refinement approach was adopted. The initial curve definition could be fairly coarse, and after a number of trial moves the curve definition would be successively refined. This approach did reduce the overall time required to obtain a converged solution in some cases, but was not successful generally. In Figure 5.11, the curves are defined using 20 points connected by straight line segments. These segments are small enough to represent the strong curvature at the leading edge. When the refinement approach was attempted for this case, a coarser curve definition misrepresents the leading edge curvature, introducing unwanted features into the curves attempting to follow the aerofoil surface. These features must then be removed by applying a large number of trial moves in the later stages when the curve definition becomes adequate.

5.3.3 Generality

The same cost function and cost function coefficients were used to process all the test cases presented. This provides some evidence that the method may be generally applicable, but realistically a far larger number of test cases from a greater range of problems should be examined before being able to state this confidently. A particular

question is how the method will perform using large numbers of small blocks; all of the test cases considered had a relatively small number of large blocks.

5.3.4 Automatically generated topologies

This report has concentrated on demonstrating the potential of an automatic topology processing method, which has been examined essentially in isolation from the other elements of the multiblock grid generation process. The next step should be to examine whether the method can fulfil its potential by linking with the other elements, see section 5.1.2. There is little doubt that algebraic grid generation and elliptic smoothing performs well when based on a sound topology. The main question is how well the processing method would perform given automatically generated initial topologies. Only one example has been presented here; although the results are encouraging, the next step should be to use the Medial Axis unstructured mesh generation method to create numerous initial topologies for the processing stage. In this way, the assertion that the approach can produce multiblock topologies possessing the required characteristics can be verified.

5.3.5 Extension to 3-D

In three dimensions the problem of multiblock grid generation is more demanding and the need for automation is even greater. As an example, a computational aerodynamic analysis, using the multiblock method, of the flow around a wing with high lift devices is feasible using present technology. Suitable flow solution methods are available, and sufficiently powerful parallel computers are becoming available. However, the enormous amount of time and effort that would be required to interactively generate a multiblock grid precludes the use of CFD as a design tool for problems with this level of geometric complexity. The present topology processing method generalises to three-dimensions; the cost function is based on the shape of block edges and the angles of block corners only. The method suggested in section 5.1.3 for the initial topology generation has already been extended successfully to three dimensions. Thus there is a clear path to extending the overall approach to three-dimensional problems, where an automatic grid generation method could not

only accelerate the analysis process, but also allow problems to be tackled which were previously considered as being prohibitively complex.

5.4 Conclusion

A new approach for automatic multiblock topology processing has been presented. A cost function which evaluates the quality of a multiblock topology has been created. The elements of the cost function are based on the objectives of the multiblock grid generation software user when interactively constructing the topology. A simple minimisation procedure is employed to obtain a topology of good quality. The potential of the method has been demonstrated using a number of test problems. It has been suggested that full automation of the entire multiblock grid generation procedure is possible using in sequence an existing unstructured grid technique to obtain an initial topology, the present processing method, then conventional algebraic grid generation and elliptic smoothing. Problems encountered during the study and future work have been discussed. The potential for extending the method to three dimensional problems shows considerable promise.

Chapter 6

Parallel Aerodynamic Simulation on Open Workstation Clusters

6.1 Introduction

Parallel computing in computational fluid dynamics is a very broad area of current research and development. Parallel computing software and hardware technology is developing very rapidly, and the CFD community is at the forefront in exploiting emerging technology to obtain the high performance computational resource required to solve large CFD problems. The enthusiasm for parallel computing in the CFD community is based on present cost effectiveness compared to conventional computing, and future projections of enormous computing power. The exploitation of parallel computing is considered to be a key to tackling the grand challenges in CFD[22].

To effectively use a parallel computer an intelligent mapping of subsets of the total computational work onto processors must be performed. There are several different levels of parallelism, ranging from job parallelism where processors execute tasks with no interdependency, to arithmetic parallelism where the work of the simplest operations is shared amongst processors. A coarse-grain data parallel approach[29] is usually employed in parallel CFD, where sub-domains of the computational grid are mapped onto the set of processors, with the objective of finding a mapping which results in the fastest overall execution of the parallel task. This

approach is commonly referred to as *domain decomposition* in the literature. The principal feature of an efficient domain decomposition is that the load is evenly distributed across the processors. A typical parallel CFD application involves a communication phase where information must be passed between the processors. Communication is necessary periodically, e.g. once every time step. If the load is evenly distributed then the processors arrive at the communication phases simultaneously, minimising processor idle time. For many applications, attempting to minimise the time spent in the communication phase is also necessary for efficient implementation. The problem of optimal domain decomposition is well known to be NP-complete[136],[137], i.e. a deterministic solution procedure is impractical. The task of achieving a parallel execution via domain decomposition can be viewed as a two-stage process; *mesh partitioning* to form the sub-domains and *allocation* of sub-domains to processors to achieve load balanced execution[136],[138]. A wide variety of methods have been proposed, see for example the proceedings of the Parallel CFD conferences[27],[139], reflecting the variety of problems considered and architectures used. For unstructured grid problems the prevalent approach is to use a mesh partitioning heuristic to obtain equally sized sub-domains and at the same time attempt to minimise the sub-domain interface length to keep down the amount of necessary communication. The resulting partition then consists of the same number of sub-domains as there are processors, and communication has already been considered implicitly in the partitioning stage, so it is sufficient to allocate the sub-domains directly onto the processors. An initially popular method was the 'Greedy' algorithm for mesh partitioning[140], so called because successive 'bites' are taken from the domain. The Greedy algorithm is very fast since it essentially involves only one sweep across the mesh, but is unreliable since the last 'bites' can leave sub-domains of inappropriate size and shape. Most researchers now employ a recursive bisection approach from graph theory, a good review of which is provided in [141]. In recent years some specific methods have become established in the CFD and structural finite-element communities and are available in the public domain[142],[143],[144]. Alternative non-deterministic approaches such as simulated annealing and stochastic evolution have been used for unstructured mesh partitioning, but have the disadvantage of being slow in comparison to recursive bisection methods[137],[138],[145].

Applying the methods of unstructured grid partitioning to multiblock structured grids is often quoted as being possible, but only one example has been found in the literature[146]. This is for two reasons. First, partitioning a multiblock structured grid is easier than an unstructured grid in that there are less possible boundary path permutations, but harder in terms of programming in that flow solver constraints (e.g. block interface matching) must be considered in the partitioning algorithm. Secondly, often the number of grid blocks naturally arising from the grid generation process is far greater than the number of processors, so this partition can be accepted as long as a heuristic is designed to arrange these blocks onto the processors such that the load is balanced. If there are very large blocks which impede a good load balance then it is a simple matter to split them 'manually', unlike unstructured grids. Hence for structured multiblock grids the emphasis in domain decomposition is much more on the allocation stage. The heuristic techniques employed, often cost function minimisation procedures, are similar to those attempted for unstructured mesh partitioning, but are better suited for this problem due to the reduced size of the state-space[136]; tens or hundreds of blocks are considered rather than tens or hundreds of thousands of grid cells. See references [136],[138] for a summary of the preferred methods.

The domain decomposition methods mentioned above have all considered the *static* problem, where the decomposition is determined before run-time. *Dynamic* re-allocation methods have not been discussed. It is necessary to reconsider the decomposition during run-time to preserve load balance if the solution procedure is adaptive, for example when adaptive grids are used. Also, some researchers seeking the last percentages of parallel performance gains maintain that a static decomposition can never exactly account for actual processor speeds and communication costs, so some degree of dynamic re-allocation is required. For an overview of this type of dynamic problem see references [26],[27]. We are interested here in a different type of dynamic problem where the decomposition may have to respond to varying processor loads; this point is returned to below.

Compared to a decade ago, parallel CFD technology is considerably more advanced. However, as noted by Knight[26], the huge amount of publications devoted to parallel CFD research is not matched by the amount of CFD research conducted

using parallel CFD as a tool. Based on contacts between the CFD group at Glasgow University and the U.K. aerospace industry, this appears to be as much the case for CFD use in industry as in academia. Knight suggests three reasons for this:

- parallel computers are perceived as lacking a decisive performance advantage
- parallel code has portability problems
- parallel code is difficult to program efficiently

Advances in hardware and software have now made the first two points an irrelevancy. Numerous recent projects have demonstrated the enormous potential and cost savings of using workstation clusters or modern commodity processors in parallel, for example[147]. The development of standards in languages (eg. High Performance Fortran[148]) and message passing (eg. MPI[149], PVM[150]) have brought the portability of parallel code almost to the same level as sequential code. The problem lies in the third point; the practical difficulties in making parallel CFD work can be discouraging[151]. To aerodynamicists, there has always been a trade-off between the amount of effort necessary to apply a prediction method and the accuracy of the results that the method produces. In addition to the effort required for a sequential CFD capability, parallel CFD requires the aerodynamicists to:

- obtain and install a message passing library or parallel compiler
- write the parallel code
- write a domain decomposition method or assimilate an 'off the shelf' method
- manage the execution of parallel tasks

The first two points are mitigated by the emergence of standards in parallel programming, as mentioned above, where message passing libraries are in the public domain, parallel compilers are available from vendors, advances have been made in making parallel programming easier and help on all of these is freely available via the internet. However, it is noted that large organisations are likely to employ specialist programmers and information technologists; small and medium-sized organisations are more likely to be discouraged by the first two points. Chien et al.[152] have

made some important observations concerning the third and fourth points. Existing domain decomposition methods are restricted to parallel systems consisting of a homogeneous processor set[1] and which are operated in single-user mode. This typifies a dedicated parallel machine possessed by a large organisation; smaller organisations are likely to make their first steps in parallel processing using a non-dedicated heterogeneous network of workstations. Making use of spare capacity on existing UNIX workstations, originally obtained for other purposes, was pioneered by Pratt & Whitney[154] and McDonnell Douglas. However in these cases a policy of interactive/sequential and batch/parallel use segregation was enforced, the parallel jobs being executed overnight, and all other jobs being suspended. This heavy-handed restriction on activity is unwanted in any environment and practically impossible to enforce in academia. To make parallel CFD more attractive on 'open' networks of workstations, the ideal parallelisation approach should

- include a domain decomposition method for a heterogeneous processor set
- be integrated seamlessly with existing sequential batch queueing
- take account of varying network load

	Type 1	Type 2	Type 3	Type 4
No. of machines	3	2	4	7
Processor	R5000	R4400	R4400	R4600
Speed (MHz)	150	150	150	133
Main memory (Mb)	96	160	64	64
Data cache (Kb)	32	16	16	16
Instruction cache (Kb)	32	16	16	16
CPU factor, k	1.9	1.6	1.6	1.2

Table 6.1: *Specifications of the workstation cluster*

[1]this assertion is perhaps slightly too strong. Varying processor power is occasionally accounted for in a cost function approach, but without the method being actually demonstrated on a heterogeneous network, for example in [136]. In addition, a successful, truly heterogeneous domain decomposition has been demonstrated[153]; however the dynamic re-allocation method used to achieve the load balance appears very communication-intensive and may only be suitable for very compute-intensive problems of the type presented.

In this chapter, the integration of a parallel multiblock structured aerodynamic simulation code into an open, heterogeneous workstation cluster environment is examined. The use of clusters of workstations for parallel CFD is of high interest to industry[165]. The expected performance increase is limited but comes essentially free since the workstations have usually already been purchased and installed for either sequential CFD work or other tasks. The workstation cluster used is located in the Department of Aerospace Engineering at the University of Glasgow. The cluster consists of a number of Ethernet-connected Silicon Graphics Indy workstations of four different types, as described in Table 6.1. The cluster is typical of departmental level computing facilities (albeit larger than usual) and the facilities at the disposal of industry, where often the development of the computing resource over time results in an inevitably heterogeneous computing environment[156]. The focus of the work is to consider the needs of small and medium sized organisations who require a parallel capability to scale up their computing resource but may at present be discouraged by the perceived practical difficulty involved. This differs from the majority of parallel CFD research where the principal or sole aim has been to achieve the high parallel efficiencies necessary for potential or actual massively-parallel applications on dedicated machines. Network load management software services are exploited to facilitate the application of the decomposition method, and assimilating parallel tasks into the overall batch scheduling and queueing system for the workstation cluster is considered.

The flow solver used is described in Chapter 2. Overlapping grids are employed with two rows of 'halo' cells associated with each internal block boundary. After each time step the updated solution is copied to these halo cells from the corresponding cells in the adjacent block, such that each block has the necessary information to form the residual vectors and Jacobian matrices for the next time step. If blocks sharing a common boundary reside on different processors, then the copying of data is enabled using message passing.

