

Computational Modelling for Shock Tube Flows

Thesis by

James M. Faddy, B.E. (Hons)

Submitted for the Degree of
Masters of Engineering Science

Department of Mechanical Engineering
The University of Queensland
Brisbane, Australia

2000

(Submitted August 31, 2000)

© 2000

James M. Faddy, B.E. (Hons)

All Rights Reserved

Statement of Originality

The work presented in this thesis is, to the best of my knowledge and belief, original, except as acknowledged in the text. The material has not been submitted, either in whole or in part, for a degree at this or any other university.

James M. Faddy

Abstract

This thesis concerns the computational modelling of high speed, transient, flow processes within facilities such as shock tubes and expansion tubes. Whole facility simulations are possible in reasonable time frames, but require limiting the extent of the modelling to one-dimensional flows and applying phenomenological models to capture macroscopic flow effects. Alternatively, by limiting the analysis to a specific part of the facility, more detailed, multi-dimensional simulations that capture complex flow features directly, become feasible. This thesis looks at the application of these two approaches to finite time diaphragm rupture and shock wave / vortex interactions.

The first part of this thesis concerns the modelling of finite time diaphragm rupture in a quasi-one-dimensional simulation code. One-dimensional flow simulation can produce an approximate analysis of whole facility operation; however, they cannot directly model multi-dimensional phenomena associated with diaphragm rupture. A quasi-one-dimensional code, E1d, was formulated to test two algebraic models that aim to capture the drop in total pressure of the expanding driver gas, assumed to be associated with finite opening diaphragms. The first model assumes an isentropic expansion from the diaphragm throat to the full area of the driven tube where a normal shock processes the flow. The second model uses data from studies of backward facing step experiments to approximate the pressure on the unruptured surface of the diaphragm. Once this pressure is known a ‘mixed out average’ flow state can be calculated by conserving mass momentum and energy. Simulated pressure traces were compared with experimental measurements. Ultimately, the underlying flow mechanisms are three-dimensional and the attempt to model them by a one-dimensional analysis was not very successful.

Rather than modelling whole facility operation, the second part of this thesis concentrates on a detailed multi-dimensional analysis of a small flow region. For this work, a two-dimensional Euler code, MACS2d, based on an adaptively refined mesh was designed

and validated on a series of test problems. Mesh adaption aims to allow an efficient distribution of computational resources by using a fine mesh only where it is needed. Issues relevant to grid adaption such as data structure, refinement criteria, and solution reconstruction are discussed.

The final part of this work applied the MACS2d code to the problem of shock wave vortex interactions. The motivation for this work was the experimental work of Skews [52], in which the multiple interaction of a shock wave with a corner vortex sometimes resulted in a patch of small scale turbulent flow. Rather than trying to model the ensemble of such interactions characteristic of a turbulent flow, the simulations focus on the vortex roll-up and the subsequent interaction of the vorticity with the reflecting shocks. There are two scales of motion. On a large scale, the reflected shock wave attempts to compress the large corner vortex, while on a smaller scale, discrete vortices on the periphery of the large vortex are produced as a result of the Kelvin-Helmholtz roll up of the shear layer shed from the corner. Whilst the large scale interactions can be modelled inviscidly, it is found that the detailed distribution of vorticity is viscosity dependent; grid resolution playing an important role. The simulations reproduced very well the flow features seen in the experiments, but there is still some ambiguity regarding the final state of the vortex.

Acknowledgements

First and foremost I would like to sincerely thank my friend and supervisor Peter Jacobs for the guidance and encouragement offered to me throughout the course of this project. The independence he allowed me, whilst always finding time for discussion was greatly appreciated. This in conjunction with his 'harsh but fair' sense of humour made for a most enjoyable 18 months.

Thanks must also go to my associate supervisor Professor R. Stalker and other members of the hypersonic group for their insightful comments. On-going discussions with fellow postgrads Richard Goozee, Ben Stewart, Vincent Wheatley and Kevin Austin were most appreciated. Your unique way of 'bringing me back to earth' whenever I made a breakthrough will not be forgotten!

This work would not have been possible without discussions with Professor B. Skews and the provision of data which provided the motivation for this work. Discussions with Professors P. Voinovich and E. Timofeev during my time in Japan provided many useful ideas. The hospitality shown to me during this time by the institute of fluid science at Tohoku University and in particular Professors K. Takayama and T. Saito was nothing short of superb.

The assistance of Barry Daniel, Barry Allsop and Steve Kimbal in administering the local computer network is appreciated

Financial support for this work, provided by a University of Queensland Postgraduate Research award and a Departmental Scholarship is gratefully acknowledged.

Thanks must also go to my parents and family for their support and patience. Finally, a special thanks to my friends at the University of Queensland Water Ski club. Without you, this thesis would surely have been finished sooner!

Contents

Statement of Originality	iii
Abstract	v
Acknowledgements	vii
Nomenclature	xi
1 Introduction	1
1.1 Modelling Shock Tunnel Phenomena	2
1.2 Thesis Outline	5
2 The Governing Equations of Gas Dynamics	7
2.1 Characteristics of the Flow equations	8
2.2 Finite-Volume Formulation	10
2.2.1 Determining the Interface State	11
2.2.2 Time Integration	13
3 One-Dimensional Flow Modelling Techniques	15
3.1 Example of Current Modelling Inaccuracy	16
3.2 A Quasi-One-Dimensional Simulation Code: E1d	19
3.3 E1d Test Cases	20
3.3.1 The Ideal Shock Tube problem	20
3.3.2 Shock Wave Propagation through an Area Change	22
3.4 Modelling Diaphragm Rupture	23
3.4.1 The Nozzle-Shock Model	24
3.4.2 The Base Pressure Model	25
3.4.3 Model Implementation	26

3.4.4	Results	26
3.5	Shock Tunnel with Driver Restrictor	28
4	Multi-Dimensional Flow Modelling	31
4.1	Computational Modelling	31
4.1.1	Discretisation and Solution Adaptive Techniques	32
4.1.2	Data Storage	33
4.1.3	Implementation of Data Storage	36
4.1.4	Accessing Neighbour Data	37
4.1.5	Refinement and Coarsening Procedure	38
4.2	Solution Procedure	45
4.2.1	Time Integration	45
4.2.2	Reconstruction on Adaptive Grids	48
4.2.3	Boundary Conditions	50
4.3	Test Cases	51
4.3.1	Sod's One-dimensional Shock Tube Problem	52
4.3.2	Mach 3 Flow over a Step	56
4.3.3	Shock Wave Diffraction around a 90 degree Sharp Corner	61
5	Shock-Vortex Interactions	67
5.1	Compressible Turbulence	68
5.1.1	Previous Modelling Work	70
5.2	Interaction with a Mach 1.5 Shock Wave	71
5.3	Interaction with a Mach 2.0 Shock Wave	82
6	Conclusion and Recommendations	89
6.1	Quasi-One-Dimensional Modelling	89
6.2	Two-Dimensional Flow Simulations	90
6.3	Shock Wave Vortex Interactions	92
	Bibliography	95

Nomenclature

Symbols

a	: sound speed m/s
A	: area m ²
C_p	: specific heat at constant pressure, J/kg K
C_v	: specific heat at constant volume, J/kg k
e	: Intensive internal energy, J/kg
E	: Intensive total energy, J/kg
f	: species mass fraction
\mathbf{F}	: Flux vector
F_{wall}	: wall friction
F_{loss}	: pipe fitting loss
h	: intensive enthalpy, J/kg
i	: generic counter
l	: refinement level
L	: Length, m
\hat{n}	: unit normal vector
M	: Mach number
P	: pressure, Pa
q	: heat flux J/(m ² .s)
\mathbf{q}	: Heat flux vector, W/m ²
\mathbf{Q}	: generalised flow variable
Q	: Vector of source terms
R	: radius, m; Gas constant, J/kg K
Re	: Reynolds number
s	: van Albada limiter term

S	: Surface area, m^2
t	: time, s
T	: temperature, K
\mathbf{T}	: Matrix of viscous stresses, Pa
t_c	: characteristic time, s
u	: velocity m/s
\mathbf{u}	: velocity vector, m/s
\mathbf{U}	: Vector of conserved variables
V	: volume, m^3
x	: cartesian coordinate, m
y	: cartesian coordinate, m
z	: cartesian coordinate, m
α	: grid compression parameter
β	: compression parameter
γ	: Ratio of specific heats C_p/C_v
δ	: increment
Δ	: Difference
ϵ	: refinement indicator; dissipation of Kinetic Energy
η	: sensitivity parameter
κ	: Blending paramter
λ	: bulk viscosity Pa s
μ	: Molecular viscosity, Pa s
ν	: Kinematic viscosity, m^2/s
ρ	: density, kg/m^3
τ	: viscous stress, Pa
ω	: strain
ϕ_m	: mass flux
ϕ_{mom}	: momentum flux
ϕ_{en}	: energy flux

Introduction

The design of aerospace vehicles requires a detailed knowledge of the interactions between flight surfaces and the ambient atmosphere through which they pass. The resulting aerodynamic pressure as well as heating loads are among the critical parameters inherent to a successful design. Insight into these conditions is obtained through a mix of analytical, computational and ground based experimental techniques. The recent paradigm shift embraced by the motto “faster, better, cheaper” calls for continual improvement in these areas through a greater understanding of the physical processes involved.

One such example is in the design of the aeroshell heat shield necessary for atmospheric re-entry. Aeroshells typically have a conical skirt with a spherical nose and, for given flow conditions, the heat transfer is governed by

$$q \propto \frac{1}{\sqrt{R}} \quad (1.1)$$

where R is the radius of curvature of the spherical nose; Figure 1.1 below illustrates this. Hence, heat shields with smaller nose radii incur larger heating loads. A heat shield with a larger nose radius and larger cone angle, whilst incurring lower heat transfer, is also less stable aerodynamically. We thus have to make a compromise as the requirement for a

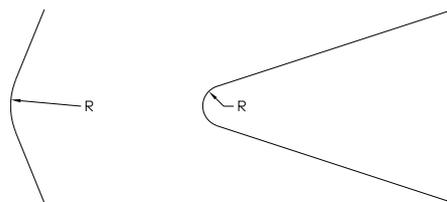


Figure 1.1: Characteristic heat shield shapes.

small, stable aeroshell leads to the need for a thicker, heavier heat shield. Table 1.1 shows

	Viking	Pathfinder	Pioneer
Destination	Mars	Mars	Venus
Base Area	9.62 m ²	5.52 m ²	1.59 m ²
Mass	980.8 kg	585.3 kg	316.4 kg
Nose Radius	0.878 m	0.664 m	0.363 m
Base Radius	1.75 m	1.325 m	0.711 m
Cone Half Angle	70°	70°	45°

Table 1.1: Heat shield specifications for three entry vehicles.

relevant data for ‘specific’ examples of heat shields used on 3 re-entry vehicles.

The increase in heat load coupled with the need to decrease the mass of the heat shield, leads to the need for better experimentally-derived data that can be used by aeroshell designers. Shock tubes, operating in either reflecting or non-reflecting modes, are one of the few classes of ground-based devices capable of full simulation. Although shock tubes are nominally capable of providing high enthalpy test flows, there are some anomalous behaviours that appear at the high enthalpy regime of facility operation. These behaviours lead to decreased performance of the facility, both in terms of the enthalpies and pressures achieved and in a decreased quality of the test flow. Since the simulation of atmospheric entry from interplanetary trajectories involves tests under very high enthalpy conditions, a better fundamental understanding of shock tube behaviour is desirable.

1.1 Modelling Shock Tunnel Phenomena

A large portion of shock wave research at the University of Queensland is conducted in the T4 shock tunnel. This facility is a free piston driven shock tunnel comprising a 26 m compression tube coupled to a 10 m long shock tube. A schematic view of its operation is shown in Figure 1.2.