6.2 Cost function minimisation

We wish to distribute structured data blocks amongst the processors of a parallel machine. The primary consideration in determining an efficient distribution is that each processor should spend the same amount of time performing calculations between the synchronous communication phases i.e. that the processors are not idle. This is the load balancing problem. Restricting our discussion at present to a homogeneous parallel machine (where all the processors are identical), for CFD applications a balanced load can be obtained, to a good approximation, by assigning an equal number of grid cells to each processor. Sub-domain shape, e.g. block aspect ratio, and boundary conditions can also influence processor load[157] but these are usually ignored as less important effects. The load balancing problem can then be modelled as a minimisation problem for the 'cost' H due to the time spent performing calculations[145]:

$$H = \frac{P^2}{N^2} \sum_{q=1}^{P} N_q^2 \tag{6.1}$$

where P is the number of processors, N is the total number of grid cells and N_q is the number of grid cells resident on the processor q. As noted in section 6.1, non-deterministic procedures are used to solve this allocation problem. No clear consensus on which method is best has appeared in the literature, although simulated annealing (S.A.) is most often cited as reliably producing near-optimal results, for example in [137],[145],[157], although there are some reservations about the relatively long execution time of the S.A. algorithm. An iterative improvement (I.I.) technique is said to often out-perform S.A. if tailored towards the particular application[158]. For these reasons I.I. and S.A. will be evaluated as minimisation procedures for the cost function (6.1). Their algorithms are described below.

Iterative Improvement

An algorithm based on iterative improvement[158],[159] is very straightforward to program. Some initial configuration of the state (which can be generated at random if necessary) is required, along with a cost function definition. In an iterative manner, a small change based on a random selection is made to the system and this 'basic

move' is either accepted or rejected. The acceptance criterion is as follows: if the move causes the cost to decrease, the move is accepted, otherwise the move is rejected. The process is terminated when a large pre-determined number of consecutive attempts are unsuccessful. Note that careful selection of the basic move is crucial to the success of the method. The method is sometimes referred to as 'hill-climbing'.

Simulated Annealing

The method of simulated annealing[158],[159],[160] is a relatively new method for the minimization of objective functions. It is particularly suited to discrete, very large configuration spaces i.e. for combinatorial optimization problems. The title of the method is due to an analogy with the slow cooling of metals. The simulated annealing algorithm is straightforward to program, and has as its kernel the iterative improvement algorithm. Again an initial configuration of the state and a cost function definition are required. The acceptance criterion is as follows: if a proposed basic move reduces the cost, then the move is accepted. If the cost is increased, then the move is only rejected with a certain probability, called the Metropolis criterion[161]. Included in this criterion is an artificial system 'temperature' such that at high temperatures almost any basic move is accepted, however costly, and at low temperatures effectively zero 'bad' moves are accepted i.e. the algorithm becomes one of iterative improvement. A high starting temperature is used, and the temperature is periodically forced down by some factor after a large number of basic moves have been proposed. The intention is to explore the entire state-space with the Metropolis criterion providing a means of escape from local minima.

Two structured multiblock grids were considered to evaluate I.I. and S.A. for the allocation problem. Details of grid dimensions are shown in Table 6.2. Note that both grids consist of a large number of blocks with widely varying block sizes. To evaluate the effectiveness of the cost minimisation procedures, an efficiency measure E_1 is defined as

$$E_1 = \frac{N/P}{N_q^{max}} \tag{6.2}$$

	Grid 1	Grid 2
Total number of cells (N)	48,425	43,417
Number of blocks	81	21
Average block size	598	2067
Biggest block size	2349	6642
Smallest block size	104	319

Table 6.2: *Details of multiblock grids used in allocation test problems*

where N_q^{max} is the greatest number of grid cells on any one processor in the final allocation. Note that for an ideal allocation E_1 is unity. Two basic moves were used for both I.I. and S.A.; a 'simple' move where two randomly chosen processors swap randomly chosen blocks, and a 'complex' move where clusters rather than single blocks are swapped. The clusters begin as randomly chosen blocks, then collect blocks on the same processor with a probability of 0.2 for each possible collection[145]. The values of E_1 obtained for each minimisation method and different numbers of processors are shown in Figures 6.1 and 6.2. In each case S.A. out-performs I.I. for the 'simple' move. S.A. provides a mechanism for avoiding local minima which can trap the I.I. procedure. However, there is negligible difference in the final result for the 'complex' move. This basic move is designed to enable larger jumps in the state-space of the type required to avoid local minima (for this problem), and has had the desired effect. Note that the complex S.A. has also out-performed the simple S.A. method. A very good discussion of the importance of choosing an appropriate basic move is included in [145]. Note that for both test problems, the efficiency of the final allocation begins to decrease when an allocation over a large number of processors is attempted. This occurs when the number of cells in the largest block becomes larger than the ideal number of cells per processor N/P. If it were desired to use a large number of processors, this problem could be avoided by manually splitting the biggest grid block.

For the remainder of this work the complex I.I. minimisation procedure will be used. More detailed cost functions will be employed, but the nature of the minimisation problem will remain the same. It is preferred to the complex S.A. procedure since it requires less execution time, less than one second compared to

about four seconds, and provides equally high quality results.

6.3 Communication cost

The majority of parallel applications of structured multiblock codes which have appeared in the literature consider only the criterion of load balancing to achieve good parallel performance. A good example is [146] where impressive results are demonstrated on a number of parallel machines, including a dedicated workstation cluster. However, a number of researchers have also stressed the need to take into account communication overhead. The simplest way to take into account the cost of communication as well as computation is to introduce a communication cost element into the cost function, and use a balance coefficient μ to scale the relative importance of the cost elements. The cost function for the allocation problem then becomes

$$H = \frac{P^2}{N^2} \sum_{q=1}^{P} N_q^2 + \mu \left(\frac{P}{N} \right)^{\frac{1}{2}} \sum_{e \leftrightarrow f} 1 - \delta_{q(e),q(f)} \tag{6.3}$$

The first term in this equation is the load balancing term of equation (6.1). The second term is the communication overhead term. For all the cells e on block edges which must communicate with cells f in other blocks, a cost is incurred if e and f do not reside on the same processor q. The choice of scaling constants for each cost element is designed to keep their relative magnitudes constant regardless of the problem size, as discussed in [145]. For codes with a great deal of calculation compared to communication μ should be small, and vice-versa. This explains why communication cost may be disregarded for some flow solvers. According to De Keyser and Roose[136], it is only important to determine approximately the relative magnitude of computational and communication cost, rather than a precise value for μ. Hence we seek a value for μ where the resulting final allocation may differ from that obtained with $\mu = 0$, indicating that 'physical' adjacency of blocks is being taken into account to a degree, but where the load balancing problem is not being overwhelmed, i.e. E_1 does not become too small. Trial allocations with varying values of balance coefficient μ and numbers of processors P for Grids 1 and 2 indicated the range $10^{-7} < \mu < 10^{-2}$. To be more certain of obtaining an appropriate value for μ, trial runs of 50 implicit time steps using Grid 1 on two

processors of Type 3 were performed for various μ. The results are shown in Figure 6.3. The timings shown are averages of ten runs performed overnight when the workstation cluster was very lightly loaded. The parallel efficiency E_p shown is defined as

$$E_p = \frac{sequential\ execution\ time}{(parallel\ execution\ time) * (no.\ of\ processors\ used,\ P)} \tag{6.4}$$

The single processor run was perfomed on a processor of Type 2 which has the same speed as Type 3 processors but enough memory for a sequential execution. From the figure the communication model has had a small effect on execution times. The shortest execution times were obtained for $10^{-5} < \mu < 10^{-2}$; note that for these cases the allocations found by the minimisation procedure were identical. For high values of μ the communication cost begins to dominate, to the detriment of the load balance. For $\mu = 1.0$ all of the blocks were allocated to one processor. The maximum parallel efficiency achieved was 82%. This indicates that communication costs for the flow solver on the workstation cluster are high. That a greater parallel efficiency was not achieved is not an indication of a failure in the cost function allocation method; regardless of which allocation is determined, communication must always occur between processors. To achieve a higher parallel efficiency without resorting to changing the flow solver algorithm, the way in which the message passing is programmed could be examined for possible improvement, or the communication network upgraded. Far greater parallel efficiency has been obtained for the same code on a dedicated parallel machine[147]. However, in the present work the objective is to achieve a scaling-up of computing power, accepting that performance gains are limited. In this context the parallel efficiencies obtained are acceptable. For subsequent results presented in this chapter, a value of $\mu = 10^{-3}$ will be used where a communication cost model is employed. The same problem was also calculated using 3 to 6 processors (of Types 2 and 3), with and without the communication overhead term in the cost function. The averaged execution times, parallel efficiencies and parallel speedups are shown in Figures 6.4, 6.5 and 6.6 respectively. The parallel speedup S is defined simply as

$$S = E_p P \tag{6.5}$$

Note that for all cases the inclusion of the communication cost element has led to improved parallel performance.

When communication overhead is taken into account, the most popular approach is to approximately account for the relative importance of computational and communication costs, as described above for the allocation problem, and also for the mesh partitioning problem. For particular applications a direct mapping of the computational domain onto processors can be visualised and exploited, as discussed in [136]. A good example of this is included in [162] where a large single block problem is decomposed into a two-dimensional array of rectangular patches to exploit the processor connectivity of a massively parallel machine where the processors are arranged in a two-dimensional array. However, this type of approach lacks generality, few computational domains decompose easily to topologies which match the target machine topology. The obvious next step in developing a communication cost model is to explicitly predict or measure the communication time, rather than approximately accounting for it. However, communication time is a function of message size, message-passing method, processor type, processor loading, network type and network loading which means creating a predictive model is prohibitively complex[136],[163]. Some researchers have attempted to measure inter-processor communication costs during run-time[163],[164] which removes some of the difficulties but the implementation of such an approach is still an order of magnitude more difficult than using the simpler method employed here, and a commensurate improvement in performance has not been demonstrated.

6.4 Heterogeneous load balancing

The computational cost is a function of the processor speed as well as the number of cells allocated to the processor. As discussed in Section 6.1, research in parallel CFD has almost exclusively concentrated on homogeneous parallel computers consisting of identical processors. However, if the parallel computer consists of a non-dedicated workstation cluster, for example that considered in this work (see Table 6.1), then the varying processor speeds of the heterogeneous computer must be included in the cost function to efficiently use the resource. Extending equation (6.3) to include

different processor speeds gives the new cost function

$$H = \frac{P^2}{N^2}\sum_{q=1}^{P}\left(\frac{N_q}{k_q}\right)^2 + \mu\left(\frac{P}{N}\right)^{\frac{1}{2}}\sum_{e\leftrightarrow f} 1-\delta_{q(e),q(f)} \qquad (6.6)$$

where k_q is a coefficent which varies directly with the processing speed of processor q. Hence if an allocation is attempted onto two processors, the first with twice the speed of the second (i.e. $k_1/k_2 = 2$), then to minimise the computational cost two thirds of the cells would be allocated to the first processor and one third to the second. The most reliable way to determine values for k is to compare execution times on each of the processors for a standard sequential problem[156]. Vendor information concerning processing speed is unreliable for this purpose, especially when processors from more than one vendor are used. On the workstation cluster considered the commercial management software LSF is used for job control and batch scheduling. Use of such management software enables efficient use of distributed computing networks[166] and is becoming widespread in industry. LSF also provides numerous functions for interrogation of processor configuration and loading that can be simply included in user programs. An LSF function for ascertaining directly the coefficents k_q (termed 'CPU factors' in LSF) was employed in the static allocation method. Values for k from the workstation cluster used are included in Table 6.1. The computational cost model could be further refined. The processor speed is influenced by the proportion of accessed memory which resides in the memory cache rather than the main memory[165], although most researchers ignore or disregard this effect as insignificant.

In order to examine the effectiveness of the new cost function (6.6) in exploiting a heterogeneous processor set, the trial problem of 50 time steps using Grid 1 was repeated using various heterogeneous workstation sets for the parallel machine. Ideally the execution time will vary inversely with the sum k_{total} of the CPU factors k of the processors used. The execution times are plotted against $1/k_{total}$ in Figure 6.7. The serial execution time is included in the figure, and is joined by a straight line through the origin to indicate optimal performance. The parallel timings are presented in three groupings, results for 2, 4 and 6 processors. For each grouping, the result with the largest $1/k_{total}$ is the result for execution on a homogeneous set of Type 4 processors (the slowest available grouping). For each group, if the

results formed a straight line passing through this end point, with gradient equal to the optimal gradient, then the usage of available processing power would be as efficient as the homogeneous case. We could expect the gradient to be slightly less than optimal since the communication time remains approximately constant for increasing processor speed. The results demonstrate a general trend of decreasing execution time with increasing processor power as required, except at a few points where the execution time has increased with increasing processor power. This is disappointing since a heterogeneous execution should always be at least as quick as a homogeneous execution using processors of the slowest type in the heterogeneous set. However it is perhaps unrealistic to expect a non-deterministic allocation method to always produce a near-optimal result; none of the timings are unacceptable and the general trend clearly indicates that heterogeneity is being reasonably accounted for. Examining some particular results helps to indicate the worth of employing the heterogeneous allocation model. Using two processors of Type 3 ($k_{total} = 3.2$, $1/k_{total} = 0.3125$) the execution time is 838 seconds. Replacing one of these processors with the faster Type 1 processor (now $k_{total} = 3.5$, $1/k_{total} = 0.2857$) would not result in a faster execution if a homogeneous allocation method were employed, the faster processor having to wait while the slower computes its half of the load. With the heterogeneous allocation model the execution time was 761 seconds, a reduction of 9.2% for a 9.4% increase in computing power.