The operating cycle of the shock tunnel involves a transfer of energy from a large compressed air reservoir, through several intermediate stages, to a small amount of test gas initially contained in the shock tube. The process begins with the release of high pressure air from the reservoir onto the back of a piston, driving it down the compression tube. This compresses the driver gas (typically air, nitrogen or argon) causing an increase in pressure, temperature and consequently speed of sound. Eventually the rupture pressure of the primary diaphragm is exceeded, and the driver gas expands into the shock tube driv-

ing a strong shock wave into the test gas. The incident shock wave compresses and heats the test gas before reflecting from the nearly-closed end of the tube at the nozzle throat. The shock wave travels back up the shock tube through the processed gas causing it to be further heated and compressed to the point where the rupture pressure of the secondary diaphragm is reached. The thin secondary diaphragm ruptures and the processed test gas expands through the nozzle into the test section. The test time begins when steady flow is established through the nozzle and continues until either upstream expansion waves arrive and change flow conditions or until driver gas contaminates the test gas. The overall process is shown on an $x-t$ diagram for tailored conditions in Figure 1.2; for this idealised case, the reflected shock brings the driver gas to rest and no additional waves are created as it passes through the contact surface. A detailed understanding of overall facility per-

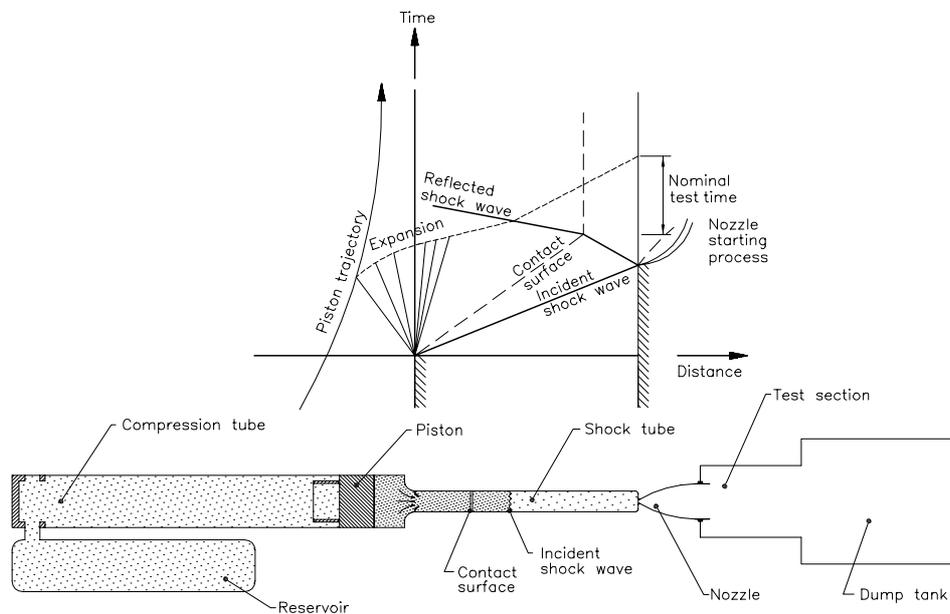


Figure 1.2: Schematic of the T4 free piston reflected shock tunnel including $x-t$ diagram for tailored conditions.

formance is an important aid to the design of improved shock tunnels and expansion tubes [21], the determination of test flow parameters [25] and in the establishment of boundary conditions for more detailed analysis of specific operating regions [37].

Traditional models of shock tube phenomena took a quasi-static approach in which the cascade of energy transferring processes are considered separately. The input for any given section coming solely from the output of the previous process. Improvements to these models have been made by using simple analyses and correlations that aim to model more complicated flow effects. Models for piston dynamics [56] as well as shock

formation from compression waves [67] have been incorporated, however, the model is ultimately limited by the quasi-static assumption.

Whilst such techniques are useful for estimating flow conditions, fully coupled numerical simulations are required to gain greater insight. These simulations solve simplified forms of the governing equations which retain sufficient terms to capture the most important aspects of the flow physics [25, 22, 37]. Computing power has increased in the last few decades to the point where such simulations of the entire facility can be performed in a relatively short time frame.

As we attempt to model a more challenging regime however, such studies are restricted by the applicability of the underlying approximations used to simplify the analysis. The inclusion of more terms from the governing equations provides greater insight into the flow processes occurring, however, this comes at the expense of computational time. Ultimately we must model the full three dimensional Navier Stokes equations with sufficient spatial and temporal resolution so as to resolve even the finest scales of motion. Unfortunately, this is, and for the foreseeable future will remain too computationally taxing to be of practical interest for anything but the simplest of flow geometries.

Depending on the problem at hand and physical time scale in which we want results, we are forced to limit the extent of the motions resolved exactly and incorporate mathematical models tuned to describe more complex processes. Such models that strive to resemble the macroscopic effect of unresolvable flow features can be applicable over a range of flow scales including adding boundary layer mass entrainment effects in a quasi-one dimensional simulation [8] right down to subgrid LES models used in turbulence modelling [39].

This thesis examines techniques for creating such models through analytical analysis, experimental data and high resolution computational simulations that model the next level of physical complexity. Of particular interest are shock-vortex interactions both as an explanation for decreased driver gas total pressure in shock tubes and as a more fundamental mechanism for decaying turbulence kinetic energy.

1.2 Thesis Outline

The aims of this thesis are threefold:

- To investigate ways of modelling finite time diaphragm rupture in a quasi-one-dimensional simulation code.
- To develop a new CFD code that implements solution adaptive remeshing for high resolution simulations.
- To investigate numerically the multiple interactions between a shock wave and a corner vortex.

The original scope of the (then PhD) project included a three-dimensional analysis of shock vortex interactions. Due to the reduction in scope of the project to a Masters thesis, this was not possible, however, two-dimensional simulation results are presented.

The thesis is organised into six chapters as follows:

Chapter 2. In this chapter the governing flow equations are presented and the applicability of some simplifications discussed. The general framework for the computational solution of the flow equations is presented. Finite volume discretisation and numerical integration is described.

Chapter 3. One-dimensional simulation techniques are analysed in this chapter. Models that attempt to correct the macroscopic flow field for features unresolvable under the quasi-one dimensional assumption are discussed. A quasi-one dimensional simulation code is formulated, demonstrated and used to compare some models for diaphragm rupture with experimental traces.

Chapter 4. In this chapter an adaptive strategy is proposed as a means of allowing high resolution simulations to be computed in a time frame that is physically realisable. Issues such as interpolation, reconstruction, and adaption parameters are discussed through the formulation and validation of a structured grid adaption code (MACS2d).

Chapter 5. Compressible turbulence and in particular the role of shock-vortex interactions in the energy cascade process are discussed in this chapter. One scale of such motions is modelled by simulating the multiple interactions of a shock wave with a corner

vortex. Simulation results and previous experimental data are used to explain the generation of fine scale vortical structures during the interactions.

Chapter 6. Finally, a summary of the thesis is presented. Conclusions are drawn and recommendations made concerning future work.

The Governing Equations of Gas Dynamics

The flow of compressible fluids in the continuum regime are thought to be described by the Navier Stokes equations. The integral form of these equations in conservation form can be written as

$$\frac{\partial}{\partial t} \int_V \mathbf{U} dV + \int_S \mathbf{F} dS = \int_V \mathbf{Q} dV \quad (2.1)$$

where \mathbf{U} is the vector of conserved properties, \mathbf{F} the flux vector and \mathbf{Q} the vector of source terms. S is the control surface bounding the control volume V . For three-dimensional flow without heat addition, the source term \mathbf{Q} is zero, whilst the vectors \mathbf{U} and \mathbf{F} can be written as

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho \mathbf{u} \\ \rho E \end{bmatrix} \quad \mathbf{F} = \begin{bmatrix} \rho \mathbf{u} \cdot \hat{n} \\ \rho \mathbf{u} \mathbf{u} \cdot \hat{n} + P \hat{n} - \mathbf{T} \hat{n} \\ \rho E \mathbf{u} \cdot \hat{n} + P \mathbf{u} \cdot \hat{n} - \mathbf{u} \cdot (\mathbf{T} \hat{n}) - \mathbf{q} \cdot \hat{n} \end{bmatrix},$$

where \hat{n} is the outward pointing normal to the control surface S . These equations express conservation laws for mass, momentum and energy in terms of the fluid velocity vector $\mathbf{u} = [u_x, u_y, u_z]^T$ and the variables density, static pressure and total energy (ρ , P , and E respectively). Viscous heat fluxes are contained in \mathbf{q} while viscous stresses are contained in the the matrix \mathbf{T} . Under Stokes' hypothesis, these stresses are related to fluid strains (a function of \mathbf{u}) via the coefficient of viscosity μ .

A relevant equation of state can be used to relate the density and internal energy to any one of the other thermodynamic state variables pressure, temperature and sound speed (P ,

T and a respectively). For a calorically perfect gas we use

$$P = \rho(\gamma - 1)e \quad (2.2)$$

Finally the system of equations is closed by the definition of Total energy

$$E = e + \frac{1}{2}|\mathbf{u}|^2 \quad (2.3)$$

2.1 Characteristics of the Flow equations

Solution techniques for the Navier-Stokes (N-S) equations must address the fact that their mathematical behaviour depends critically upon the flow regime being modelled. At supersonic speeds the steady-state N-S equations are spatially hyperbolic. Flow properties at a point in space influence only downstream properties bounded by lines termed characteristics. This inability for downstream properties to influence the upstream flow allows efficient space marching techniques to be used. Under subsonic conditions, however, the steady-state N-S equations are spatially elliptic. Characteristics are imaginary and information can propagate in all directions. As the Mach number is decreased, diffusive transport phenomena becomes increasingly dominant over fluid convection. In the limit of zero Mach number the flow is governed by the incompressible N-S equations. The assumed infinite sound speed in such fluids means that any given point in the flow is influenced by the entire flow domain.

These differences in behaviour must be addressed when analysing practical flows which contain both subsonic and supersonic regions. Rather than trying to employ different solution techniques in different flow regions [41], a universal technique is devised by considering the transient terms; Regardless of the flow regime the equations exhibit hyperbolic behaviour with respect to time allowing a time marching solution technique. By limiting the extent of acoustic wave propagation during each time step, the equations for the whole flow domain can be treated in the same manner.

The transport phenomena of viscosity and heat conduction, give rise to a spectrum of time and length scales all of which must be resolved if a time accurate description of the flow field is sought. Direct Numerical Simulation (DNS) is the name given to high

resolution schemes that attempt to model all such scales of motion. This technique has been applied successfully to a range of problems particularly for incompressible flows at low Reynold's numbers. As the Reynold's number is increased, however, so do the spectrum of length and time scales of the resulting eddying motions. The number of grid points that must be used to resolve all scales is proportional to $Re^{\frac{9}{4}}$ [44] which quickly becomes excessive.

Large Eddy simulation (LES) addresses this problem by modelling directly only the large scales of motion which are influenced greatly by the geometry of the problem. The simplifying assumption comes with the hope that the small scale motions neglected by the computational simulation are more universal and less affected by the boundary conditions. Their effect is incorporated by mathematical subgrid models that are dynamically tuned as the simulation progresses. The computational costs of LES are large and as such much effort in recent times has been devoted to the development of turbulent models that aim to capture the average effect of turbulent phenomena. The Reynolds averaged N-S equations are solved with complete disregard for the scales of turbulent motions captured directly; their effects being incorporated by the determination of a turbulent viscosity as specified by the chosen model. Such simulations have enjoyed a high degree of success particularly as a means of providing insight into problems where shock-boundary layer interaction effects are important.

A fortunate consequence of the high Reynolds numbers exhibited in high speed compressible flows, however, is that viscous effects are confined to thin boundary layers. If the frictional drag associated with these boundary layers can be neglected, it is suffice to model pressure drag which is essentially inviscid in nature. The Euler equations describe the motion of such inviscid fluids in the continuum regime and are obtained by setting the viscous stresses and heat fluxes to zero in equation 2. These are the equations solved by the two dimensional computational fluid dynamics program MACS2d developed for the simulation of shock-vortex interactions. A more detailed description of the code and the applicability of the Euler equations to the phenomenon of shock-vortex interactions is deferred until Chapter 4.

Further simplifications to the multidimensional Euler equations are permitted if we can assume flow properties to be uniform in any of the spatial coordinate directions. Flow simulations of entire shock tunnels are primarily concerned with average values of condi-

tions across the breadth of the tube and as such it is sensible to model the flow exclusively in the streamwise direction. If we neglect flow variations in the other two coordinate directions but allow gradual variations in cross sectional area the quasi-one dimensional (q1d) Euler equations are obtained. They are the one-dimensional analogue of the multi-dimensional Euler equations described earlier with an additional source of momentum arising from the differential pressure force due to the varying cross sectional area. Compensation is made for the lack of viscous terms by using friction factors and heat transfer correlations to model the effect of viscosity at the tube walls.

The q1d Euler equations are the building blocks of the computational fluid dynamics code E1d to be discussed in detail in Chapter 3.

All of the simplified forms of the governing flow equations just presented represent coupled sets of nonlinear Partial Differential equations. Closed form analytical solutions to these equations in general do not exist and we are forced to consider numerical methods as a means for obtaining approximate solutions.