Note that in Figure 6.7 results are again presented for allocations determined both with and without communication cost modelling. In every case the executions were faster when the communication cost element was included.

6.5 Dynamic load balancing

In Section 6.1 it was described how the available parallel computing resource often takes the form of a non-dedicated heterogeneous network of workstations. Most parallel CFD work is performed at present using dedicated, single-user parallel computers. The presence of other users' tasks causes a serious problem for parallel applications. Even if the subset of processors to be used for the parallel task is carefully selected before run-time, either manually or using management software, the

load during run-time on these processors can vary dramatically and unpredictably. A load histogram for the workstation cluster considered in this work is included in [166]. One sequential task running interactively on a processor already being used for a parallel task can double the execution time of the parallel task. Inexperienced use of a workstation can lead to disk space/main memory 'swapping' which can easily reduce the effective processing speed by an order of magnitude and have an even worse effect on the parallel task. Even seemingly benign activities such as using an internet browser or a graphical electronic mail tool can have a significant effect. This *dynamic load balancing* problem must be tackled if the objective of reliable, robust parallel execution is to be achieved.

The recorded execution times for twenty trial runs of the test problem described in Section 6.3 using four processors of Type 4 are shown in Figure 6.8 for various network conditions. The timings denoted 'quiet' are the results of overnight runs when the cluster was lightly loaded. The variation in execution time from the fastest possible is small. The timings denoted 'busy' are the results of day-time runs when the workstation cluster was moderately to heavily loaded. The timings are far less predictable, some taking 30% longer than the fastest possible. From experience, some of these timings could have been even greater. The longest execution times shown are probably due to interactive use of internet browsers and mail tools on one or more processors. Other common workstation cluster activities which would have a greater impact are the use of graphical grid generation and solution visualisation software. A 'worst case' timing is also included in the figure. After initialising the parallel task, an interactive serial task was deliberately started on one of the processors. This has increased the execution time by approximately 55%. The present dynamic load balancing problem is then to bring the 'busy' and 'worst case' timings down to the 'quiet' level. The averaged parallel efficiencies are 64%, 57% and 42% for the 'quiet', 'busy' and 'worst case' situations respectively.

Chien et al.[152],[155] present an advanced dynamic load balancing method. The effective speed of each processor is continually monitored by measuring and comparing the waiting time for the communication phase to complete on each processor, adjusting coefficients in the cost function if necessary, and re-allocating the mesh partitions if necessary. The method is very efficient for re-allocating a dynamically

adapted grid, and enables eventual complete migration of the parallel task from a heavily loaded processor if necessary. The approach of Chien et al. produces impressive results but at the expense of considerable complexity and programming effort. Furthermore, the group have themselves asserted[2] that the dynamic load balancing problem has been over-elaborated in recent years, with very complex methods being developed to achieve increasingly small performance gains, and that the only real problem in dynamic load balancing on open workstation clusters can be presented simply as

- recognise when processor A is being heavily used by another task
- identify a lightly loaded processor B
- migrate the work of processor A onto processor B
- do all of this as quickly and simply as possible

These are also the objectives of the present work. From Figure 6.8, some interference of the parallel task can be tolerated (where the 'busy' timings are only slightly longer than the 'quiet'), any performance gains in sending a subset of the blocks on the 'busy' processor to other processors are likely to be small and would not justify the programming effort. The only real problem arises when a processor becomes heavily loaded, and the entire load from that processor should be migrated. Note that this also protects the interests of the interactive user, who then becomes the sole user of the processor. A dynamic re-allocation method was implemented as follows, using native LSF and PVM functions called from within the flow solver code for simplicitly rather than creating custom software:

- periodically monitor processor loadings (LSF)
- if a processor is too heavily loaded, find a candidate alternative (LSF)
- initiate a new task on the new processor (PVM), pass all the necessary information including the solution and the grid to the new task (PVM)
- stop the old task and proceed with the calculation

[2]during their ECCOMAS conference presentation[152]

Note that the frequency of load monitoring and the threshold for deciding whether a processor is overloaded are decided before run-time by the user. The major part of the information passed to the new process consists of the solution and grid for the partition allocated to that processor. For the present test problem this is approximately 500Kb for each migration, which is a manageable figure for an Ethernet network. It is not necessary to pass the Jacobian matrices (for the implicit scheme) which form the major part of the total memory usage for the flow solver. The 20 trial runs in the 'worst case' scenario were repeated, but this time using the dynamic re-allocation method. The results are denoted 'dynamic' in Figure 6.8. The load monitoring frequency was set at every 10 time steps, recalling that 50 time steps are executed in total. It was detected that one processor was over-loaded at the first call of the load monitoring function (i.e. after 10 time steps) and the load from that processor was transferred to a lightly loaded processor. There is therefore a clear improvement over the 'worst case' execution time. An average parallel efficiency of 57% was achieved in the 'dynamic' case, as opposed to 42% for the 'worst case'. The 'dynamic' execution times could be further reduced by increasing the load monitoring frequency. Note that this model parallel CFD task has a lower associated parallel efficiency than would be the case for a real problem. It is unlikely that an engineer would use four processors for a problem which comfortably executes on two of the same processors, as in this case. It is well known that larger problems have greater parallel efficiencies (since the communication cost to computational cost ratio decreases), so since the dynamic re-allocation method effectively reduces the computational cost the performance gains for real problems would be larger. In addition, dynamic re-allocation would be of greater use for typical CFD jobs with longer execution times than the ten minutes in the current test problem.

In the event of no suitable alternative processor being available, the present method proceeds with the calculation on the same processor. This could be improved by first attempting to contract the problem onto one less processor, or if this is not possible by automatically re-submitting the parallel task to the batch queue, re-starting from the latest checkpoint files. The present method includes periodic checkpointing to local and main disks to enable re-starting in the event of a network failure.

Most organisations with a distributed computing network employ a batch scheduling and queueing system, either developed in-house or proprietary, to enable transparent load management and achieve high productivity. Users are becoming accustomed to the convenience of high performance environments, where the system does the work of prioritizing batch jobs and selecting resources, and the user must only submit the (sequential) job and can depend on the timely arrival of the results. Ideally executing parallel tasks should be as simple and reliable. The dynamic reallocation method presented here coupled with management software such as LSF which fully supports sequential and parallel applications alike makes this possible.

6.6 Discussion

A domain decomposition method for a parallel, structured multiblock flow solver has been presented. The method is suitable for use on a non-dedicated parallel computer consisting of a heterogeneous workstation cluster. It has been noted that the majority of work concerning parallel CFD considers dedicated, homogeneous parallel computers. The additional difficulties encountered in a non-dedicated heterogeneous environment have been discussed. The parallel computing resource available to many engineers in small and medium-sized enterprises is of this type, although widespread use of parallel CFD to achieve a scaling-up in computational resource appears to be hindered by the perceived complexity involved. With this in mind, the domain decomposition strategy presented here attempts to deliver an effective resource in as straightforward a manner as possible.

The method employs a cost function minimisation approach. It is assumed that the multiblock grid consists of enough small blocks to enable a reasonably balanced distribution. The cost function consists of computational and communication cost elements. The time required for a processor to compute its share of the load is assumed to vary directly with the number of grid cells assigned to that processor. The time required for inter-processor communication is assumed to vary directly with the number of cells on the block boundaries which must communicate with blocks which reside on different processors. The relative importance of the cost elements is defined by a coefficient, a value for which is determined from timing

experiments. The various processor speeds are ascertained using the management software LSF and are accounted for in the cost function. LSF is also used to monitor interference of other users' tasks with parallel execution, and to select a lightly loaded processor as a target for migration. The method enables effective parallel execution in the demanding environment of an open heterogeneous workstation cluster. Implementation is straightforward, facilitated by modern management software and message-passing libraries, and does not require a specialist programming or information technology effort.

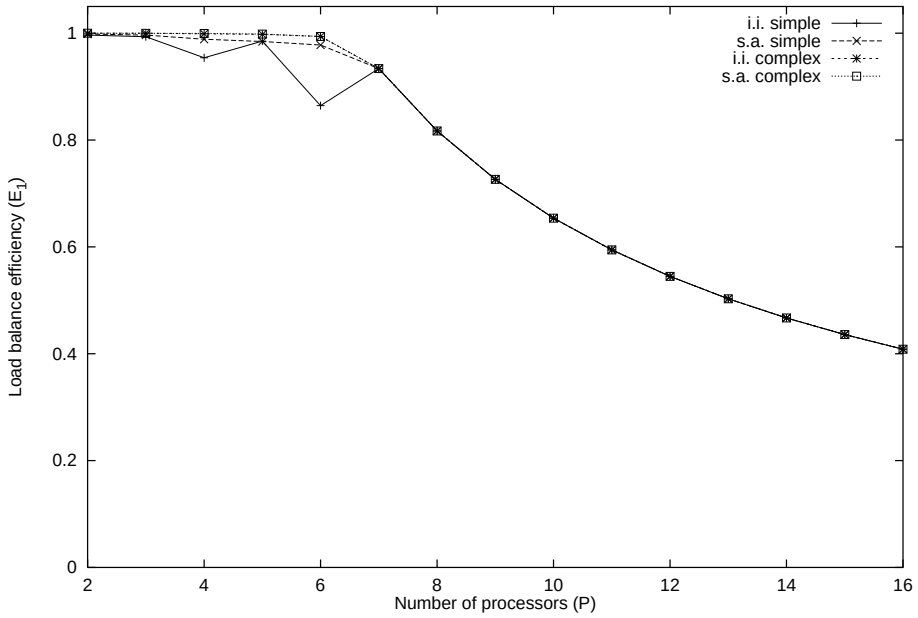


Figure 6.1: *Cost function minimisation test, Grid 1*

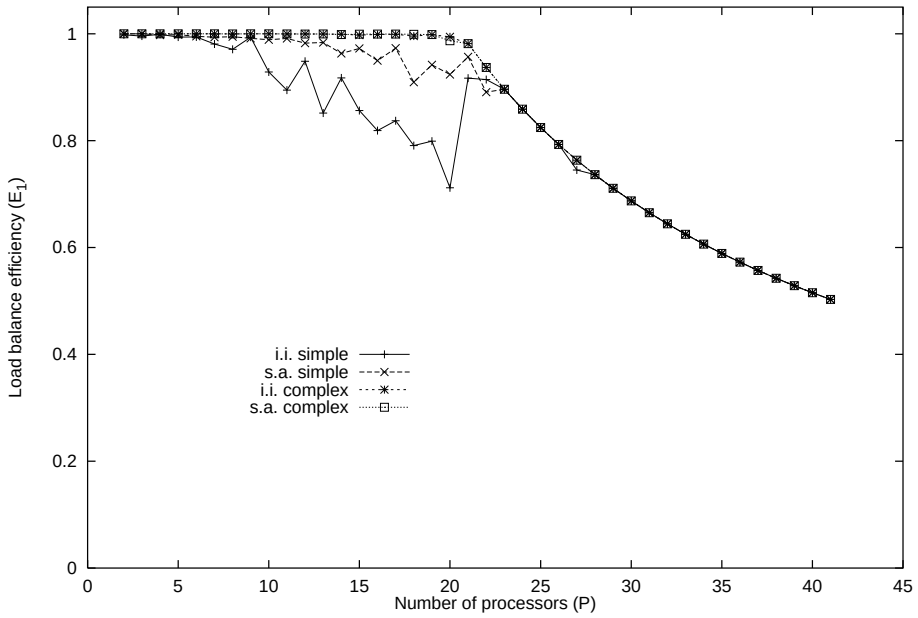


Figure 6.2: *Cost function minimisation test, Grid 2*

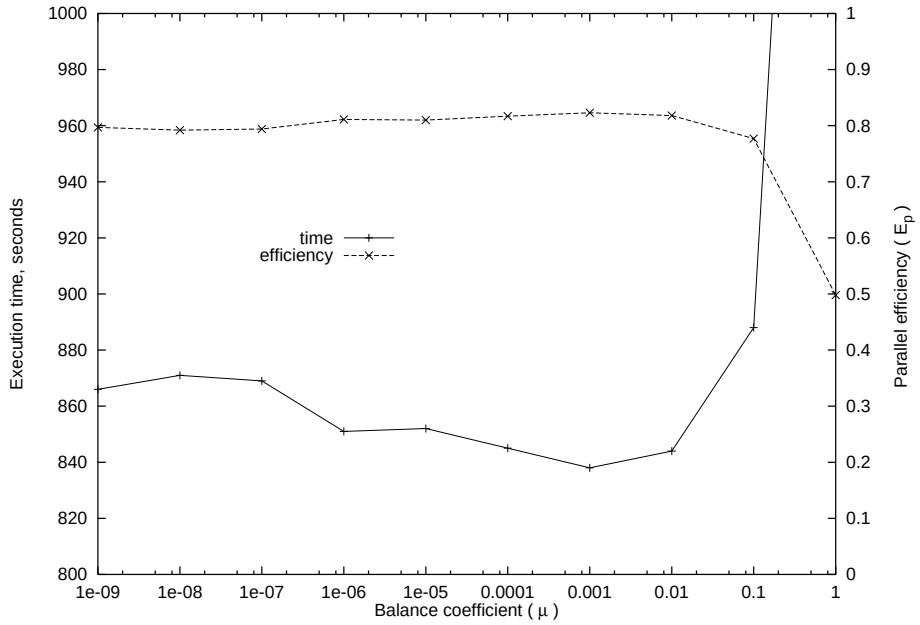


Figure 6.3: *Influence of balance coefficient* μ

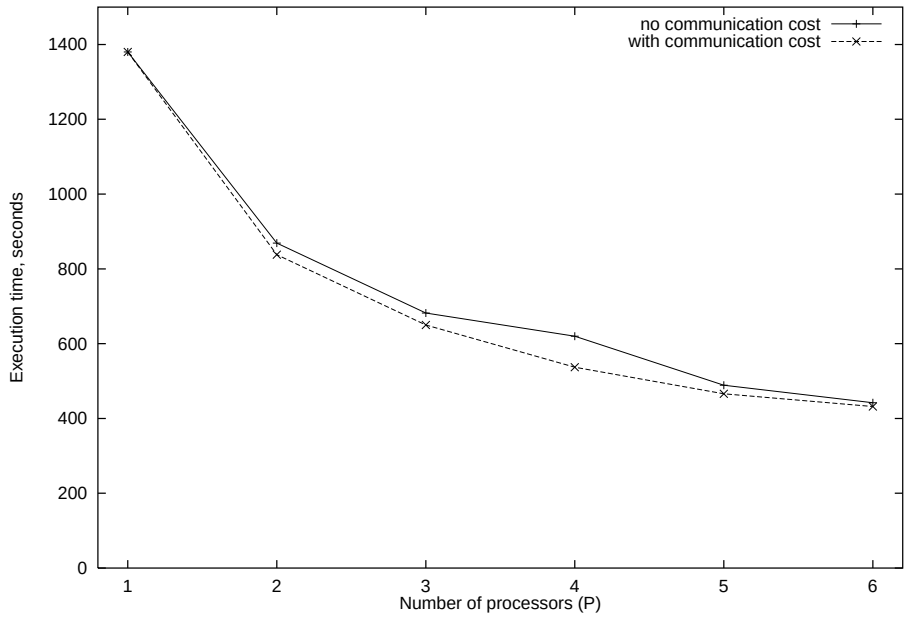


Figure 6.4: *Execution times, homogeneous network*

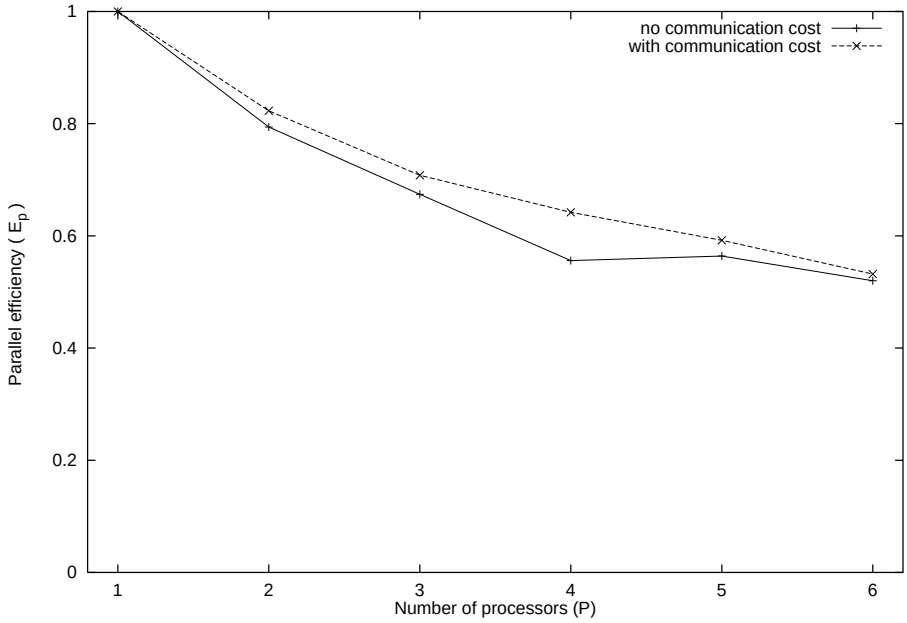


Figure 6.5: *Parallel efficiencies, homogeneous network*

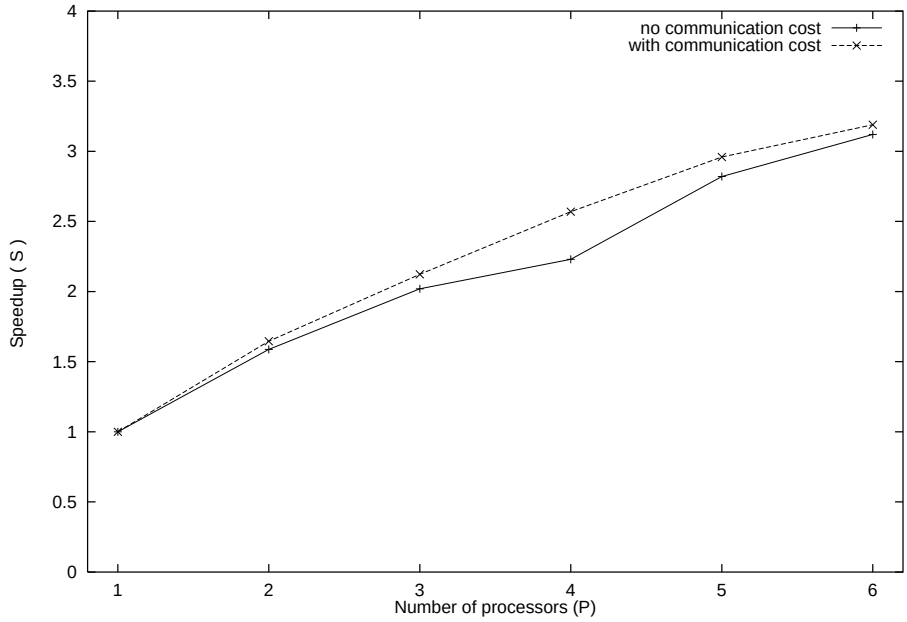


Figure 6.6: *Parallel speedup, homogeneous network*

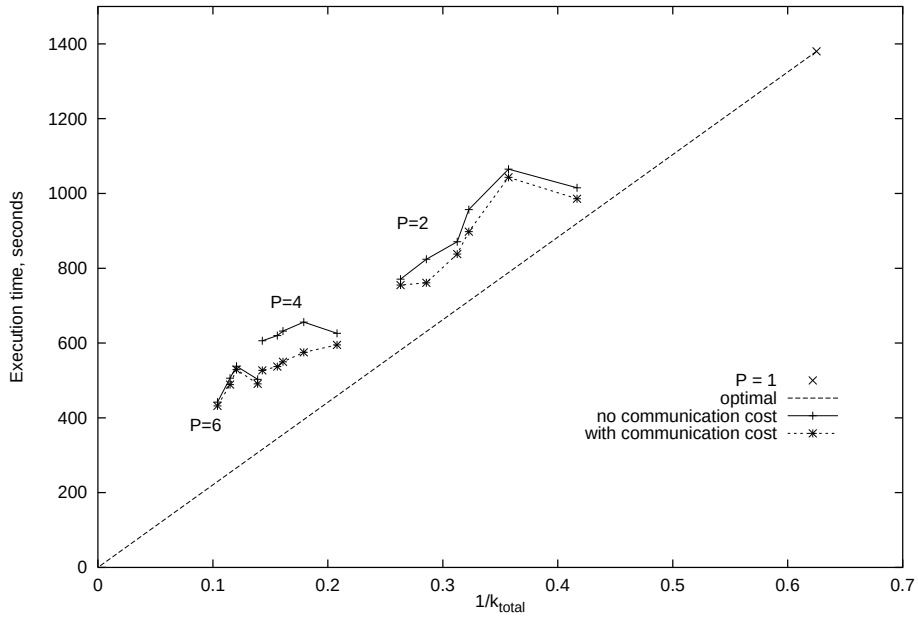


Figure 6.7: *Execution times, heterogeneous network*

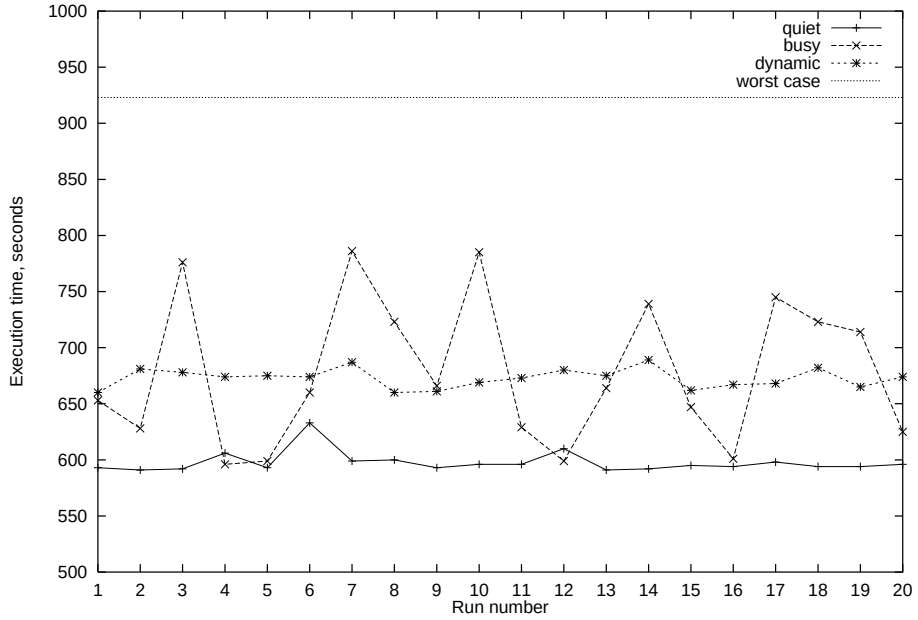


Figure 6.8: *Dynamic re-allocation performance*

Chapter 7

Conclusion

The general aim of this work has been to promote computational aerodynamics as a useable technology in engineering design and scientific enquiry. Several aspects of applied CFD have been considered, each reflecting current problems, concerns and requirements of CFD users.

In Part I of this work the implementation of an axisymmetric flow solver was described, and its use as a tool for engineering design and scientific enquiry was examined. First, in Chapter 2 the adaption of an existing two-dimensional flow solver for axisymmetric flow is described. The main points of the chapter are as follows:

- the axisymmetric Euler and Navier-Stokes equations can be cast in a form very similar to the two-dimensional equations
- for axisymmetric problems, the potential performance gains of an axisymmetric flow solver over a fully three-dimensional method are considerable
- the inclusion of the axisymmetric source terms for the Navier-Stokes and $k-\omega$ turbulence model equations in the numerical scheme is described
- an exact, laminar test case is considered, for which very good agreement with theory is demonstrated.

The axisymmetric flow solver was applied to high speed forebody and base flow cases in Chapter 3. The motivation was to assess the capability of the method as

an engineering tool; as well as evaluating the accuracy of the results, pre- and post-processing effort, robustness and required computing power were also considered. These issues can be grouped together as 'cost', and as such are important to engineers in an industrial/commercial environment. For the high speed forebody applications, the results of the study can be summarised as:

- good agreement was obtained with experimental data and other calculations
- the flow solver is robust and fast, the run-times being measured in minutes
- an implicit treatment of the inviscid part of the axisymmetric source terms improves the performance of the method
- the very significant performance advantage over a three-dimensional method was demonstrated
- the computing resource required is modest
- pre- and post-processing are straightforward.

The method therefore clearly satisfies the demanding requirements of an engineer operating in an industrial/commercial environment. The design of forebody geometries is not straightforward, involving compromise between several aerodynamic effects which can be counter to intuition[62]. The present method enables solutions to be obtained in minutes using modest computing power. This affords the possibility of employing automatic design techniques for forebody geometries which appear at present to have been limited to inviscid aerofoil and wing calculations[17],[167]. This is a potentially fruitful avenue for future work.

Application of the method to base flow problems proved more problematic. The results of the study can be summarised as follows:

- fairly poor agreement with experimental data was obtained, similarly to other published calculations for the problem considered
- the method is not robust for this type of application. The numerical instability is associated with the implementation of the $k - \omega$ turbulence model, and originates in the vicinity of the free stagnation point

- grid convergence is an important and demanding issue for this type of application
- a method for including upstream boundary condition information in the simulation has been described
- the computing resource required is modest
- pre- and post-processing are straightforward.

In general, for turbulent base flow calculations with large scale separation the method performs less well than for the forebody problems. Although it is still possible to obtain solutions relatively quickly, the robustness should be improved before the method can be considered practical for this problem. The deficiencies of a two equation turbulence model, employing the Boussinesq approximation, are well known for highly separated flows, although in the context of available prediction methods the balance between accuracy and complexity is reasonable. If in future work the numerical instability problem could be solved, the method would then become viable as an engineering tool. The form of the turbulent source term Jacobian matrix appears to be a key to this problem. The next step would then be to evaluate the inclusion of more advanced turbulence models, which have the potential to improve the accuracy of the simulation, but may incur a large penalty in computational cost and complexity.