2.2 Finite-Volume Formulation

Numerical techniques for solving the flow equations involve discretising the flow domain, and integrating the resulting approximate ordinary differential equations at each of the computational nodes. Finite element, finite volume and finite difference schemes are three techniques for obtaining a system of such equations. Whilst all three should produce results that converge to the same solution in the limit of an infinitely fine computational domain, there are advantages to be gained from each depending on the problem requiring solution. For compressible flows, containing discontinuous flow domains, conservation must be ensured. The finite volume technique involves dividing the domain into non-overlapping finite-control-volumes on which the integral form of the governing equations are applied. A weak solution of this form permits discontinuities whilst ensuring global conservation. Additionally, complicated boundary conditions for complex flow domains can be implemented in a relatively straight-forward manner. Figure 2.1 shows an example of a q1d and 2d finite volume discretisation for an expanding flow domain. Algebraic equations are obtained for each control volume by approximating the volume and surface integrals using quadrature formulae. Volume integrals can be evaluated with second order

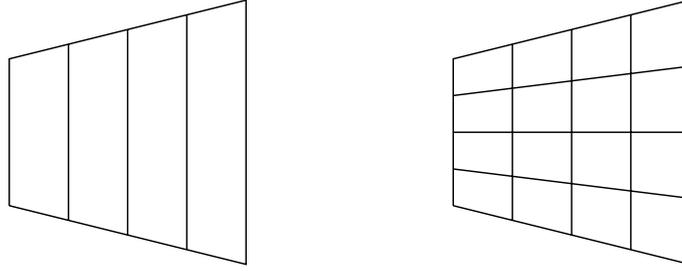


Figure 2.1: Q1d (left) and 2d (right) Finite Volume discretisation of an expanding domain

accuracy by the product of the mean value and the cell volume whilst surface integrals are calculated by summation over the sides of the cell. The integral on each face being approximated by the midpoint rule. Under this scheme the semi-discrete form of the governing equations 2.1 are written for each cell as

$$\frac{\partial \mathbf{U}_{cell}}{\partial t} = \mathbf{Q}_{cell} - \frac{1}{V} \sum_{if} \mathbf{F}_{if} A_{if} \quad (2.4)$$

with A and V being the cell edge interface area and cell volume respectively. The discretised equations applied to each control volume can be advanced in time from an initial solution once a technique for determining the interface fluxes is specified.

2.2.1 Determining the Interface State

Finite volume solution of the flow equations requires the determination of fluxes at the interfaces between cells. Since flow properties are known only at the cell centres some form of approximation must be made about the variation of flow properties across the domain in order to compute these entities. The simplest assumption about the flow domain is that of piecewise constant flow states. The first order scheme of Godunov [16] involves the solution of the Riemann problem at the interface between two such flow states. Whilst this scheme can produce stable results, its low order of accuracy introduces excessive dissipation.

For smooth continuous flow fields interpolation schemes of any order can be devised to approximate the variation of flow properties. Such schemes when applied to discontinuous flow domains instigate high frequency numerical error however, which can lead to code failure. This can be overcome by the introduction of non-linear limiters which act to restrict the action of the higher order interpolation schemes around flow discontinuities

whilst having no effect in regions of smooth flow [29]. The process of interpolation and limiting is termed reconstruction. Using this technique higher order sequels to Godunov's method have been devised [63].

Reconstruction schemes can be devised that prohibit the generation of new extrema (Total Variation Diminishing or TVD schemes), place a bound on the generation of extrema (Total Variation Bounded or TVB schemes) or allow any reduction in variation between timesteps to be regained at the next (Essentially Non-Oscillatory or ENO schemes). The codes presented in this thesis use the van-Albada [62] limiter, which is thought to be TVB [64], with Johnston's modification for non-equispaced cells [29]. With reference to Figure 2.2 this scheme can be described in MUSCL (monotone upwind schemes for conservation laws) like form by

$$Q_{i+1/2}^L = Q_i + \frac{s_i}{4}[(1 - s_i\kappa)\Delta_i^- + (1 + s_i\kappa)\Delta_i^+], \quad (2.5)$$

and,

$$Q_{i+1/2}^R = Q_{i+1} - \frac{s_i}{4}[(1 - s_i\kappa)\Delta_{i+1}^+ + (1 + s_i\kappa)\Delta_{i+1}^-], \quad (2.6)$$

where,

$$\Delta_i^- = Q_i - Q_{i-1}$$

$$\Delta_i^+ = Q_{i+1} - Q_i,$$

and,

$$s = \frac{2\Delta^+\Delta^- + \epsilon}{(\Delta^+)^2 + (\Delta^-)^2 + \epsilon};$$

Here κ is a blending parameter bounded by $0 \leq \kappa \leq 1$ that governs the influence of up and downstream components during interpolation. Additionally, the parameter ϵ is introduced to avoid division by zero and is set slightly higher than machine precision.

Fluxes of mass, momentum and energy are calculated from the reconstructed states either side of the interface by an appropriate flux calculator. There exists a myriad of such flux solvers each one exhibiting its own successes and failures. The flux calculator used in the codes developed in this thesis is an updated version of the approximate Riemann solver developed by Jacobs [24]. Approximate Riemann solvers have the advantage of

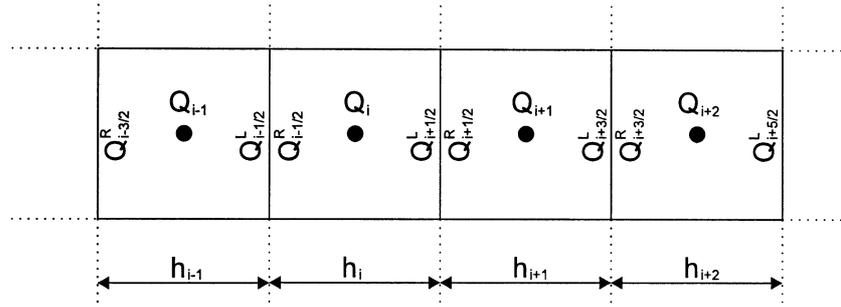


Figure 2.2: Cell configuration for reconstruction (Figure 3.1 in [29]).

being significantly less computationally intensive than their exact counterparts, whilst still retaining sufficient accuracy.

2.2.2 Time Integration

Given the current flow state, the discretised equations can be advanced in time by selecting an appropriate numerical integration technique. Schemes are classified as being either explicit, implicit or a mixture of the two. Explicit integration uses knowledge of only the current flow state and as such is not very computationally intensive. The equations are advanced in small time steps governed by strict stability criteria. For example, a wave starting at a cell interface should not cross more than half of the cell width during a time step.

Schemes of any order can be devised by taking any number of ‘exploratory steps’ aimed at producing a more accurate estimate of the temporal derivative. This comes at the price of increased computational effort and precious memory storage must be used to keep track of the intermediate results. In contrast, implicit schemes use knowledge of the flow state at the end of the time step. This need for ‘prior knowledge’ results in a set of simultaneous equations which must be solved by an iterative technique. The resulting schemes are more computationally intensive; however, numerical stability is maintained for relatively large time steps.

For the flows considered in this thesis the time-scales associated with the phenomena of interest are short. Hence whilst implicit integration would permit large time-steps from a numerical perspective, many of the physical interactions occurring would be lost. For

this reason we employ the explicit two stage predictor-corrector scheme defined by.

$$\begin{aligned}
 \Delta \mathbf{U}^{(1)} &= \Delta t \frac{d\mathbf{U}^{(n)}}{dt} , \\
 \mathbf{U}^{(1)} &= \mathbf{U}^{(n)} + \Delta \mathbf{U}^{(1)} , \\
 \Delta \mathbf{U}^{(2)} &= \Delta t \frac{d\mathbf{U}^{(1)}}{dt} , \\
 \mathbf{U}^{(n+1)} &= \mathbf{U}^{(1)} + \frac{1}{2} (\Delta \mathbf{U}^{(2)} - \Delta \mathbf{U}^{(1)}) ,
 \end{aligned} \tag{2.7}$$

where the subscripts (1) and (2) indicate intermediate results.

Stability requirements for explicit integration schemes dictate that the time step is limited to

$$\delta t \leq CFL * t_c \tag{2.8}$$

where CFL is the Courant-Friedrichs-Lewy number, and t_c is the smallest characteristic timescale of physical processes occurring in the flow. For the case of the Euler equations, only the phenomenon of fluid convection is considered, and t_c becomes the shortest time taken for an acoustic wave to traverse any single control volume in the domain. For a given control volume in three-dimensional space

$$t_c = \min \left(\frac{\Delta x}{|u| + a}, \frac{\Delta y}{|v| + a}, \frac{\Delta z}{|w| + a} \right). \tag{2.9}$$

Whilst stability is predicted for $0 \leq CFL \leq 1$ experience shows it to be very problem dependent with CFL numbers less than 0.7 generally producing stable results.

One-Dimensional Flow Modelling Techniques

In order to model accurately the flows in impulse facilities such as free piston shock tunnels and expansion tubes it is necessary to consider the interplay between gas dynamic interactions, piston dynamics and viscous effects. Simulation codes have been developed that solve the quasi-one-dimensional Euler equations with the addition of empirical correlations to model phenomena such as friction and heat transfer [18, 25, 37, 40]. These codes allow for multiple diaphragms, pistons and gas slugs, in a flexible manner which makes them ideal for analysing the performance of an entire facility with a reasonable amount of computational effort.

Improved accuracy can be obtained by tracking singularities in the flow, such as shocks and contact surfaces [12], and adapting the computational domain accordingly [34]. By incorporating elements of a characteristics solution in this manner, discontinuities are no longer smeared over several cells. However, solutions are still limited by the applicability of the models embodied in the simulation codes. Other researchers have tried to address the fundamental shortcomings of the one-dimensional analysis by adding additional terms for phenomena such as mass-entrainment from the core flow into the boundary layers [8] as well as sudden changes in tube geometry [22]. Despite the advances and associated success for a range of facilities operating at various conditions [26, 22, 37], there still exist operating conditions where the present simulation models are not accurate [28] [50].

Of particular interest are operating conditions in which the finite time required for the opening of the main diaphragm significantly affects the flow development in the shock tube. In a previous study, Ikui & Matsuo [20] modified White's [67] model in which the primary shock was formed from the coalescence of compression waves that were pro-

duced from the gradual opening of the diaphragm. This was a purely one-dimensional flow mechanism and was aimed at explaining higher-than-ideal incident shock speeds. In contrast, the experimental study by Roberts & East [47] used a fixed-geometry restriction at the main diaphragm location to intentionally decrease the total pressure of the driver gas as it expanded into the shock tube and thus provide a high enthalpy flow with rarefied conditions. The presence of the restriction introduced a strong non-isentropic process that was assumed to consist of a stationary oblique shock pattern. More recent multi-dimensional simulations [43] have confirmed that, due to the diaphragm's finite rupturing time, the complex three-dimensional flow in the region downstream of the opening diaphragm includes oblique shocks, vorticity and a highly-distorted contact surface. Although transient, the oblique shock waves persist for some time after the diaphragm has opened and appear to play an important role which should not be neglected for operating conditions where the driver gas expands to moderately high Mach numbers. In this chapter we consider two models that attempt to include the effects of the shock structure downstream of the opening diaphragm in a one-dimensional simulation code.

3.1 Example of Current Modelling Inaccuracy

Although the T4 facility was designed for high enthalpy operation, many shots are done at low enthalpies of about 3 MJ/kg. Jacobs *et al* [28] investigated the conditions produced using both helium and argon drivers in this regime. A quasi-one dimensional Lagrangian code [25] was used for the study; the advantage gained through the use of a heavier driver gas can be seen in Figure 3.1. Since the heavier driver gas has a lower sound speed a constant reservoir pressure is maintained for a longer period of time. This results in a relatively long period of constant pressure test gas. Of greater significance to the present study however, is that whilst excellent agreement is found between computation and experiment for the helium driver, there exists a significant discrepancy for the argon driver simulation. Under this scenario the simulation computes over-tailored conditions. The simulated driver gas has too much momentum (or too high a total pressure) to be brought to rest by the reflected shock wave alone; a secondary shock wave propagates into the test gas causing the rise in simulated supply pressure evident just after $t = 222$ ms. This modelling inaccuracy provides the motivation for a more detailed description of the

complex interactions occurring in the shock tube section.