Chapter 4 describes how a complex shock interaction phenomenon was successfully examined using a computational aerodynamics method. The main results of the study are:

- the hysteresis in the shock reflection type occurring in the plume of an underexpanded jet has been successfully predicted, showing good agreement with experiment
- nozzle calculations over a range of Reynolds number have helped explain the scatter in the experimental results
- the detailed results from CFD have enabled identification and examination of flow features not initially recognised in the experiments:

  - Mach disc curvature
  - recirculation and continuing compression behind the Mach disc
  - the presence of a small diameter Mach disc in the apparent regular reflection

- two-dimensional shock reflection theory has been reviewed and employed to aid interpretation of the results
- appropriate boundary conditions for this type of problem have been established.

The numerical investigation, used in conjunction with experimental data and shock reflection theory, added significantly to the understanding of the problem. Experimental investigation of shock interaction problems is hampered by practical difficulties, thus the potential for numerical investigations is large. At the 1998 International Mach Reflection Symposium it was noted that more than 90% of the presentations included numerical results, either as the sole analysis method or complementing experiment and/or theory, underlining this point. Although the role of theory in this study has been stressed, a comprehensive theory for the reflection of conical shock waves does not exist at present. In a manner analogous to the two-dimensional shock reflection problem, the present method could be used to study reflection of conical shock waves outwith the context of a complex plume structure. This would strengthen the theoretical framework available for analysis of shock interaction problems, and would facilitate the investigation of more complex axisymmetric problems. The success of the numerical method in the present work, and the recent extension to a fully three-dimensional flow solver, points to another area of future work. Numerical investigation of further shock interaction problems from the same series[107] should now be possible, for which the necessary computing power is now available[147].

Part II of this work concerns the issues of pre-processing and parallel computing, both of which are important to the practicality of routine CFD analyses, and are important sub-topics of CFD in their own right. Chapter 5 describes a new approach to multiblock grid generation. The main conclusions of the study are:

- an unstructured quadrilateral/hexahedral mesh generation method which has the potential for use as an automatic multiblock topology generator has been identified
- a new approach for automatic topology processing has been presented
- a cost functon for multiblock topologies has been constructed, based on experience using interactive grid generation tools
- the approach has worked well for a number of test problems, using simple curve definitions and a straightforward, non-deterministic minimisation method
- there is a clear route to extending the method to three-dimensional problems.

The large amount of human effort required for grid generation is increasingly becoming a frustration of CFD users. As solution methods improve and available computing power continues to increase, the main obstacle to performing large, complex simulations is now frequently grid generation, the technology for which has not progressed at the same rate as other elements of the CFD analysis process. The topology processing method described in this work provides the possibility, in conjunction with an identified unstructured quadrilateral/hexahedral grid generation method and existing structured grid generation technology, for a fully automatic grid generation capability for even complex configurations. The unstructured grid generation method is well established in structural analysis. The potential of the topology processing method has been demonstrated in this work. The capabilities (and limitations) of existing structured grid generation technology are well known. Assembling all of the elements together would undoubtedly be a significant task, but the technology does exist at present and the potential rewards surely justify the effort. Immediate future work in this area should begin by improving the curve definitions and minimisation algorithm. This should not be problematic since methods for both problems are well established. The topology processing method should then be coupled with the initial topology generator and structured grid generation routines to obtain a fully automatic grid generation tool. If this can be achieved without unforeseen difficulties, which seems likely since all the elements of the pro-

cess are well established, then the generalisation to an automatic three-dimensional grid generation tool becomes a real prospect.

Chapter 6 concerns the implementaton of a parallel computing strategy for computational aerodynamic simulation using a non-dedicated, heterogeneous network of workstations. Some care was taken in assessing the current status of parallel CFD, with the following conclusions:

- parallel computing is accepted at present as providing the most cost-effective route to high performance computing, a requirement of the CFD community
- the perceived difficulty in achieving a parallel CFD capability is impeding widespread adoption of the technology
- this is especially true of small and medium sized enterprises, who do not possess dedicated parallel machines and therefore have the additional problems of heterogeneity and dynamic load imbalance
- the vast majority of parallel CFD research has concerned homogeneous, dedicated parallel machines, compounding the point above.

Parallel computing has for some time been seen as the solution to CFD's continual requirement for increasing computing power. The gradual maturing of the associated technology, parallel solution algorithms, message passing libraries and parallel compilers, has encouraged the adoption of parallel computing by the CFD community. This work has addressed many of the issues which are discouraging more widespread exploitation of parallel CFD. It has been demonstrated that it is now possible to implement parallel CFD even in a demanding open workstation cluster environment where heterogeneity and dynamic load imbalance must be considered. The conclusions of the study are as follows:

- load-balanced allocations of grid blocks to processors can be achieved using a cost function approach
- a tailored iterative improvement algorithm is an effective minimisation method, out-performing simulated annealing in terms of execution time

- the importance of communication time to parallel performance is application-specific. For the present parallel flow solver, employing an artificial balance coefficient to scale the relative importance of computation and communication cost is effective
- the heterogeneity of a parallel computer can be accounted for easily in the cost function
- proprietary management software is useful for obtaining processor information for static load balancing and dynamic load information
- dynamic load balancing is important for maintaining acceptable productivity when using a non-dedicated parallel machine
- a dynamic re-allocation method has been described.

It is likely that parallel CFD will become commonplace if the perception that it is a difficult technology can be overcome, especially with continued improvements in parallel computing technology, notably resource management software.

Appendix A

The two-dimensional (planar) Navier-Stokes equations

A.1 Introduction

In this appendix the two-dimensional Navier-Stokes equations are presented in various forms for the sake of completeness and ease of reference to the axisymmetric equations presented in the main body of the report.

A.2 Non-dimensional form

The derivation of the Navier-Stokes equations is included in most fluid dynamics texts, for example [53]. In a two-dimensional cartesian frame they can be written as

$$\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial (\mathbf{F}^i - \mathbf{F}^v)}{\partial x} + \frac{\partial (\mathbf{G}^i - \mathbf{G}^v)}{\partial y} = 0 \tag{A.1}$$

The vector $\mathbf{W}$ is the vector of conserved variables:

$$\mathbf{W} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix} \tag{A.2}$$

where ρ is the density, $\mathbf{V} = (u, v)$ is the Cartesian velocity vector and E is the total energy per unit mass. The flux vectors $\mathbf{F}$ and $\mathbf{G}$ consist of inviscid $(^i)$ and $(^\nu)$

viscous diffusive parts. These are written in full as :

$$\mathbf{F}^i = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{pmatrix}$$
$$\mathbf{G}^i = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vH \end{pmatrix} \tag{A.3}$$

$$\mathbf{F}^\nu = \frac{1}{Re} \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} + q_x \end{pmatrix}$$
$$\mathbf{G}^\nu = \frac{1}{Re} \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} + q_y \end{pmatrix} \tag{A.4}$$

The stress tensor and of the heat flux vector components are written as:

$$\begin{aligned} \tau_{xx} &= -\mu \left(2\frac{\partial u}{\partial x} - \frac{2}{3} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \right) \\ \tau_{yy} &= -\mu \left(2\frac{\partial v}{\partial y} - \frac{2}{3} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \right) \\ \tau_{xy} &= -\mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ q_x &= -\frac{1}{(\gamma - 1) M_\infty^2} \frac{\mu}{Pr} \frac{\partial T}{\partial x} \\ q_y &= -\frac{1}{(\gamma - 1) M_\infty^2} \frac{\mu}{Pr} \frac{\partial T}{\partial y} \end{aligned} \tag{A.5}$$

Here γ is the specific heat ratio, Pr is the laminar Prandtl number, T is the static temperature and M_∞ and Re are the freestream Mach number and Reynolds num-

ber, respectively. The various flow quantities are related to each other by the perfect gas relations:

$$\begin{aligned} H &= E + \frac{p}{\rho} \\ E &= e + \frac{1}{2}\left(u^2 + v^2\right) \\ p &= (\gamma - 1)\,\rho e \\ \frac{p}{\rho} &= \frac{T}{\gamma M_\infty^2} \end{aligned} \tag{A.6}$$

Finally, the laminar viscosity μ is evaluated using Sutherland's law:

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0}\right)^{3/2} \frac{T_0 + 110}{T + 110} \tag{A.7}$$

where μ_0 is a reference viscosity at a reference temperature T_0. These can be taken as $\mu_0 = 1.7894\text{x}10^{-5}$ kg/(m.s) with $T_0 = 288.16$ K. It is stressed that the quantities presented here have been non-dimensionalised. The procedure used is as follows:

$$x = \frac{x^*}{L^*}, \qquad y = \frac{y^*}{L^*}, \qquad t = \frac{t^*}{L^*/V_\infty^*},$$

$$u = \frac{u^*}{V_\infty^*}, \qquad v = \frac{v^*}{V_\infty^*}, \qquad \mu = \frac{\mu^*}{\mu_\infty^*},$$

$$\rho = \frac{\rho^*}{\rho_\infty^*}, \qquad p = \frac{p^*}{\rho_\infty^* V_\infty^{*2}}, \qquad T = \frac{T^*}{T_\infty^*}, \qquad e = \frac{e^*}{V_\infty^{*2}} \tag{A.8}$$

A.3 Reynolds-averaged form

The Reynolds-averaged form of the Navier-Stokes equations permits turbulent flow to be considered. The development is not presented here. It is merely noted that fundamental to this approach is the consideration of the flow variables as consisting of two components, a time averaged component and a turbulent fluctuation component. For example, density and velocity components are decomposed as

$$u = \bar{u} + u', \qquad v = \bar{v} + v', \qquad \rho = \bar{\rho} + \rho'$$

The quantities k (the turbulent kinetic energy), μ_T (the turbulent viscosity) and Pr_T (the turbulent Prandtl number) are introduced via the important Boussinesq assumption in an attempt to model the fluctuating-variable stress terms arising from the Reynolds averaging. For a complete discussion of this subject see [53]. The Reynolds-averaged form of the Navier-Stokes equations are identical to those presented in appendix A.2, except for the stress tensor and heat flux vector components shown below. The variables should be considered as mean flow quantities (superscripts are dropped for clarity). The turbulent nature of the flow is modelled via μ_T and k and a closure hypothesis or turbulence model, for example the $k-\omega$ model, appendix B.

$$\begin{aligned}
\tau_{xx} &= -(\mu+\mu_T)\left(2\frac{\partial u}{\partial x}-\frac{2}{3}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\right)+\frac{2}{3}\rho k \\
\tau_{yy} &= -(\mu+\mu_T)\left(2\frac{\partial v}{\partial y}-\frac{2}{3}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)\right)+\frac{2}{3}\rho k \\
\tau_{xy} &= -(\mu+\mu_T)\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)
\end{aligned} \tag{A.9}$$

$$\begin{aligned}
q_x &= -\frac{1}{(\gamma-1)M_\infty^2}\left(\frac{\mu}{Pr}+\frac{\mu_T}{Pr_T}\right)\frac{\partial T}{\partial x} \\
q_y &= -\frac{1}{(\gamma-1)M_\infty^2}\left(\frac{\mu}{Pr}+\frac{\mu_T}{Pr_T}\right)\frac{\partial T}{\partial y}
\end{aligned} \tag{A.10}$$

A.4 Curvilinear form

The governing equations are written in curvilinear (ξ,η) form to facilitate use on curvilinear grids of arbitrary local orientation and density. A space transformation from the Cartesian coordinate system to the local coordinate system must then be introduced:

$$\begin{aligned}
\xi &= \xi(x,y) \\
\eta &= \eta(x,y) \\
t &= t
\end{aligned}$$

The Jacobian determinant of the transformation is given by

$$J = \frac{\partial(\xi,\eta)}{\partial(x,y)}$$

The equations A.1 can then be written as

$$\frac{\partial \hat{\mathbf{W}}}{\partial t} + \frac{\partial(\hat{\mathbf{F}}^i - \hat{\mathbf{F}}^v)}{\partial \xi} + \frac{\partial(\hat{\mathbf{G}}^i - \hat{\mathbf{G}}^v)}{\partial \eta} = 0 \tag{A.11}$$

where

$$\begin{aligned}
\hat{\mathbf{W}} &= \frac{\mathbf{W}}{J} \\
\hat{\mathbf{F}}^i &= \frac{1}{J}\left(\xi_x \mathbf{F}^i + \xi_y \mathbf{G}^i\right) \\
\hat{\mathbf{G}}^i &= \frac{1}{J}\left(\eta_x \mathbf{F}^i + \eta_y \mathbf{G}^i\right) \\
\hat{\mathbf{F}}^v &= \frac{1}{J}\left(\xi_x \mathbf{F}^v + \xi_y \mathbf{G}^v\right) \\
\hat{\mathbf{G}}^v &= \frac{1}{J}\left(\eta_x \mathbf{F}^v + \eta_y \mathbf{G}^v\right)
\end{aligned} \tag{A.12}$$

The expressions for the inviscid fluxes can be simplified somewhat by defining

$$\begin{aligned}
U &= \xi_x u + \xi_y v \\
V &= \eta_x u + \eta_y v
\end{aligned} \tag{A.13}$$

The inviscid fluxes can then be written as

$$\begin{aligned}
\hat{\mathbf{F}}^i &= \begin{pmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho U H \end{pmatrix} \\
\hat{\mathbf{G}}^i &= \begin{pmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho V H \end{pmatrix}
\end{aligned} \tag{A.14}$$

The derivative terms found in the viscous fluxes are evaluated using the chain rule, for example

$$\frac{\partial u}{\partial x} = \xi_x \frac{\partial u}{\partial \xi} + \eta_x \frac{\partial u}{\partial \eta}$$

The evaluation of the metrics of the transformation is clearly important, and is described in full in [53].