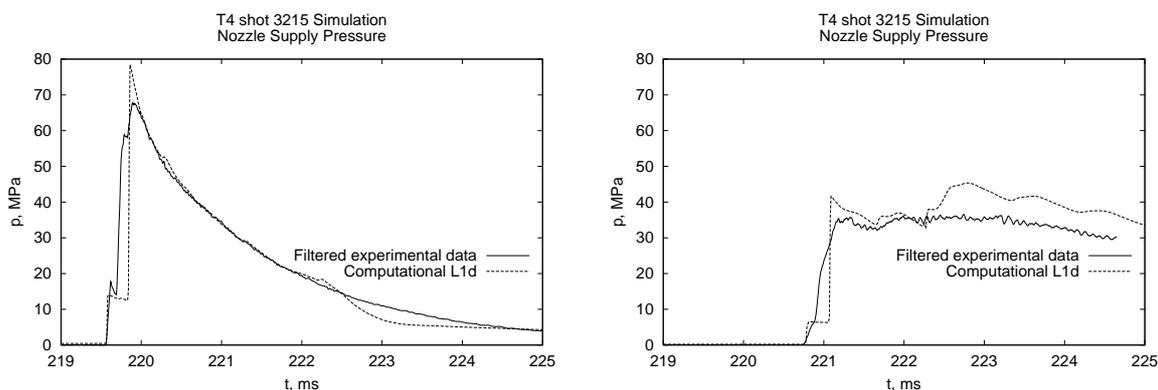


Figure 3.1: Comparison of the experimental nozzle supply pressure and that predicted by L1d for shot 1098(Helium Driver) and 3215(Argon Driver) [28].

In order to construct a model representative of the highly non-isentropic flow in this region it is necessary to modify existing approximations. Traditional models of diaphragm rupture assume that the material is removed instantaneously. From a computational view point this means that two cells change instantaneously from boundary cells to internal cells separated by a common edge. In reality, large steel diaphragms rupture from the centre out in a petalling manner [43] over a period of approximately $500 \mu s$ [69]. As we have hypothesised earlier, the errors associated with the assumption of instantaneous diaphragm removal may become unreasonably high as shock tube operation is pushed to a regime where the time scale for shock processing becomes shorter.

A detailed study of the diaphragm rupture process was conducted by Petrie [43, 42]. His study used a two-dimensional axisymmetric inviscid model in which the boundary conditions on the edges of cells along the diaphragm interface are modified in accordance with diaphragm rupturing theories [48]. The results for a model of the Langley expansion tube are reproduced in Figure 3.2. The bottom half of each frame represents pressure contours while the top half represents density contours thus indicating the contact surface as well as the shocks. Oblique shocks form in the radially expanding driver gas in order to redirect the gas along the tube walls and they appear to remain in the shock tube long after the diaphragm has fully opened. These shocks form the mechanism for the previously mentioned non-isentropic process that decreases the driver gas momentum (or total pressure). The strength of these oblique shock waves is Mach number dependent which offers an explanation for the breakdown of the quasi-one-dimensional flow model when applied

to shot 3215 (argon driver) as opposed to its success for shot 1098 (helium) despite the fact that both shots generated the same incident shock speed. Heavier driver gases such as argon have a lower sound speed and for a given test gas enthalpy will expand to a higher Mach number.

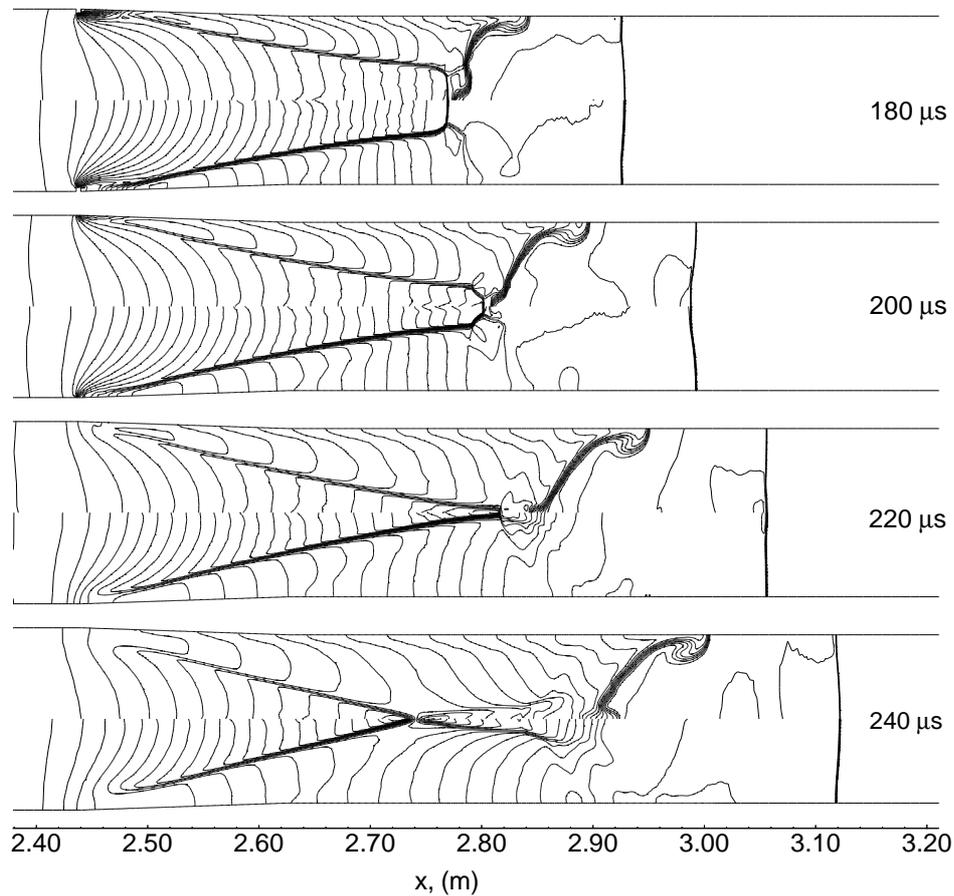


Figure 3.2: Density contours (top half of frame) and pressure contours (bottom half) for the flow evolution downstream of the diaphragm in the Langley expansion tube [43].

The implementation of a model that correctly incorporates aspects of this highly non-uniform phenomenon should produce improved results for shots in which the driver gas expands to a higher Mach number. For this reason shot 3215 is used as a test case for potential diaphragm-rupture models. The analysis is simplified by considering only the compression tube and the shock tube. Piston dynamics are neglected and the compression tube is modelled as being 70 cm long and filled with gas at a pressure slightly lower than the diaphragm rupture pressure; the corresponding temperature is calculated assuming an isentropic compression from initial fill conditions. A fixed wall is placed at the end of the 10 m compression tube corresponding to the nozzle throat.

3.2 A Quasi-One-Dimensional Simulation Code: E1d

E1d is a quasi-one-dimensional Eulerian code that was written for the purpose of experimenting with non-isentropic process models at the diaphragm station. Like its Lagrangian counterpart L1d [26] from which it was derived, E1d allows for the modelling of multiple regions of gas, separated by diaphragms. Coupling between adjacent regions is achieved by setting appropriate boundary conditions depending on the state of the adjoining diaphragm (intact, partly ruptured or fully ruptured).

The governing equations applied to each region are the quasi-one dimensional conservation equations for mass, momentum, energy and species fraction as discussed in Chapter 2. In semi-discretised form they can be written

$$\frac{d \langle \mathbf{U} \rangle}{dt} = \langle \mathbf{Q} \rangle - \frac{1}{\delta x} (\mathbf{F}_R - \mathbf{F}_L) \quad (3.1)$$

where,

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho E \\ \rho f_{is} \end{bmatrix} \quad (3.2)$$

is the algebraic vector of conserved quantities,

$$\mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho E u + P u \\ \rho f_{is} u \end{bmatrix} \quad (3.3)$$

is the inviscid flux vector and

$$\mathbf{Q} = \begin{bmatrix} 0 \\ \overline{P}\delta A - \overline{F}_{wall} - \overline{F}_{loss} \\ \overline{q} \\ 0 \end{bmatrix} \quad (3.4)$$

represents additional sources of mass, momentum, energy and species density. These terms incorporate the component of momentum change due to the quasi-one-dimensional formulation as well as empirical correlations for wall friction \overline{F}_{wall} , pipe-fitting losses \overline{F}_{loss} and heat transfer effects \overline{q} [26].

3.3 E1d Test Cases

The ideas presented in the previous chapter describe numerical techniques that can be applied to solve the quasi-one dimensional Euler equations. Combining these techniques into a fully functioning CFD code provides the potential for a powerful computational tool; however, we must first verify that it performs as expected. In order to test that the numerical simulation solves the discretised form of the governing equations accurately, it is customary to perform several simulations on problems for which there exists either an analytical solution or proven data.

3.3.1 The Ideal Shock Tube problem

The first series of test cases to be considered concern the ideal shock tube problem (or Riemann problem) in which two gas states initially separated by a diaphragm are allowed to interact. The ideal shock tube with pressure ratio of 10 across the diaphragm is a commonly used benchmark problem for compressible CFD codes [55]. The diaphragm is assumed to rupture instantaneously allowing the high-pressure gas to expand and drive a shock wave into the low pressure gas. An expansion wave propagates upstream into the high pressure driving gas. If viscous effects along the tube walls are ignored the problem can be modelled using the one-dimensional Euler equations.

For this simulation 100 computational cells were used to model Sod's [55] initial value

problem described by

$$x \leq 0.5 \text{ m} : \rho_L = 1.0 \text{ kg/m}^3, P_L = 10^5 \text{ Pa}, u_L = 0 \text{ m/s}$$

$$x > 0.5 \text{ m} : \rho_R = 0.125 \text{ kg/m}^3, P_R = 10^4 \text{ Pa}, u_R = 0 \text{ m/s}$$

The diaphragm is removed at $t=0$, and the one-dimensional Euler equations are integrated in time subject to the stability condition $\text{CFL} \leq 0.8$. Comparisons with the theoretical distributions for pressure, temperature, density and velocity were made for the flow state at $t=0.6 \text{ ms}$; it can be seen from Figure 3.3 that good agreement is found. The shock is resolved adequately over 3 computational cells. In contrast, the contact surface appears quite diffuse. These phenomena are standard characteristics of attempts [55] to model Sod's problem, and as such, the results are deemed satisfactory.

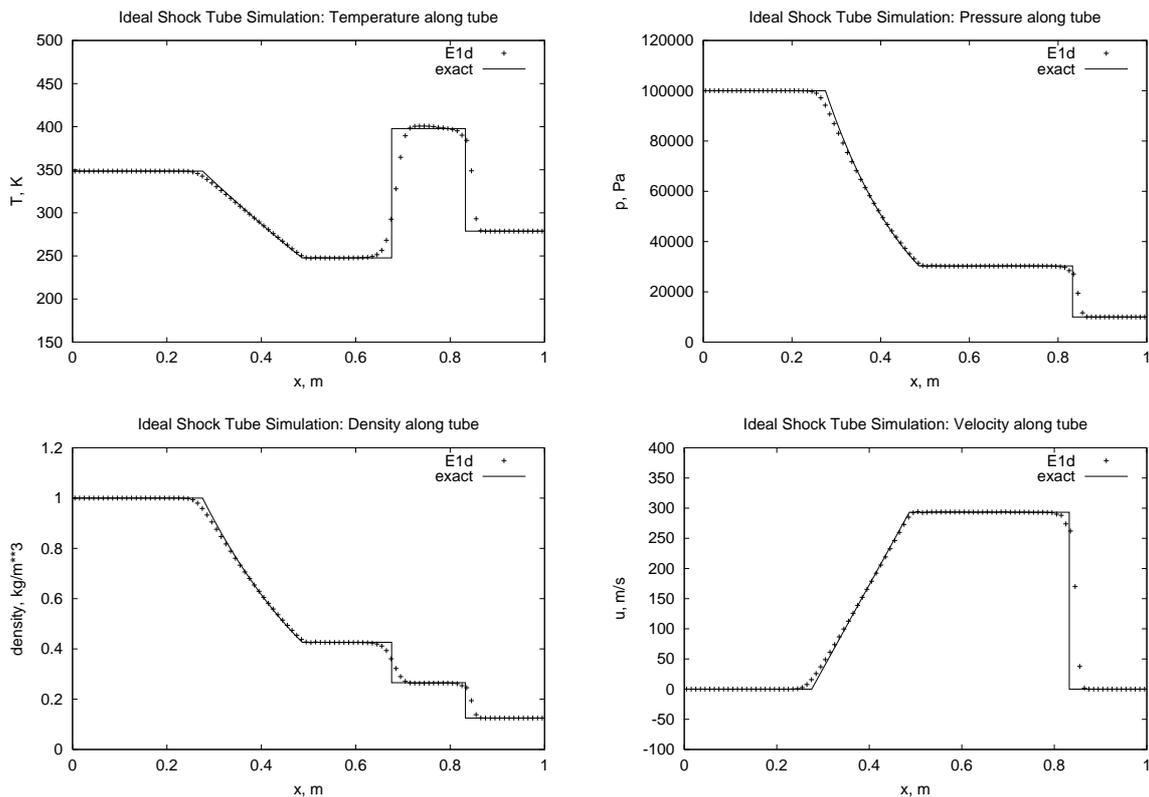


Figure 3.3: Comparison between numerical and analytical solution for Sod's shock tube problem [55].

3.3.2 Shock Wave Propagation through an Area Change

The final test case to be considered concerns the propagation of a shock wave through a relatively sudden increase in duct area. This provides a mechanism to validate the numerical implementation of the variable area term in the quasi-one-dimensional momentum equation. Considerable experimental data is available for this problem [49] as well as numerical results from other quasi-one-dimensional codes [27]. The discretised flow domain comprises 200 cells representing a 4 metre long duct with a smooth area change from 0.5 m^2 to 1.0 m^2 over a transitional length of 0.8 m. Initially, a Mach 2 shock wave is established in the tube through the implementation of a state specified by the Hugoniot jump conditions:

$$x \leq -0.5 \text{ m} : \rho_L = 0.41662 \text{ kg/m}^3, P_L = 71250.0 \text{ Pa}, u_L = 585.62 \text{ m/s}$$

$$x > 0.5 \text{ m} : \rho_R = 0.125 \text{ kg/m}^3, P_R = 10^4 \text{ Pa}, u_R = 0 \text{ m/s} .$$

Supersonic inflow boundary conditions are implemented on the left of the domain by specifying post shock flow conditions to the adjacent ghost cells. In a similar manner, extrapolation is used to impose the outflow condition on the right of the domain. The quasi-one-dimensional Euler equations are integrated in time using a CFL limit of 0.8. A space-time plot showing the important flow features is shown in Figure 3.4. The high speed flow expands through the divergent section of the duct lowering its pressure. An upstream-facing shock processes this gas so as to increase its pressure to equal that behind the primary shock wave. A plot of Mach number for this scenario is shown in Figure 3.5. From the good agreement found with the experimental work of Salas [49], it can be concluded that the numerical algorithm is correctly implementing the varying area terms.

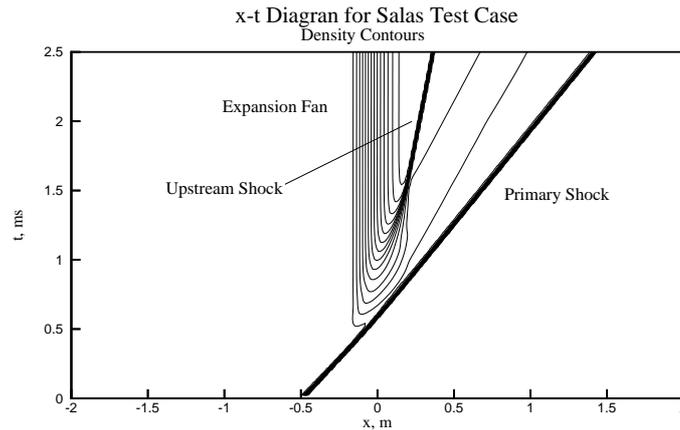


Figure 3.4: Space-time plot for the Salas area change problem.

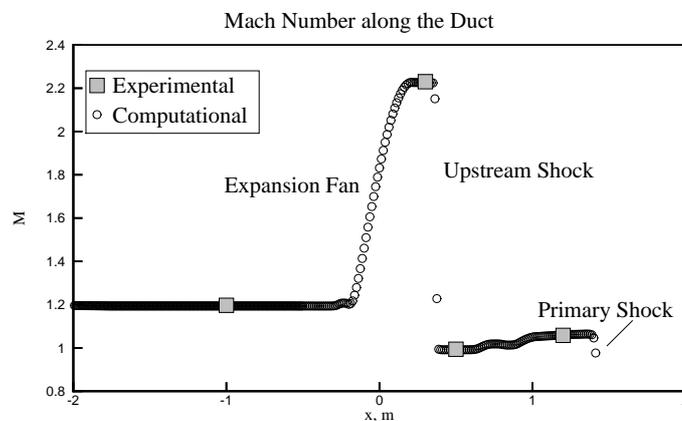


Figure 3.5: Mach number as a function of distance at $t=2.5$ ms for the Salas area change problem; comparison with experimental data [49].

3.4 Modelling Diaphragm Rupture

The highly three-dimensional flow that develops downstream of a real diaphragm rupturing in a non-ideal manner cannot be captured by equations that allow variations in only a single direction. Instead it is hoped that the macroscopic effect of this complex phenomena can be approximated and incorporated into the one-dimensional equations by means of a ‘correction’ to the ideal flow state. This section looks at two models that aim to modify the properties of the gas passing through the diaphragm station so as to be representative of the real mixed out flow state far enough downstream where effects are essentially one-dimensional again.

In order to develop a model for the effects of the two-dimensional mixing processes taking place, a quasi-static mixing zone is considered (Figure 3.6). Gas enters the region

as choked flow at the diaphragm opening and is assumed to change instantaneously to some ‘mixed out’ average flow at the exit plane (or entry to the shock tube). Because the details of the process are unspecified, extra information is required to determine the mixed out state. We now consider two possibilities for closing the governing set of equations.

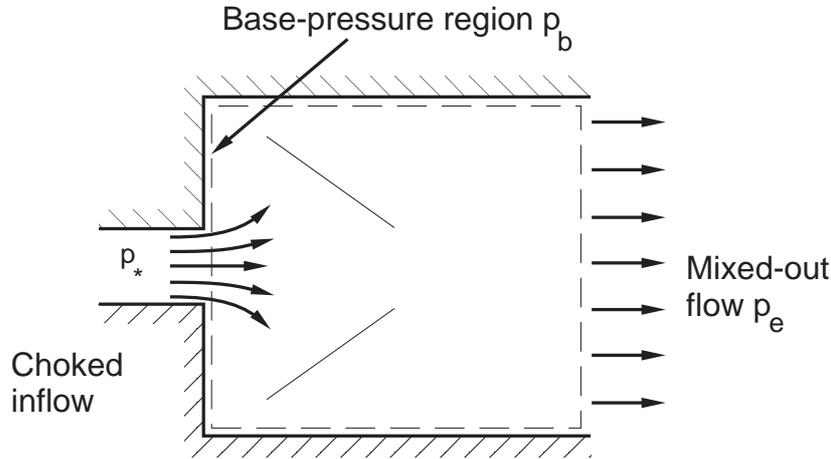


Figure 3.6: Schematic of the mixing-zone region.

3.4.1 The Nozzle-Shock Model

Whilst the full details of the interactions occurring within the mixing zone are unknown they can be assumed to comprise, in part, of a mixture of expansion and shock compression processes. A model centred around these observations can be constructed by first assuming supersonic isentropic flow from the diaphragm throat to the full area of the shock tube characterised by [14]

$$\frac{A_e}{A_*} = \sqrt{\frac{1}{M_1^2} \left(\frac{2}{\gamma + 1} \left(1 + \frac{\gamma - 1}{2} M_1^2 \right) \right)^{\frac{\gamma + 1}{\gamma - 1}}}. \quad (3.5)$$

Given the fraction of diaphragm ruptured so far, the Mach number at the completion of the expansion process, M_1 , can be determined numerically or approximately by interpolating polynomials. The complete flow state is then calculated from fundamental isentropic relations [14]. Finally the gas is processed by a normal shock to give the conditions at the exit state. This model is referred to as the ‘Nozzle-Shock’ model and is considered the strongest model since a normal shock presents the maximum entropy increase and corresponding loss of total pressure.

3.4.2 The Base Pressure Model

The second model considered assumes that the pressure on the downstream face of the unruptured part of the diaphragm is known (Figure 3.6). Although the bulk flow becomes supersonic, there exist patches of subsonic flow in the region behind the unruptured diaphragm. These regions provide a feedback path whereby the downstream mixed-out condition will influence the pressure seen on the downstream surface of the opening diaphragm (p_b). Because of the geometric similarity with flow about the base of a projectile, we will call this the ‘base pressure’ model. An expression giving the ratio of base pressure to the choked inflow pressure, for the analogous case of internal flow through an abrupt cross sectional area change, was derived by Korst [33]. We use a linear interpolant of this theory (taken from Figure 7 in [33]) applicable to conditions when the diaphragm is half open. For the very small orifice areas encountered at the beginning of diaphragm rupture, the value of the ratio is taken to be $p_b/p_* = 0.13$. The governing mass, momentum and energy equations can thus be written.

$$\rho_e u_e = \rho_* u_* \frac{A_*}{A_e} = \phi_m \quad (3.6)$$

$$p_e + \rho_e u_e^2 = p_* \frac{A_*}{A_e} + p_b \left(1 - \frac{A_*}{A_e}\right) + \rho_* u_*^2 \frac{A_*}{A_e} = \phi_{mom} \quad (3.7)$$

$$\left(\frac{\gamma}{\gamma - 1}\right) u_e p_e + \frac{1}{2} \rho_e u_e^3 = \rho_* u_* h_* \frac{A_*}{A_e} = \phi_{en} . \quad (3.8)$$

These can be solved to give a quadratic equation expressing the mixed-out velocity.

$$\left(\frac{\gamma + 1}{2} \phi_m\right) u_e^2 - (\gamma \phi_m) u_e + (\gamma - 1) \phi_{en} = 0 .$$

Knowing this velocity, the downstream flow state can be obtained from the definitions for the mass, momentum and energy fluxes $\phi_m, \phi_{mom}, \phi_{en}$ in equations (5)-(7).

3.4.3 Model Implementation

Both of the above mentioned models require a function governing the diaphragm's opening behaviour. This is required to specify the fraction of diaphragm ($\frac{A^*}{A_1}$) that is open at any particular time (starting with a fully closed diaphragm at $t=0$). The linearly varying relationships used by Petrie [43] is employed; however, we introduce the concept of a diaphragm influence time.

Justification for modifying the period over which gradual diaphragm rupture influences the flow comes from the computations of Petrie [43] shown in Figure 3.2. The oblique shock waves remain in the shock tube well after the diaphragm has fully opened and hence continue to process the driver gas. This is modelled by (slowly) opening the diaphragm over a significantly longer period than the nominal $500 \mu s$ rupture time; this longer period is the "diaphragm influence time".

In each of the two models presented, the downstream flow state is assigned as boundary conditions to the region downstream of the rupturing diaphragm. In this manner, the details of the mixing process are contained only in the calculation of appropriate boundary fluxes. These fluxes are used as the outflow boundary condition for the upstream region in order to ensure conservation of mass, momentum and energy overall.

3.4.4 Results

The results for the 'base-pressure' model are shown in Figure 3.7 for a variety of diaphragm influence times. It can be seen that with no diaphragm model included, E1d also computes the pressure jump just after $t=222$ ms indicative of over-tailored operation (cf. Figure 3.1). It can be seen that application of the diaphragm rupture model makes no appreciable difference. Grid independence was investigated by performing simulations at several different resolutions; no qualitative differences in the results were found.

In contrast, results for the 'nozzle-shock' model are shown in Figure 3.8 for a variety of diaphragm influence times. Application of the diaphragm rupturing model can be seen to decrease the over-tailored pressure jump. With an opening time of 1ms, the strength of the re-reflected shock is reduced significantly; however, it still appears that the full nature of the phenomenon is not captured. The simulated pressure trace rises above the experimental trace in the latter stages; however, this may be attributed partly to the fact

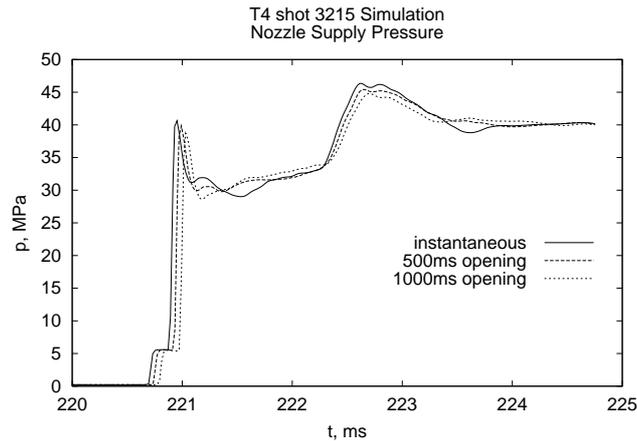


Figure 3.7: Simulated Nozzle Supply Pressure for T4 shot 3215 using the base pressure model with varying diaphragm opening times.

that no effort was made to model piston dynamics and the associated expansion waves that eventually process the test gas.