Appendix B

The two-equation $k-\omega$ turbulence model

B.1 Non-dimensional form

The $k-\omega$ turbulence model of Wilcox[54] in non-dimensional form can be written as follows:

Eddy Viscosity

$$\mu_T = \rho k/\omega$$

Turbulence Kinetic Energy

$$\rho\frac{\partial k}{\partial t}+\rho\mathbf{V}.\nabla k-\frac{1}{Re}\nabla.\left[\left(\mu+\sigma^*\mu_T\right)\nabla k\right] = \mu_T P-\frac{2}{3}\rho kS-\beta^*\rho k\omega$$

Specific Dissipation Rate

$$\rho\frac{\partial \omega}{\partial t}+\rho\mathbf{V}.\nabla \omega-\frac{1}{Re}\nabla.\left[\left(\mu+\sigma\mu_T\right)\nabla \omega\right] = \alpha\frac{\omega}{k}\left[\mu_T P-\frac{2}{3}\rho kS\right]-\beta\rho\omega^2$$

Closure Coefficients

$$\alpha=5/9, \qquad \beta=3/40, \qquad \beta^*=9/100, \qquad \sigma=1/2, \qquad \sigma^*=1/2 \quad \text{(B.1)}$$

In the above relations,

$$P = \left[\left(\nabla\mathbf{V}+\nabla\mathbf{V}^T\right):\nabla\mathbf{V}-\frac{2}{3}\left(\nabla.\mathbf{V}\right)^2\right]$$
$$S = \nabla.\mathbf{V}$$

The equations as shown above use the same non-dimensional quantities as in section section A.2, with the addition of

$$k = \frac{k^* Re}{V_\infty^{*2}}, \qquad \omega = \frac{\omega^* L^*}{V_\infty^*}, \qquad \mu_T = \frac{\mu_T^*}{\mu_\infty^*}$$

B.2 Curvilinear form

The equations for k and ω can be written in a curvilinear form analogous to that used for the mean flow equations in section A.4 . Written in full, the two-dimensional Cartesian form of equations B.1 and B.1 become

$$\frac{\partial \hat{\mathbf{W}}_T}{\partial t} + \frac{\partial (\hat{\mathbf{F}}_T^i - \hat{\mathbf{F}}_T^v)}{\partial \xi} + \frac{\partial (\hat{\mathbf{G}}_T^i - \hat{\mathbf{G}}_T^v)}{\partial \eta} = \frac{\hat{\mathbf{S}}_T}{J} \tag{B.2}$$

where the vectors of conserved variables, convective and diffusive fluxes are respectively

$$\hat{\mathbf{W}}_T = \frac{1}{J}\begin{pmatrix} \rho k \\ \rho \omega \end{pmatrix} \qquad \hat{\mathbf{F}}_T^i = \frac{1}{J}\begin{pmatrix} \rho k U \\ \rho \omega U \end{pmatrix} \qquad \hat{\mathbf{G}}_T^i = \frac{1}{J}\begin{pmatrix} \rho k V \\ \rho \omega V \end{pmatrix}$$

$$\hat{\mathbf{F}}_T^v = \frac{1}{J}\left(\xi_x \hat{\mathbf{M}} + \xi_y \hat{\mathbf{N}}\right) \qquad \hat{\mathbf{G}}_T^v = \frac{1}{J}\left(\eta_x \hat{\mathbf{M}} + \eta_y \hat{\mathbf{N}}\right)$$

where the tensors M and N are equal to

$$\hat{\mathbf{M}} = \frac{1}{Re}\begin{pmatrix} (\mu + \sigma^* \mu_T)(\xi_x k_\xi + \eta_x k_\eta) \\ (\mu + \sigma \mu_T)(\xi_x \omega_\xi + \eta_x \omega_\eta) \end{pmatrix}$$

$$\hat{\mathbf{N}} = \frac{1}{Re}\begin{pmatrix} (\mu + \sigma^* \mu_T)(\xi_y k_\xi + \eta_y k_\eta) \\ (\mu + \sigma \mu_T)(\xi_y \omega_\xi + \eta_y \omega_\eta) \end{pmatrix}$$

Finally, the source term is written as

$$\hat{\mathbf{S}}_T = \begin{pmatrix} \hat{\mathbf{P}}_k - \hat{\mathbf{D}}_k \\ \hat{\mathbf{P}}_\omega - \hat{\mathbf{D}}_\omega \end{pmatrix}$$

with the components

$$\hat{\mathbf{P}}_k = \mu_T \left\{ \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + 2 \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] - \frac{2}{3} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)^2 \right\} - \frac{2}{3} \rho k \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)$$

$$\hat{\mathbf{D}}_k = \beta^* \rho \omega k$$

$$\hat{\mathbf{P}}_\omega = \alpha \frac{\omega}{k} \hat{\mathbf{P}}_k$$

$$\hat{\mathbf{D}}_\omega = \beta \rho \omega^2$$

Again the velocity derivative terms are evaluated in (ξ,η) space via the chain rule, as mentioned in section A.4, but remain unexpanded in the source term components above for brevity.

Bibliography

[1] R.T. Jones. Properties of Low Aspect Ratio Pointed Wings at Speeds Below and Above the Speed of Sound. *NACA Report*, TR-835, 1946.

[2] W.D. Hayes. Linearized Supersonic Flow. *North American Aviation Report*, AL-222, 1947.

[3] T. Theodorsen. Theory of Wing Sections of Arbitrary Shape. *NACA Report*, TR-411, 1931.

[4] Z. Kopal. Tables of Supersonic Flow Around Cones. *M.I.T. Center of Analysis Technical Report*, 1, U.S. Govt. Printing Office, Washington, DC, 1931.

[5] J.L. Hess and A.M.O. Smith. Calculation of Non-Lifting Potential Flow About Arbitrary Three-Dimensional Bodies. *Douglas Aircraft Report*, ES-40622, 1962.

[6] P.E. Rubbert and G.R. Saaris. A General Three-Dimensional Potential Flow Method Apllied to V/STOL Aerodynamics. *Society of Automotive Engineers*, Paper 680304, 1968.

[7] F.A. Woodward. An Improved Method for the Aerodynamic Analysis of Wing-Body-Tail Configurations in Subsonic and Supersonic Flow, Part 1 - Theory and Application. *NASA*, CR-2228, 1973.

[8] E.M. Murman and J.D. Cole. Calculation of Plane Steady Transonic Flows. *AIAA Journal*, 9(1):114–121, 1971.

[9] A. Jameson. Iterative Solution of Transonic Flows over Airfoils and Wings, Including Flows at Mach 1. *Communications on Pure and Applied Mathematics*, 27:283–309, 1974.

[10] A. Jameson. Transonic Potential Flow Calculations in Conservation Form. In *Proc. 2nd AIAA Computational Fluid Dynamics Conference*, pages 148–161, 1975.

[11] A. Eberle. A Finite Volume Method for Calculating Transonic Potential Flow Around Wings from the Minimum Pressure Integral. *NASA*, TM-75324 (Translated from MBB UFE 1407(0)), 1978.

[12] M. Hafez, J.C. South and E.M. Murman. Artificial Compressibility Method for Numerical Solutions of the Transonic Full Potential Equation. *AIAA Journal*, 17:838–844, 1979.

[13] R. Magnus and H. Yoshihara. Inviscid Transonic Flow Over Airfoils. *AIAA Journal*, 8(12):2157–2162, 1970.

[14] R.W. MacCormack. The Effect of Viscosity in Hyper-Velocity Impact Cratering. *AIAA-69-354*, 1969.

[15] R.W. Beam and R.F. Warming. An Implicit Finite Difference Algorithm for Hyperbolic Systems in Conservation Form. *Journal of Computational Physics*, 23:87–110, 1976.

[16] J.L. Steger and R.F. Warming. Flux Vector Splitting of the Inviscid Gas Dynamics Equations with Applications to Finite Difference Methods. *Journal of Computational Physics*, 40:263–293, 1981.

[17] A. Jameson. The Present Status, Challenges and Future Developments in Computational Fluid Dynamics. In *AGARD-CP-578, Progress and Challenges in CFD Methods and Algorithms*, 1996.

[18] C. Hirsch. *Computational Methods for Inviscid and Viscous Flows*, volume 2. Numerical Computation of Internal and External Flows, John Wiley, 1990.

[19] A. Jameson. The Evolution of Computational Methods in Aerodynamics. *Journal of Applied Mechanics*, 50:1052–1070, 1983.

[20] J.P. Boris. New Directions in Computational Fluid Dynamics. *Annual Review of Fluid Mechanics*, 21:345–385, 1989.

[21] D.R. Chapman. A Perspective on Aerospace CFD. *Aerospace America*, January:16–19, 1992.

[22] N. Kroll. Technical Evaluation Report. In *AGARD-CP-578, Progress and Challenges in CFD Methods and Algorithms*, 1996.

[23] R.G. Bradley. Future Directions for Applied Computational Fluid Dynamics. In P.A. Henne, editor, *Applied Computational Aerodynamics, Progress in Astronautics and Aeronautics, Vol. 125*, pages 889–904, 1990.

[24] J.F. Thompson and N.P. Weatherill. Aspects of Numerical Grid Generation: Current Science and Art. In *AIAA Applied Aerodynamics Conference, Monterey, California*, August 9-11, 1993.

[25] J.F. Thompson. A Reflection on Grid Generation in the 90s: Trends, Needs, and Influences. In *5th International Conference on Numerical Grid Generation*, Mississippi State University, 1996.

[26] D.D. Knight. Parallel Computing in Computational Fluid Dynamics. In *AGARD-CP-578, Progress and Challenges in CFD Methods and Algorithms*, 1996.

[27] D.R. Emerson, A. Ecer, J. Periaux, N. Satofuka and P. Fox (editors). *Parallel Computational Fluid Dynamics, Recent Developments and Advances Using Parallel Computers*. Elsevier Science B.V., The Netherlands., 1997.

[28] A. Jameson. Full-Potential, Euler and Navier-Stokes Schemes. In P.A. Henne, editor, *Applied Computational Aerodynamics, Progress in Astronautics and Aeronautics, Vol. 125*, pages 39–88, 1990.

[29] M.L. Sawley and J.K. Tegnér. A Comparison of Parallel Programming Models for Multi-Block Flow Computations. *Fluid Mechanics Laboratory, École Polytechnique Fédérale de Lausanne*, T-94-23, October 1994.

[30] K.J. Badcock, W. McMillan, M.A. Woodgate, B.J. Gribben, S. Porter and B.E. Richards. Integration of an Implicit Multiblock Code into a Workstation Cluster Environment. In P. Schiano et al., editor, *Parallel Computational Fluid Dynamics: Algorithms and Results using Advanced Computers*, pages 408–415. Elsevier Science B.V. Amsterdam, 1996.

[31] F. Cantariti, L. Dubuc, B.J. Gribben, M. Woodgate, K.J. Badcock and B.E. Richards. Approximate Jacobians for the Euler and Navier-Stokes Equations. *University of Glasgow, Aero Report 9705*, 1997.

[32] L. Dubuc, F. Cantariti, M. Woodgate, B.J. Gribben, K.J. Badcock and B.E. Richards. Solution of the Euler unsteady equations using deforming grids. *University of Glasgow, Aero Report 9704*, 1997.

[33] L. Dubuc, F. Cantariti, M. Woodgate, B. Gribben, K.J. Badcock and B.E. Richards. Solution of the Unsteady Euler Equations Using an Implicit Dual-Time Method. *AIAA Journal*, 36:1417–1424, 1998.

[34] R.M. Gatiganti, K.J. Badcock, F. Cantariti, L. Dubuc, M. Woodgate and B.E. Richards. Evaluation of an Unfactored Method for the Solution of the Incompressible Flow Equations using Artificial Compressibility. *Applied Ocean Research*, 20:179–187, 1998.

[35] J. Sahu, C.J. Nietubicz and J.L. Steger. Navier-Stokes Computations of Projectile Base Flow with and without Mass Injection. *AIAA Journal*, 23(9):1348–1355, 1985.

[36] C.C. Chuang and C.C. Chieng. Supersonic Base-Flow Computation Using Higher-Order Closure Turbulence Models. *Journal of Spacecraft and Rockets*, 33(3):374–380, 1996.

[37] R.E. Childs and S.C. Caruso. Assessment of Modeling and Discretization Accuracy for High-Speed Afterbody Flows. *Journal of Propulsion*, 7(4):607–616, 1991.

[38] F. Magagnato. Computation of Axisymmetric Base Flow with Different Turbulence Models. In *AGARD-CP-493, Missile Aerodynamics*, 1990.