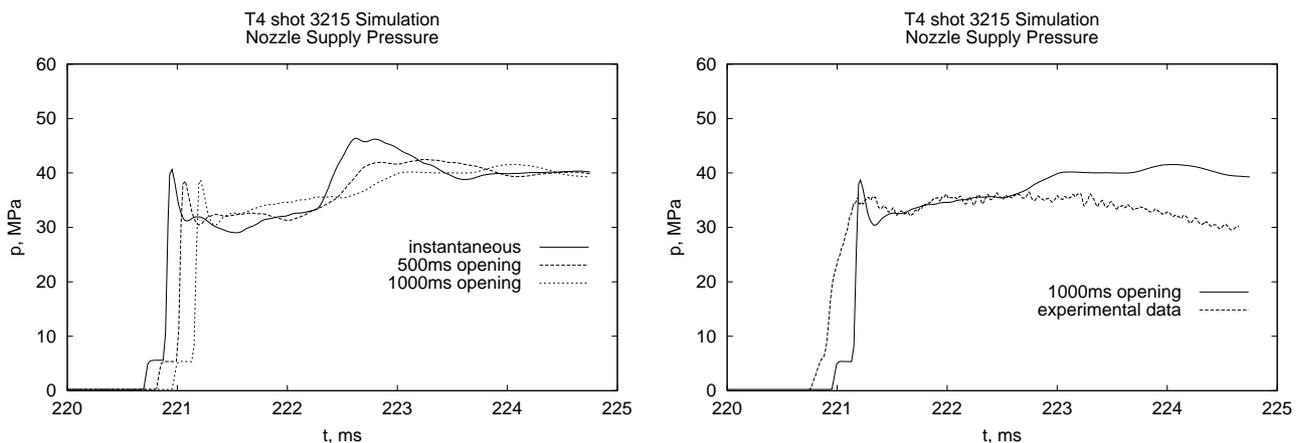


Figure 3.8: Nozzle Supply Pressure for T4 shot 3215 using the nozzle shock model with varying diaphragm opening times (left). Comparison between 1 ms diaphragm influence model and experimental data (right).

Although Petrie's inviscid simulation suggests that all oblique shock waves would have 'washed' downstream and weakened considerably 1 ms after the commencement of diaphragm rupture, we would expect that the boundary layers present on the real tube walls to slow this process. This would allow the oblique shocks to process the driver gas for a relatively long time; however, we do not have experimental data on the near-diaphragm flow processes for T4. The current arrangement does not permit investigation of the existence and duration of oblique shock waves within the shock tube.

3.5 Shock Tunnel with Driver Restrictor

Whilst the previous simulations offer encouraging results, it is clear that the full nature of the slowly opening diaphragm is not captured by either of the models. By altering the diaphragm influence time, the driver gas total pressure can be changed; however, a significant departure from experimental traces remains. In order to study the situation that would exist without the transient effects of the rupturing diaphragm, simulations were performed on a shock tube arrangement in which an orifice plate placed at the driver/driven interface restricts the flow of driver gas. It is presumed that the flow structure downstream of the restrictor consists of oblique shocks similar to those present during diaphragm rupture. This allows the 'nozzle-shock' and 'base-pressure' models to be compared for a constant area ratio at the diaphragm throat.

The experimental work of Roberts *et. al.* [47] provides the benchmark for comparison. They showed that by using the restrictor, a slower shock speed could be produced for the same diaphragm pressure ratio. This allowed them to produce a high specific enthalpy but low pressure, test gas reservoir. The experimental set-up comprised a uniform bore shock tube 127 mm in diameter. The driven section was 9.6 m in length and filled with hydrogen at a pressure of 5 Torr. Shock speeds were determined for a variety of diaphragm pressure ratios using helium as the driver gas. Results with and without the 27 mm diameter driver restrictor (producing an area ratio of 25) were compared. Because viscous losses cause the shock speed to drop off along the length of the tube, a range within which the shock speed lies is presented.

Computational simulations were performed using 800 cells to model the 9.6 m driven section, and 200 cells to model 5 m of the driver section. A constant cross sectional area was used, the restriction being simulated with the 'nozzle-shock' model described previously. Diaphragm pressure ratios of 3000 and 1000 were chosen, as these represented the upper and lower limits of the experimental investigation.

A comparison between numerical and experimental shock speeds both with ($A/A_* = 1$) and without ($A/A_* = 25$) the driver restriction is shown in Table 3.1. Computational shock speeds were calculated by comparing simulation pressure traces at two locations near the end of the driven section. Grid convergence was obtained to about 2%. It can be seen that without the restrictor the computational simulation produces shock speeds that

are slightly higher than the maximum shock speed encountered during the experiments. This maximum shock speed would be encountered at the beginning of the driven section though, viscous losses causing shock speed to drop off along the duct. Hence, at the end of the duct, where the computational shock speeds were calculated, the discrepancy would be larger. Also apparent is that whilst the implementation of the ‘nozzle-shock’ model produces a slower shock speed at both diaphragm pressure ratios, the decrease falls well short of that observed experimentally. Due to the limitations of the ‘nozzle-shock’ model, no attempt was made to implement the ‘base-pressure’ model which has already been shown to influence the flow to a lesser extent.

	$P_4/P_1 = 1000$		$P_4/P_1 = 3000$	
	$A/A_* = 1$	$A/A_* = 25$	$A/A_* = 1$	$A/A_* = 25$
experimental	2050-2300 m/s	1200-1300 m/s	2500-2650 m/s	1600-1800 m/s
computational	2333 m/s	2041 m/s	2769 m/s	2482 m/s

Table 3.1: Comparison between computational and experimental shock speeds for a uniform bore shock tube, with and without a driver restrictor

These results are consistent with the earlier simulations of T4 shot 3215. Whilst the model appears to influence the downstream flow properties to some extent, it does not capture the full nature of the phenomenon. It is uncertain what needs to be incorporated into the model to produce a stronger effect than that produced by the ‘nozzle-shock’ model. A more detailed analysis of the wave pattern downstream of the driver restrictor might give greater insight.

Multi-Dimensional Flow Modelling

Whilst there are instances where the quasi-one dimensional flow assumption discussed in the previous chapter is a valid approximation, many flow features are inherently multi-dimensional. Oblique shock waves and vortical structures are two such features that we tried to compensate for by the introduction of models based on some macroscopic knowledge of the flow. If a detailed study of these flow features is to be made however, greater complexity must be introduced to the set of governing equations through the introduction of additional spatial coordinates. This chapter looks at computational techniques for efficiently solving the two-dimensional Euler equations. It must be stressed that this in turn is still a simplistic model since vorticity and shock waves are three dimensional phenomenon.

4.1 Computational Modelling

The computational simulation of problems involving complex interactions requires a fine numerical grid in order to adequately resolve the underlying processes. As we refine the resolution, both spatially and temporally, more processor time is needed to solve the resulting discretised equations. One approach is simply to accept the long computational time needed to obtain a solution. The time frames associated with such a ‘brute-force’ technique quickly become too large however. Additionally, advances in computer technology quickly makes old hardware obsolete; it is not unreasonable to expect a doubling in memory, and speed every 1-2 years [31].

Another possibility is to run simulations using multiple processors. Termed parallel computing, this approach has reached a level of maturity [17] and can significantly reduce physical solution time in hydrodynamic [30] and other computationally intensive

problems. Producing parallel code is not a straight forward task however and often requires re-formulation to be applied on a different system. Since code-parallelisation involves significant effort tailoring the program to the system it is sensible to first ensure the efficiency of the sequential algorithm.

Hyperbolic partial differential equations, such as the Euler equations, describe flows which inherently contain sub-regions of greater physical complexity. Shocks and contact surfaces are two such features that result in significantly higher gradients of flow properties than arise from the weak acoustic waves propagating in the mean flow. A result of this is that as the numerical grid is refined in order to resolve highly detailed aspects of the flow, the same refinement is applied in regions exhibiting a relatively uniform state. A large part of the total computational effort associated with solving hydrodynamic problems is in the computation of fluxes between cells. Using an approximate Riemann solver reduces the number of computations that must be performed, however, this function and the associated reconstruction is still expensive and should be used sparingly. This is the motivation behind adaptive algorithms that aim to cluster grid points around regions of high flow gradients, thus providing high resolution only where it is needed. Other regions in the flow domain are more coarsely discretised and so contribute less to the overall computational cost of the simulation. The characteristic smearing of shocks and contact surfaces over several cells still occurs; however, the increased cell concentration effectively reduces the physical thickness. Such a dynamically adaptive grid can be constructed in several ways.

4.1.1 Discretisation and Solution Adaptive Techniques

Early effort was directed at schemes that use a fixed number of grid points. The mesh is then stretched in some regions and compacted in others as the flow field develops [15, 3, 6]. In addition to clustering cells around shock discontinuities this also has the advantage of aligning cell boundaries with important flow features where Godunov type schemes perform extremely well.

A more conceptually simpler approach is to simply add more cells in regions where they are required. Patch methods achieve this in a relatively efficient manner by layering meshes of increased refinement over existing coarse sub-grids at strategic locations [2,

45]. This has the added advantage that a temporally adaptive integration scheme can be easily implemented since each block can be handled almost independently of all others. Meshes can even be layered at different angles of rotation relative to the coarse grid [1]. This allows more freedom over regions to be refined and also permits meshes that align with relevant flow features; however, increased complexity results. Unfortunately, the extension of patch methods to truly three dimensional flow is relatively unexplored and in my opinion would be too restrictive for anything but simple flow fields.

A more flexible approach, albeit more memory intensive, involves the sub-division of existing cells to form new cells in a hierarchical manner. Unstructured meshes comprising triangles [43] or polygons assembled from bisecting triangles covering the domain [65] have been used successfully to model complex phenomena in this way. This is a very versatile approach and is especially useful when complex geometries are involved. In the latter, interfaces are always midway between cell centres allowing uniform mesh reconstruction techniques to be applied. However, our main goal is to produce a very detailed analysis of interacting shock vortex phenomena on a very simple geometry. Under such circumstances it can become counter productive to use an unstructured mesh [45] particularly since temporal adaptivity appears rarely to be implemented and quite difficult under such schemes.

In contrast rectangular cells can be organised into a tree structure allowing geometric links to be made in a relatively straightforward manner [30, 51]. Such schemes have been developed that incorporate a temporally adaptive strategy [4] allowing explicit time marching and the benefits of increased transient resolution. For simulation of flows that involve several complicated yet isolated flow features the overheads of tree-data storage seem justified by its flexibility and relative simplicity. For this reason, we adopt the hierarchical sub-division of square cells as the adaptive strategy for MACS2d.

4.1.2 Data Storage

Tree-algorithms provide an intuitive yet powerful mechanism for organising data storage for adaptive mesh algorithms. Traditional quad-tree data storage applied to two dimensional adaptive grids involve linking every cell to the 4 cells below it (children cells) and the cell from which it originated (the parent cell) [7]. At the top of the tree is the root

cell which has no parent (Figure 4.1). In addition to this the level of the cell in the tree is stored, for a total of 5 words per cell. By storing all the necessary pointers in this manner, it is possible to locate any data, at any level of the tree by traversing the tree until a common branch is found. The downside to this method however, is that whilst all information is equally available, the process of recovering relevant data such as neighbour cell state information may demand excessive tree traversals; in the worst case scenario, the tree must be retraced to its root. However, this general access to all points of the flow domain is wasteful when applied to hydrodynamic problems in which information propagates at a finite speed. Regions of the flow domain remain unaffected by other regions bounded by lines termed characteristics. Explicit time integration of the Euler equations exploits this phenomenon through the implementation of a CFL criterion to ensure that wave interaction is felt at most one cell away. Since the numerical technique requires information from neighbouring cells only (and neighbours of neighbours if solution reconstruction is employed), it is wasteful to use a generalised tree algorithm that doesn't discriminate between neighbouring cells, and cells on the other side of the flow domain.

Threaded tree algorithms address this matter. They store the same pointers as for the tree algorithm presented above but in addition pointers to neighbour cells are kept, making a total of 9 words per cell. Neighbours are constrained so as to differ by one level of refinement; however, the links between them are not reciprocal as shown in Figure 4.1. A cell's neighbour is either at the same level of refinement or 1 level lower. By storing these extra links it is possible to travel across the tree without retracing it to the root cell. Unfortunately, these direct neighbour links must be re-targeted whenever the grid is modified. This causes problems for parallelisation, particularly if neighbouring cells are to be removed simultaneously.

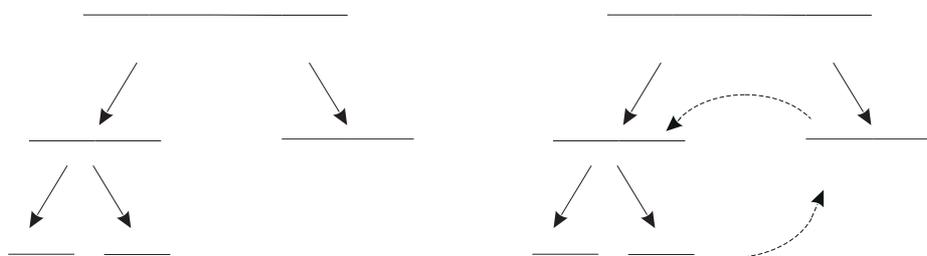


Figure 4.1: Pointer links for tree algorithms (left) and threaded tree algorithms (right).

A more efficient tree-algorithm can be designed subject to the following three obser-

vations regarding cell-neighbour relations [32].

- During the refinement procedure a cell is split into four children cells. Since they are created simultaneously they can be stored contiguously in memory, allowing a single pointer to reference all four children.
- the neighbour relations between these four children cells are known explicitly.
- Neighbours that aren't known explicitly are children of neighbouring parent cells. Hence they can be determined without search by accessing the neighbouring parent.

By organising cells into groups of four called quads, we exploit the explicit relation between children cells of a common parent. The quad contains data storage for the four children cells in a prescribed manner as shown in Figure 4.2, as well as storing a pointer to the parent cell from which it was derived. Additionally, pointers to the parent cells of neighbouring quads are stored (in the order shown in Figure 4.2). As with conventional threaded algorithms this provides fast neighbour retrieval; however, rather than links being between cells, the links are effectively made one level higher in the tree. By referencing every link to cells higher up the hierarchy in this manner, they will not require later modification. Finally, the refinement level of the quad is stored as are the coordinates of the parent cell centroid.

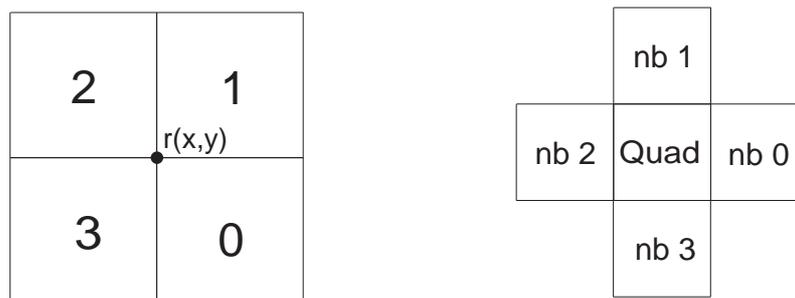


Figure 4.2: The four children are stored in a prescribed manner in a quad (left). Each quad has a link to the parent cells of four neighbouring quads (right)

Each cell requires only one pointer, linking it to the quad containing its children cells if any, or a null pointer. State information as well as primitive variables are also stored for each cell.

The basic C-data-structures for cells and quads used to implement this efficient, threaded algorithm are shown below to emphasize the inter-dependency between the two.

```

struct Cell {
    struct Oct *OctCh;      /* pointer Oct containing children cells */
    double U[4];           /* state vector */
    struct flowstate state; /* the flow state P, rho, e, u, v, a */
};

struct Oct {
    struct Cell cells[4];  /* the four comprising the Oct */
    struct Cell *Octpr;   /* pointer to the parent cell */
    int OctLv;            /* the level of the Oct */
    struct Cell *OctNb[4]; /* pointers to the parent cells of
                           neighbouring Octs */
    double r[2];          /* location of centroid x,y,(z) */
};

```

The resulting linked structure requires 10 words of memory per quad, or 2.5 words per cell. This represents a doubling of storage efficiency over traditional methods whilst incorporating improved threaded links to neighbour cells. As well as this, modification of a cell is independent of any other cell as shall be discussed in section 4.1.5.

4.1.3 Implementation of Data Storage

The quad-tree data storage scheme described previously allows square domains to be discretised in a straight forward manner; the root cell is the computational domain itself! One obvious problem with this technique however, is that arbitrary Cartesian flow grids cannot be generated as all cells cannot be linked back to a common root. In addition to this, the memory storage required to keep quad information at all levels quickly becomes excessive, particularly since computations are rarely performed on anything but the finest half-dozen or so grids. Both of these problems can be alleviated by realising that all neighbour information can be obtained without propagating any higher than the quad level in the tree (Figure 4.3).

Arbitrary Cartesian grids can be generated by creating an i-j ordered array of square quads that cover the domain. The necessary links are obtained by creating pseudo cells that contain these quads; the cells perform no purpose other than to link neighbouring quads. This mesh represents the coarsest discretised solution domain on which computations can be performed.

Whilst non-adaptive algorithms use fixed length arrays to store all data, the dynamic modifications to adaptive grids require memory to be constantly re-allocated. By organising all cell information in a quad-tree as previously discussed, it is possible to manage

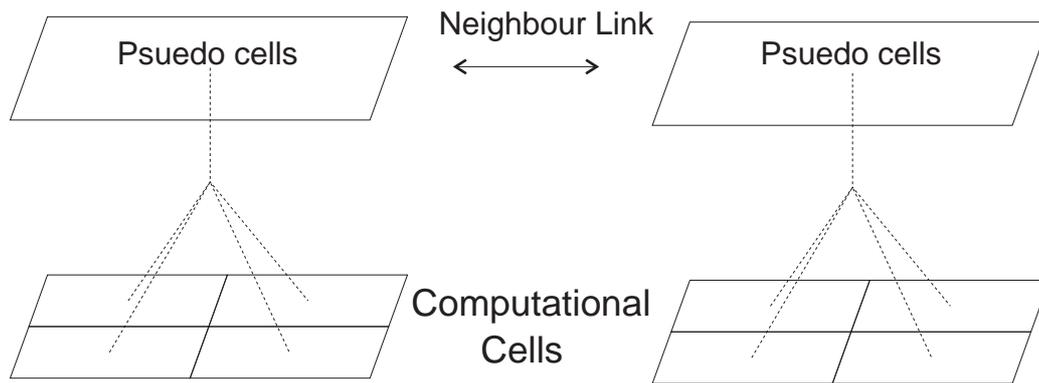


Figure 4.3: Since quads are linked to the parent cells containing neighbouring quads arbitrary Cartesian grids can be generated by creating pseudo cells. These cells are the parents of cells in the coarsest computational mesh and serve the purpose of linking the structure together.

data storage in a structure similar to that employed in Object Oriented programming. Whenever a new quad is required, memory is assigned dynamically, and a pointer to its location created. All such pointers are stored in a fixed length vector whose length is specified in the parameter file at the beginning of a simulation. Upon removal, the memory allocated to a quad is made free, and its pointer destroyed. Once all such modifications to the pointer array have been made it is compressed to remove all NULL pointers. A single reference to this array is all that is ever required to access all cell data and the associated links.

4.1.4 Accessing Neighbour Data

During the solution procedure the need to access neighbour data frequently arises. Whilst the threaded algorithm contains links to neighbouring quads, it is important to develop a universal function for locating neighbouring cells. Depending on the mesh configuration the immediate neighbour will differ by up to one level of refinement. This means that there are 3 possibilities that must be considered in the following order.

- The cell referenced in the neighbour link is a leaf. This is the case when a fine cell interfaces a coarse cell. Under this scenario the cell and its level are returned.
- The cell referenced in the neighbour link has children. This time the child cell interfacing with the cell under consideration is located. If the child cell is a leaf then the interface is between two cells at the same level. The cell and level are returned by the function.

- If however, the child cell is itself a parent, the interface is between a coarse cell and two at a finer level. Rather than returning the two interfacing cells, it is sufficient to return the parent, and an indication that it contains children.

Given the cells position within its quad (as shown in Figure 4.2a), and the neighbour required (according to Figure 4.2b), the function *LookUpNb* returns a pointer to the neighbour cell and the refinement level of the neighbouring leaf according to these criteria. Look-up tables based on these parameters are used to ensure efficiency by avoiding complex boolean statements. As shall be seen in the subsequent sections this function allows the retrieval of all necessary information from the threaded-quad-tree data structure.

4.1.5 Refinement and Coarsening Procedure

An integral part of any adaptive algorithm is the ability to cluster points around regions of large flow gradients whilst covering the mean flow with a relatively coarse mesh. This reduces the computational cost of a simulation since it aims to produce a uniform discretisation error across the domain. During the simulation of complex gas dynamic processes it is envisaged that regions of the computational domain will oscillate between high and low levels of refinement as flow features pass through it. For this reason it is important that the refinement and coarsening procedures be as simple as possible. The quad-tree structure described previously allows for the implementation of both these tasks in a relatively straightforward manner.

Adaption Criteria

The first step in dynamically modifying a computational mesh is to determine a measure of solution error across the domain. Cells are removed where it is deemed that a coarser discretisation can adequately describe the flow, and added in regions where the solution error is unacceptably large, in an attempt to produce the optimal distribution of computational resources. There are several ways that computational cells can be considered for modification.

One technique is to formulate indicators that aim to distinguish flow features that are difficult to resolve. Shock waves, contact surfaces and expansion fans are three features that can be distinguished from one another by monitoring pressure, density and velocity

variations. Gradient indicators have been used to detect these features and refine the mesh accordingly [60]. They generally monitor the relative change in a prescribed variable in each of the coordinate directions [32]. With reference to Figure 4.5, a gradient indicator can be defined at the interface between two cells as

$$\epsilon_{grad} = \frac{|f(x)_{nb} - f(x)_{cell}|}{\max(f(x)_{nb}, f(x)_{cell})}. \quad (4.1)$$

Shock waves are identified in regions where pressure variations are large and velocity gradients negative, whilst high density, but no pressure gradient, indicates a contact surface. Cells in which any one of these features are identified are marked for refinement. Conversely cells where the indicators are below a preset value are marked for deletion.

Another method that can potentially produce optimal configurations, is to monitor the truncation error in the discretised equations directly. This can be done by calculating a solution on both a fine and a coarse grid [2] and comparing the difference; however, this is computationally expensive. Alternatively, an approximate expression for the truncated terms can be formulated by considering the computation of first order derivatives as required for the evaluation of fluxes. An expression for the derivative is obtained from a one-dimensional Taylor series expansion which can be rearranged to give

$$f'(x) = \frac{f(x + \Delta x) - f(x)}{\Delta x} - \frac{\Delta x}{2} f'' + \dots \quad (4.2)$$

the leading error term being associated with the second spatial derivative. The relative truncation error is thus

$$\epsilon = \frac{\frac{\Delta x}{2} f''(x)}{f'(x)} \quad (4.3)$$

The calculation of this error indicator for each cell requires a finite-difference approximation for the ratio of second to first order derivatives [57]. On the non-uniform grid shown in Figure 4.4 the cell centred second derivative can be approximated by the difference

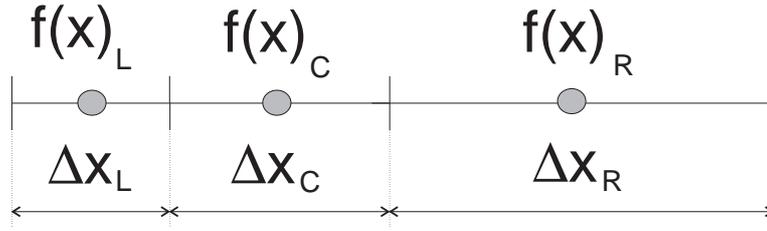


Figure 4.4: Cell nomenclature for non-equispaced cells.

equation

$$\begin{aligned}
 f''(x)_c &= \frac{1}{\Delta x_c} \left[\left. \frac{df(x)}{dx} \right|_R - \left. \frac{df(x)}{dx} \right|_L \right] \\
 &= \frac{1}{\Delta x_c} \left[\frac{f(x)_r - f(x)_c}{\frac{1}{2}(\Delta x_r + \Delta x_c)} - \frac{f(x)_c - f(x)_l}{\frac{1}{2}(\Delta x_c + \Delta x_l)} \right]
 \end{aligned} \tag{4.4}$$

whilst the first derivative is approximated by

$$\begin{aligned}
 f'(x)_c &= \frac{1}{2} \left[\left. \frac{df(x)}{dx} \right|_R + \left. \frac{df(x)}{dx} \right|_L \right] \\
 &= \frac{1}{2} \left[\frac{f(x)_r - f(x)_c}{\frac{1}{2}(\Delta x_r + \Delta x_c)} - \frac{f(x)_c - f(x)_l}{\frac{1}{2}(\Delta x_c + \Delta x_l)} \right]
 \end{aligned} \tag{4.5}$$

since the size of neighbouring cells at differing refinement levels differ by a factor of two, the dimensions x_l and x_r can be expressed as a function of x_c by introducing α_l and α_r such that

$$x_l = \alpha_l x_c, \quad x_r = \alpha_r x_c. \tag{4.6}$$

Hence the expressions for the first and second derivatives can be substituted into equation (4.3) and manipulated to give the error indicator in the form

$$\epsilon = \frac{|f(x)_r(c_r) + f(x)_l(c_l) - f(x)_c(c_r + c_l)|}{\eta f(x)_{ref} + |f(x)_r(c_r) - f(x)_l(c_l) + f(x)_c(c_l - c_r)|} \tag{4.7}$$

where,

$$c_l = \frac{1}{1 + \alpha_l}, \quad c_r = \frac{1}{1 + \alpha_r} \tag{4.8}$$

The expression $\eta f(x)_{ref}$ has been added to the denominator to avoid division by zero. The parameter η can be used to control the sensitivity of the indicator (to noise) by altering

the fraction of the reference value $f(x)_{ref}$ used. The local cell value is a good choice for the reference value, particularly for flows with large variations in flow properties, since it allows a uniform comparison irrespective of the absolute value. In some instances, a prior knowledge of the flow domain allows a more ‘appropriate’ value to be selected as shall be discussed in section 4.3.