[39] C. Vassilopoulus, K.C. Giannakoglou and K.D. Papailiou. Supersonic Rearward-Facing Step Calculations Using an Explicit Fractional-step Method and a Two-Equation Turbulence Model. In *ETMA Workshop on Turbulence Modelling for Compressible Flow arising on Aeronautics*, 1994.

[40] British Aerospace contribution, Action Group 09. Flow past Missile Afterbodies. In *GARTEUR TP no. 061*, pages 137–144, November 1991.

[41] Matra Defense contribution, Action Group 09. Flow past Missile Afterbodies. In *GARTEUR TP no. 061*, pages 156–160, November 1991.

[42] M-H. Chen, C-C. Hsu and W. Shyy. Assessment of TVD Schemes for Inviscid and Turbulent Flow Computation. *International Journal for Numerical Methods in Fluids*, 12:161–177, 1991.

[43] R.C-C. Luh and C.K. Lombard. Projectile Aerodynamics Prediction with CSCM-S Upwind Implicit Relaxation Algorithm. *AIAA-85-1838*, 1985.

[44] K.J. Moran and P.S. Beran. Navier-Stokes Simulations of Slender Axisymmetric Shapes in Supersonic, Turbulent Flow. *AIAA Journal*, 32(7):1446–1456, 1994.

[45] P.D. Orkwiss and D.S. McRae. Newton's Method for the Axisymmetric Navier-Stokes Equations. *AIAA Journal*, 30(6):1507–1514, 1992.

[46] E. Weide, E.Issman, H.Deconinck and G.Degrez. A Parallel Implicit Multidimensional Upwind Cell Vertex Navier-Stokes Solver for Hypersonic Applications. In *AGARD-R-813, Aerothermodynamics and Propulsion Integration for Hypersonic Vehicles*, 1996.

[47] S.S. Gokhale and R. Suresh. Numerical Computations of Internal Flows for Axisymmetric and Two-dimensional Nozzles. *International Journal for Numerical Methods in Fluids*, 25(5):599–610, September 1997.

[48] B.J. Gribben, K.J. Badcock and B.E. Richards. Shock Reflection Hysteresis in an Underexpanded Jet: a CFD Study. *University of Glasgow, Aero Report 9808*, 1998.

[49] P. Glaister. Flux Difference Splitting for the Euler Equations with Axial Symmetry. *Journal of Engineering Mathematics*, 22:107–121, 1988.

[50] C. Hirsch. *Fundamentals of Numerical Discretisation*, volume 1. Numerical Computation of Internal and External Flows, John Wiley, 1990.

[51] J.H. Heinbockel. *Tensor Analysis and Continuum Mechanics.* Old Dominion University, `http://www.math.odu.edu/~jhh/johnh.html`, 1996.

[52] S. Eskinazi. *Principles of Fluid Mechanics.* Allyn and Bacon, Inc., Boston, 1968.

[53] D.A. Anderson , J.C. Tannehill and R.H. Pletcher. *Computational Fluid Mechanics and Heat Transfer.* Series in Computational Methods in Mechanics and Thermal Sciences, Hemisphere, New York, 1984.

[54] D.C. Wilcox. *Turbulence Modelling for CFD.* DCW Industries, Inc., La Cañada, California, 1993.

[55] A. Berry. Flow Over a Backwards Facing Step. *University of Glasgow, Aerospace Engineering, Summer School in CFD*, September 1997.

[56] C.I.R.A. , Italian Aerospace Research Center. *First Europe-US High Speed Flow Field Database Workshop - Part II.* Naples, Italy. `http://www.cira.it/hsff-2/`, November 12-14, 1997.

[57] GARTEUR Action Group 24. Unpublished ONERA experimental data.

[58] GARTEUR Action Group 24. Unpublished ONERA computational results.

[59] X. Zheng, C. Liao, C. Liu, C.H. Sung and T.T. Huang. Multigrid Methods for Two-Equation Turbulent Models. *Lecture Notes for the Short Course "Advances in Computational Fluid Dynamics"*, Louisiana Tech University, Ruston, Louisiana, `http://www-math.cudenver.edu/~xzheng/publ.html`, June 24-28, 1996.

[60] F.R. Menter. Zonal Two Equation $k-\omega$ Turbulence Models for Aerodynamic Flows. *AIAA-93-2906*, 1993.

[61] A. Hellsten. Implementation of a One-Equation Turbulence Model into the FINFLO Flow Solver. *Helsinki University of Technology, Laboratory of Aerodynamics, Report No. B-49*, September 17, 1996.

[62] J. Henderson, K.J. Badcock and B.E. Richards. Investigation of High Velocity Flows Around Axisymmetric Bodies. In K.D. Papailiou, D. Tsahalis, J. Périaux and D. Knörzer, editor, *Fourth European Computational Fluid Dynamics Conference*, pages 514–519. John Wiley and Sons, U.K., 1998.

[63] J. Delery and R.G. Lacau. Prediction of Base Flows. In *AGARD-FDP-VKI Special Course on Missile Aerodynamics*, April 3, 1987.

[64] GARTEUR Action Group 09. *Flow past Missile Afterbodies, Final Report.* GARTEUR TP no. 061, Main Part, 1991.

[65] J. Sahu and K.R. Heavey. Numerical Investigation of Supersonic Base Flow with Base Bleed. *Journal of Spacecraft and Rockets*, 34(1):62–69, 1997.

[66] B.J. Guidos. Navier-Stokes Computations of Finned Kinetic Energy Projectile Base Flow. *Journal of Spacecraft and Rockets*, 34(4):471–477, 1997.

[67] J. Sahu. Numerical Computations of Supersonic Base Flow with Special Emphasis on Turbulence Modelling. *AIAA Journal*, 32(7):1547–1549, 1994.

[68] GARTEUR Action Group 09. *Flow past Missile Afterbodies, Final Report.* GARTEUR TP no. 061, Appendix 3, 1991.

[69] C.G. Speziale. On Nonlinear $K-l$ and $K-\epsilon$ models of turbulence. *Journal of Fluid Mechanics*, 178:459–475, 1987.

[70] C. Crist, P.M. Sherman and D.R. Glass. Study of the Highly Underexpanded Sonic Jet. *AIAA Journal*, 4(1):68–71, 1966.

[71] F.P. Welsh. Shock Reflection Hysteresis in Low Density Under-Expanded Jets. *DRA Technical Report DRA/DWS/WX9/CR97361*, March 1997.

[72] P. Reijasse, B. Corbel and J. Delery. Flow Confluence Past a Jet-On Axisymmetric Afterbody. *Journal of Spacecraft and Rockets*, 34(5):593–601, Sep-Oct 1997.

[73] S.M. Prudhomme, and H. Haj-Hariri. Investigation of Supersonic Underexpanded Jets using Adaptive Unstructured Finite Elements. *Finite Elements in Analysis and Design*, 17:21–40, 1994.

[74] P.S. Cumber, M. Fairweather, S.A.E.G. Falle and J.R. Giddings. Predictions of the Structure of Turbulent, Highly Underexpanded Jets. *Journal of Fluids Engineering*, 117:599–604, Dec 1995.

[75] V. Salemann and J.M. Williams. New Method of Modelling Underexpanded Exhaust Plumes for Wind Tunnel Aerodynamics Testing. *Journal of Engineering for Gas Turbines and Power*, 111(4):748–754, Oct 1989.

[76] C. Dankert, R.D. Boettcher, G. Detteleff and H. Legge. Plume Size Measurements on Underexpanded Jets in Vacuum Chambers. *AIAA-85-0932*, Oct 1985.

[77] T.F. Hu and D.K. McLaughlin. Flow and Acoustic Properties of Low Reynolds Number Underexpanded Supersonic Jets. *Journal of Sound and Vibration*, 141(3):485–505, Sept 1990.

[78] V.V. Riabov. Aerodynamic Applications of Underexpanded Viscous Jets. *Journal of Aircraft*, 32(3):471–479, May-June 1995.

[79] M. Abbett. Mach Disk in Underexpanded Exhaust Plumes. *AIAA Journal*, 9(3):512–514, March 1971.

[80] J.H. Fox. On the Structure of Jet Plumes. *AIAA Journal*, 12(1):105–107, 1974.

[81] I.S. Chang and W.L. Chow. Mach Disk from Underexpanded Axisymmetric Nozzle Flow. *AIAA Journal*, 12(8):1079–1082, 1974.

[82] D.W. Eastman and L.P. Radtke. Location of the Normal Shock Wave on the Exhaust Plume of a Jet. *AIAA Journal*, 1(4):918–919, 1963.

[83] A.T. Hsu and M.S. Liou. Computational Analysis of Underexpanded Jets in the Hypersonic Regime. *Journal of Propulsion and Power*, 7(2):297–299, 1991.

[84] P. Birkby, J.C. Dent and G.J. Page. CFD prediction of Turbulent Sonic Underexpanded Jets. In *Proceedings of the 1996 ASME Fluids Engineering Summer Meeting. Part 2 (of 3)*, pages 465–470, Jul 7-11 1996.

[85] H.G. Hornung, H. Oertel Jnr. and R.J. Sandeman. Transition from Mach Reflection of Shock Waves in Steady and Pseudosteady Flow With and Without Relaxation. *Journal of Fluid Mechanics*, 90:541–560, 1979.

[86] A. Chpoun, D. Passerel, H. Li and G. Ben-Dor. Reconsideration of Oblique Shock Wave Reflection in Steady Flows. Part I: Experimental Investigation. *Journal of Fluid Mechanics*, 301:19–35, 1995.

[87] J. Vuillon, D. Zeitoun and G. Ben-Dor. Reconsideration of Oblique Shock Wave Reflection in Steady Flows. Part II: Numerical Investigation. *Journal of Fluid Mechanics*, 301:37–50, 1995.

[88] M.S. Ivanov, S.F. Gimelshein and A.E. Beylich. Hysteresis Effect in Stationary Reflection of Shock Waves. *Physics of Fluids*, 7(4):685–687, 1995.

[89] M.S. Ivanov, D. Zeitoun, J. Vuillon, S.F. Gimelshein and G. Markelov. Investigation of the Hysteresis Phenomena in Steady Shock Reflection Using Kinetic and Continuum Methods. *Shock Waves*, 5(6):341–346, 1996.

[90] H.G. Hornung. Regular and Mach Reflection of Shock Waves. *Annual Reveiw of Fluid Mechanics*, 18:33–58, 1986.

[91] G. Ben-Dor. Steady, Pseudo-Steady and Unsteady Shock Wave Reflections. *Progress in Aerospace Sciences*, 25:329–412, 1988.

[92] H.G. Hornung and M.L. Robinson. Transition from regular to Mach Reflection of shock waves Part 2 . The steady-flow criterion. *Journal of Fluid Mechanics*, 123:155–164, 1982.

[93] H. Li and G. Ben-Dor. Application of the Principle of Minimum Entropy Production to Shock Wave Reflections. I. Steady Flows. *Journal of Applied Physics*, 80(4):2027–2037, 1996.

[94] L.F. Henderson and A. Lozzi. Further Experiments on Transition to Mach Reflection. *Journal of Fluid Mechanics*, 94(3):541–559, 1979.

[95] Ames Research Staff. Equations, Tables and Charts for Compressible Flow. *NACA Report 1135*, 1953.

[96] D.J. Azedevo and C.S. Liu. Engineering Approach to the Prediction of Shock Patterns in Bounded High-Speed Flows. *AIAA Journal*, 31(1):83–90, 1993.

[97] L.F. Henderson and A. Lozzi. Experiments on Transition of Mach Reflection. *Journal of Fluid Mechanics*, 68(1):139–155, 1975.

[98] G. Ben-Dor, T. Elperin and E. Golshtein. Monte Carlo Analysis of the Hysteresis Phenomenon in Steady Shock Wave Reflections. *AIAA Journal*, 35(11):1777–1779, 1997.

[99] P. Glansdorff and I. Prigogine. *Thermodynamic Theory of Structure, Stability and Fluctuations*. Wiley-Interscience, U.K., 1971.

[100] M.D. Salas and B.D. Morgan. Stability of Shock Waves Attached to Wedges and Cones. *AIAA Journal*, 21(12):1611–1617, 1983.

[101] H. Li and G. Ben-Dor. Application of the Principle of Minimum Entropy Production to Shock Wave Reflections. I. Pseudosteady Flows. *Journal of Applied Physics*, 80(4):2038–2048, 1996.

[102] R.E. Balzhiser, M.R. Samuels and J.D. Eliassen. *Chemical Engineering Thermodynamics*. Prentice-Hall, U.S.A., 1972.

[103] W.G. Vincenti and C.H. Kruger Jr. *Introduction to Physical Gas Dynamics.* John Wiley & Sons, U.S.A., 1965.

[104] S. Chapman and T.G. Cowling. *The Mathematical Theory of Non-Uniform Gases.* Cambridge University Press, U.K., 1960.

[105] J. D. Anderson Jr. *Hypersonic and High Temperature Gas Dynamics.* McGraw-Hill Book Company, U.S.A., 1989.

[106] F.L. Daum and G. Gyarmathy. Condensation of Air and Nitrogen in Hypersonic Wind Tunnels. *AIAA Journal*, 6(3):458–465, 1968.

[107] F.P. Welsh. Dual Solution Shock Reflection and Shock Layer Instability in Under-Expanded Jets. *DERA Technical Report DERA/WSS/WX9/CR980114/1.1*, 1998.

[108] S. Molder, A. Gulamhussein, E.V. Timofeev and P.A. Voinovich. Focusing of conical shocks at the centre-line of symmetry. *21st International Symposium on Shock Waves, Great Keppel, Australia*, 1997.

[109] J. D. Anderson Jr. *Computational Fluid Dynamics.* McGraw-Hill, U.S.A., 1995.

[110] F.S. Lien, W.L. Chen and M.A. Leschziner. Multiblock implementation of a non-orthogonal, collocated finite volume algorithm for complex turbulent flows. *International Journal for Numerical Methods in Fluids*, 23(6):567–588, 1996.

[111] D. Bohn, H. Schoenenborn and H. Wilhelmi. Numerical simulation of supersonic, chemically reacting flow using an implicit finite volume method. *Journal of Thermophysics and Heat Transfer*, 11(1):45–51, 1997.

[112] F. Grasso and G. Iaccarino. Influence of crossflow and turbulence on vortex flow around a supersonic missile. *Journal of Spacecraft and Rockets*, 35(1):37–45, 1998.

[113] J.A. Ekaterinaris. Numerical simulation of incompressible two-blade rotor flowfields. *Journal of Propulsion and Power*, 14(3):367–373, 1998.

[114] J. Hauser and A. Vinckier. Recent Developments in Grid Generation. In *AGARD-CP-464, Applications of Mesh Generation to Complex 3-D Configurations*, 1989.

[115] J.F. Thompson. General Structured Grid Generation Systems. In *AGARD-CP-464, Applications of Mesh Generation to Complex 3-D Configurations*, 1989.

[116] J.F. Dannenhoffer. Automatic Blocking for Complex Three-Dimensional Configurations. In *NASA-CP-3291, Workshop Proceedings: Surface Modelling, Grid Generation and Related Issues in Computational Fluid Dynamics*, May 1995.

[117] J.F. Dannenhoffer. A Technique for Optimizing Grid Blocks. In *NASA-CP-3291, Workshop Proceedings: Surface Modelling, Grid Generation and Related Issues in Computational Fluid Dynamics*, May 1995.

[118] A. Gaither, K. Gaither, B. Jean, J. Remotigue, J. Whitmire, B.K. Soni, J.F. Thompson, J.F. Dannenhoffer and N.P. Weatherill. The National Grid Project: A System Overview. In *NASA-CP-3291, Workshop Proceedings: Surface Modelling, Grid Generation and Related Issues in Computational Fluid Dynamics*, May 1995.

[119] M.E.M. Stewart. Domain-Decomposition Algorithm Applied to Multielement Airfoil Grids. *AIAA Journal*, 30(6):1457, 1992.

[120] M.E.M. Stewart. A Multiblock Grid Generation Technique Applied to Jet Engine Configuration. In *NASA-CP-3143, Proceedings of the Software Systems for Surface Modelling, and Grid Generation Workshop*, 1992.

[121] J.A. Shaw and N.P. Weatherill. Automatic Topology Generation for Multiblock Grids. *Applied Mathematics and Computation*, 52:355, 1992.

[122] A. Wulf and V. Akdag. Tuned Grid Generation with ICEM CFD. In *NASA-CP-3291, Workshop Proceedings: Surface Modelling, Grid Generation and Related Issues in Computational Fluid Dynamics*, May 1995.

[123] T. Schonfeld, P. Weinerfeld and C.B. Jenssen. Algorithms for the Automatic Generation of 2D Structured Multi-Block Grids. In *NASA-CP-3291, Workshop Proceedings: Surface Modelling, Grid Generation and Related Issues in Computational Fluid Dynamics*, May 1995.

[124] B. Kim and S.D. Eberhardt. Automatic Multi-Block Grid Generation for High-Lift Configuration Wings. In *NASA-CP-3291, Workshop Proceedings: Surface Modelling, Grid Generation and Related Issues in Computational Fluid Dynamics*, May 1995.

[125] Proceedings, 5th International Meshing Roundtable. *Sandia National Laboratories.* 1996.

[126] T.K.H. Tam and C.G. Armstrong. 2D Finite element mesh generation by medial axis subdivision. *Advances in Engineering Software*, 13(5/6):313–324, 1991.

[127] T.K.H. Tam and C.G. Armstrong. Finite element mesh control by integer programming. *International Journal for Numerical Methods in Engineering*, 36:2581–2605, 1993.

[128] C.G. Armstrong, C.L. Toh, D.J. Robinson and T.S. Li. Adaptive Hexahedral Meshing using Solid Primitives. In *Proceedings of the Fourth NAFEMS International Conference on Quality Assurance and Standards in Finite Element and Associated Technologies*, 1993.

[129] M.A. Price and C.G. Armstrong. Hexahedral mesh generation by medial surface subdivision: Part I. Solids wih convex edges. *International Journal for Numerical Methods in Engineering*, 38:3335–3359, 1995.

[130] C.G. Armstrong, R.J. Donaghy and S.J. Bridgett. Derivation of Appropriate Idealisations in Finite Element Modelling. In *Proceedings of the Third International Conference on Computational Structures Technology*, 1996.

[131] W. Haase (editor). *EUROVAL, A European Initiative on validation of CFD Codes.* Notes on Numerical Fluid Mechanics, Vol 42, Braunschweig: Vieweg, Germany, 1993.

[132] A. Parodi. Final Year Project: Two Element Aerofoil. *University of Glasgow, Department of Aerospace Engineering*, 1998.

[133] G. Kalitzin, A.R.B. Gould and J.J. Benton. Application of Two-Equation Turbulence Models in Aircraft Design. *AIAA Paper*, 96-0327, 1996.

[134] B.R. Williams. An Exact Test Case for the Plane Potential Flow About Two Adjacent Lifting Aerofoils. *ARC R&M 3717*, 1973.

[135] R.H.J.M. Otten and L.P.P.P van Ginneken. *The Annealing Algorithm*. Kluwer Academic Publishers, U.S.A., 1989.

[136] J. De Keyser and D. Roose. Load Balancing Data Parallel Programs on Distributed Memory Computers. *Parallel Computing*, 19:1199–1219, 1993.

[137] D. Vanderstraeten and R. Keunings. Optimized Partitioning of Unstructured Finite Element Meshes. *International Journal for Numerical Methods in Engineering*, 38:433–450, 1995.

[138] N.P. Chrisochoides, E.N. Houstis and C.E. Houstis. Geometry Based Mapping Strategies for PDE Computations. In *ACM Supercomputing Conference '91*, pages 115–127, Association for Computing Machinery, New York, 1991.

[139] P. Schiano, A. Ecer, J. Periaux and N. Satofuka (editors). *Parallel Computational Fluid Dynamics, Algorithms and Results Using Parallel Computers*. Elsevier Science B.V., The Netherlands., 1996.

[140] C. Farhat. A Simple and Efficient Automatic FEM Domain Decomposer. *Computers and Structures*, 28:579–602, 1988.

[141] Yufeng Yao. *High Order Resolution and Parallel Implementation on Unstructured Grids*. PhD thesis, Department of Aerospace Engineering, University of Glasgow, 1996.

[142] G. Karypis and V. Kumar. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *University of Minnesota, Department of Computer Science*, Technical Report 95-035, 1995.

[143] B. Hendrickson and R. Leland. The CHACO User's Guide, Version 1.0. *Sandia National Laboratories*, Technical Report SAND93-1301, 1993.

[144] C. Walshaw, M. Cross and M. Everett. Mesh Partitioning and Load Balancing for Distributed Memory Parallel Systems. In B.H.V. Topping, editor, *Advances in Computational Mechanics for Parallel Distributed Processing*, pages 97–104. Saxe-Coburg Publications, Edinburgh, 1997.

[145] R.D. Williams. Performance of Dynamic Load Balancing Algorithms for Unstructured Mesh Calculations. *Journal of Concurrency: Practice and Experience*, 3(5):457–481, 1991.

[146] J.B. Vos, V. Van Kemenade, A. Ytterström and A.W. Rizzi. Parallel NSMB: An Industrialized Aerospace Code for Complete Aircraft Simulations. In P. Schiano et al., editor, *Parallel Computational Fluid Dynamics: Algorithms and Results using Advanced Computers*, pages 49–58. Elsevier Science B.V. Amsterdam, 1996.

[147] C.A. Masson, M. Woodgate, W. McMillan, F. Cantariti, K.J. Badcock and B.E. Richards. The Development of an Affordable High Powered Clustered Computer for Parallel Applications. *submitted for publication*, 1998.

[148] H.P.F. Forum. High Performance Fortran Language Specification. *Scientific Programming*, 2:1–170, 1993.

[149] M.P.I. Forum. MPI: A Message-Passing Interface Standard. *Journal of Supercomputer Applications*, 8, 1994.

[150] V. Sunderam, G. Giest, J. Dongarra and R. Manchek. PVM: A Framework for Parallel Distributed Computing. *Journal of Concurrency: Practice and Experience*, 2:315–339, 1990.

[151] B. Eisfeld, H. Ritzdorf, H. Bleeke and N. Kroll. Portable Parallelization of a 3D Euler/Navier-Stokes Solver for Complex Flows. In *AGARD-CP-578, Progress and Challenges in CFD Methods and Algorithms*, 1996.

[152] Y.P. Chien, J. Huang and A. Ecer. A Two-Stage Computer Load Balancing Method for Parallel Finite Element Analysis Using the Domain Decomposition Approach. In K.D. Papailiou, D. Tsahalis, J. Périaux and D. Knörzer, editor, *Invited Lectures, Minisymposia and Special Technological Sessions of the Fourth European Computational Fluid Dynamics Conference*, pages 164–170. John Wiley and Sons, U.K., 1998.

[153] P. Henriksen and R. Keunings. Parallel Computation of the Flow of Integral Viscoelastic Fluids on a Heterogeneous Network of Workstations. *International Journal for Numerical Methods in Fluids*, 18:1167–1183, 1994.

[154] C. Fischberg, C. Rhie, R. Zacharias, P. Bradley and T. DesSureault. Using Hundreds of Workstations for Production Running of Parallel CFD Applications. In A. Ecer et al., editor, *Parallel Computational Fluid Dynamics: Implementations and Results using Parallel Computers*, pages 9–22. Elsevier Science B.V. Amsterdam, 1995.

[155] Y.P. Chien, A. Ecer, H.U. Akay, F. Carpenter and R.A. Blech. Dynamic Load Balancing on a Network of Workstations for Solving Computational Fluid Dynamic Problems. *Computer Methods in Applied Mechanics and Engineering*, 119:17–33, 1994.

[156] Y.P. Chien, F. Carpenter, A. Ecer and H.U. Akay. Load-Balancing for Parallel Computation of Fluid Dynamic Problems. *Computer Methods in Applied Mechanics and Engineering*, 120:119–130, 1995.

[157] C. Farhat and M. Lesoinne. Automatic Partitioning of Unstructured Meshes for the Parallel Solution of Problems in Computational Mechanics. *International Journal for Numerical Methods in Engineering*, 36:745–764, 1993.

[158] R.A. Rutenbar. Simulated Annealing Algorithms: An Overview. *IEEE Circuits and Devices Magazine*, pages 19–26, Jan. 1989.

[159] R.H.J.M. Otten and L.P.P.P. van Ginneken. *The Annealing Algorithm*. Kluwer Academic Publishers, 1989.

[160] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[161] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller and E. Teller. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[162] C.C. Wong, F.G. Blottner and J.L. Payne. Implementation of a Parallel Algorithm for Thermo-Chemical Nonequilibrium Flow Simulation. *AIAA paper*, 95-0152, 1995.

[163] Y.P. Chien, A. Ecer, H.U. Akay and S.Secer. Communication Cost Function for Parallel CFD in a Heterogeneous Environment Using Ethernet. In P. Schiano et al., editor, *Parallel Computational Fluid Dynamics: Algorithms and Results using Advanced Computers*, pages 1–10. Elsevier Science B.V. Amsterdam, 1996.

[164] A. Ålund, P. Lötsdedt and M. Sillén. Parallel Solution of Industrial Compressible Flow Problems with Static Load Balancing. In P. Schiano et al., editor, *Parallel Computational Fluid Dynamics: Algorithms and Results using Advanced Computers*, pages 336–343. Elsevier Science B.V. Amsterdam, 1996.

[165] K. Stüben, H. Mierendorff, C.-A. Thole and O. Thomas. EUROPORT: Parallel CFD for Industrial Applications. In P. Schiano et al., editor, *Parallel Computational Fluid Dynamics: Algorithms and Results using Advanced Computers*, pages 39–48. Elsevier Science B.V. Amsterdam, 1996.

[166] B. McMillan. Parallel Computing on Workstation Clusters. *Informal Symposium on Implicit Methods, Parallel Computing and Aerospace Applications*, Department of Aerospace Engineering, University of Glasgow, `http://www.aero.gla.ac.uk/Research/ASCU/Index.html`, 14th June 1995.

[167] A. Jameson. Optimum Aerodynamic Design Using CFD and Control Theory. *AIAA-95-1729-CP*, 1995.