High values of the indicator will be calculated in regions exhibiting a large curvature (second derivative which is representative of the truncation error) of flow properties where the solution technique generally performs badly. The adaption criteria in MACS2d uses the above curvature indicator applied in both coordinate directions to the density field.

$$\begin{aligned}\epsilon_i &= \frac{|c_{i+1}\rho_{i+1,j} + c_{i-1}\rho_{i-1,j} - (c_{i+1} + c_{i-1})\rho_{i,j}|}{\eta \rho_{ref} + |c_{i+1}\rho_{i+1,j} - c_{i-1}\rho_{i-1,j} + (c_{i-1} - c_{i+1})\rho_{i,j}|} \\ \epsilon_j &= \frac{|c_{j+1}\rho_{i,j+1} + c_{j-1}\rho_{i,j-1} - (c_{j+1} + c_{j-1})\rho_{i,j}|}{\eta \rho_{ref} + |c_{j+1}\rho_{i,j-1} - c_{j-1}\rho_{i,j+1} + (c_{j-1} - c_{j+1})\rho_{i,j}|}.\end{aligned}\quad (4.9)$$

The refinement indicator for the cell ϵ_{curv} is selected as the maximum of the indicators calculated in each of the coordinate directions. Cells in which ϵ_{curv} is greater than a predetermined refinement limit ϵ_r are marked as candidates for refinement whilst cells with ϵ_{curv} less than the coarsening limit ϵ_c are candidates for deletion. In addition to this a pressure gradient indicator ϵ_p calculated from equation 4.1 is used to identify shock waves. Cells in which ϵ_p is greater than a pre-set value characteristic of a shock wave, ϵ_s , across any of its interfaces are marked for immediate refinement to the finest level.

The calculation of refinement indicators requires knowledge of neighbour cell information. Considering only the interface between a cell being considered for adaption and one of its neighbours, there are 3 possible configurations as shown in Figure 4.5.

For neighbour cells at a refinement level less than or equal to that of the cell being considered, all relevant information can be obtained by the data access procedures discussed in section 4.1.4. When the neighbour is located at a finer level, however, it is more difficult to determine the refinement indicator uniquely. Whilst the optimum technique is to calculate two indicators and use the largest, it was found adequate to calculate a single indicator based simply on the properties of the neighbouring cell’s parent; the parent is at the same level as the cell being considered.

In order to eliminate the introduction of undesirable features into the computational

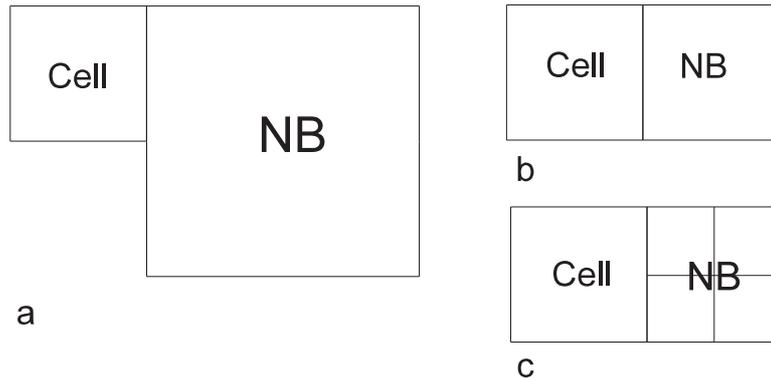


Figure 4.5: There are 3 possible neighbour configurations that must be considered during the calculation of refinement indicators. Cell NB is the neighbour used to evaluate indicators for the cell under consideration.

mesh (known as ‘mesh trashing’), grid modification requires several iterations to ensure that certain rules are obeyed. To maintain a smooth transition between coarse and fine regions, neighbouring quads are constrained so as to differ by no more than one level of refinement. In addition to this, pre-refinement is required ahead of moving shock waves. This ensures that the shock-wave doesn’t propagate into coarser regions causing spurious noise to be generated [57]. Wave speeds calculated by the flux solver can be used to predict how the shock will move [4] allowing cells to be refined in advance. Since the computational mesh in MACS2d is modified at every time step, however, it is sufficient to simply include a one cell ‘buffer zone’ around shock waves as shown in Figure 4.6.

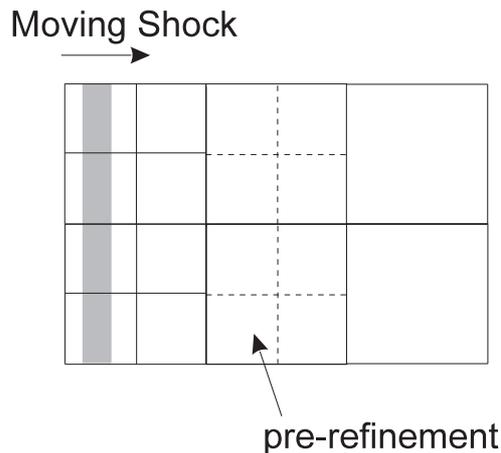


Figure 4.6: A one-cell buffer zone is included ahead of shock waves to prevent the wave from entering less refined regions (adapted from [57]).

If all four cells within a ‘quad’ are marked for deletion, the ‘quad’ can be removed provided the new configuration still satisfies the condition that neighbour cells differ by at most one level of refinement. Additionally, quads created during the previous mesh

adaption are not removed in an attempt to stop ‘border-line’ cells from oscillating between coarse and fine states. Only once a satisfactory configuration has been achieved are cells actually modified. The modification procedure consists of a series of refinement and coarsening procedures both of which can be carried out independently of other cells at the same level. This greatly reduces the amount and complexity of the code as well as allowing for later parallelisation. At the end of the mesh updating step, one pass through the quad pointer array is made to eliminate ‘holes’ due to deleted quad pointers.

Refinement

The refinement procedure involves splitting a cell to form a new quad. Memory is allocated dynamically for the new data and a pointer to its location inserted into the quad pointer array. Pointers to the parent cells of four neighbouring quads are located using the procedures defined in section 4.1.4. As can be seen from Figure 4.7, two of these cells are contained in the same quad as the new quad’s parent cells. The other two are simply children cells belonging to the neighbours of the quad containing the new parent. By enforcing the rule that neighbour cells differ by at most one level of refinement, it can be seen that regardless of the refinement configuration, the neighbour pointed to is always at the same level as the new parent cell (Figure 4.7). This emphasises the importance of linking quads to cells rather than standard cell-to-cell referencing [7]. Once a neighbour link has been made it requires no later modifications regardless of how the mesh may develop. Direct removal of the neighbour cell pointed to in the link violates the criterion that neighbour cells differ by at most one level of refinement; in other words, the quad would have to be removed before any adaption criterion could consider removing the cell referenced by the link. This also means that the refinement operation applied to one cell is completely independent of the refinement operations on other cells at the same level in the tree hierarchy. Finally the level of the new quad is updated, and a link made to its parent cell.

State information for the newly formed children cells must be determined. The simplest method is to copy the value from the parent cell directly to each child [10]. This is the most simple splitting strategy and was found to generate excessive noise particularly around shock waves. As more levels of refinement were allowed, the numerical oscillations resulted in code failure. This problem was resolved by assuming a variation of

properties within the new cell by interpolating values in neighbouring cells. Conservation of mass, momentum and total energy is of critical importance during this procedure and a significant amount of research has focused on achieving this during re-distribution of properties between complex meshes [9]. Due to the relatively simple geometric relations between square cells, sufficient accuracy can be obtained whilst guaranteeing conservation by calculating first order cell-centred increments in each of the two co-ordinate directions according to

$$\delta U_x = \frac{U_{i+1,j} - U_{i-1,j}}{8.0}, \quad \delta U_y = \frac{U_{i,j+1} - U_{i,j-1}}{8.0}, \quad (4.10)$$

where U represents the vector of conserved quantities (equation 2). Properties for each of the newly formed cells are then found by adding or subtracting the incremental values from the cell centred parent value. When determining neighbour properties in 4.10 the same cell configuration possibilities as in Figure (4.5) must be considered. Once again it was found adequate to use cell averaged values in the case of more refined neighbour cells (Figure 4.5c).

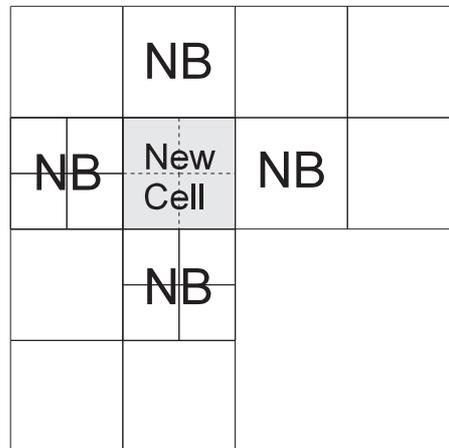


Figure 4.7: Since links are made from a quad to the parent cell containing the neighbouring quad, it can be seen that regardless of the configuration neighbours are always at the same level of refinement. This allows the refinement procedure on any cell to be independent of all other refinement procedures.

Coarsening

The only change made during the coarsening procedure involves destroying the link between parent cell and quad; the memory pointed to by the parent cell is freed, and the

pointer set to null. No other changes are required at cells or quads located at the same level of the tree. State information already exists for the parent cell as a volumetric average of the old children cells, as will be discussed later.

4.2 Solution Procedure

The governing equations are the 2 dimensional Euler equations which in semi-discrete form can be written as

$$\frac{\partial \mathbf{U}_{cell}}{\partial t} = \mathbf{Q}_{cell} - \frac{A}{V} \sum_{if} \mathbf{F} \quad (4.11)$$

A and V being the cell edge interface area and cell volume respectively. Their application to adaptive domains requires modification of the solution techniques presented in Chapter 2, a discussion of which is now presented.

4.2.1 Time Integration

On structured uniform Cartesian grids the integration procedure involves visiting every cell in order and updating the conserved properties based on fluxes calculated at the interfaces with 4 (6 in 3 dimensions) neighbour cells. When using adaptive grids however, any given cell can interface with up to 8 (24 in 3 dimensions) neighbouring cells and as such care must be taken to ensure all flux contributions are accounted for.

Additionally, the presence of cells that differ in volume by several orders of magnitude can result in excessively small time steps being taken in some regions. This introduces the possibility of temporal adaptivity as a means of further optimising the computational procedure [4]. At the beginning of a global timestep, all cells are considered in order to find the fastest wave speed. The time taken for a wave of this speed to traverse a cell at the finest level is determined and a corresponding maximum ‘base’ time-step calculated from the CFL criterion. This is then propagated to every level in the tree such that one global timestep comprises $2^{l-l_{min}}$ timesteps at a given level l in the tree hierarchy.

The improvements obtained from a temporally adaptive scheme are a result of a decrease in computational time spent on cells at the coarsest levels. These large cells have been selected during the spatial adaption step as containing regions of the flow domain

in which fluctuations are small. Hence only a small amount of the total computational effort is associated with computations on these cells anyway. For these reasons MACS2d currently implements spatial adaptivity only; however, the integration procedure is implemented in a way that allows for the later inclusion of a temporally adaptive scheme.

In the most simple case, the equations of motion can be advanced in time at each level of the tree hierarchy using an explicit first order scheme of the form described by

$$\Delta \mathbf{U}^{(1)} = \Delta t \frac{\partial \mathbf{U}^{(n)}}{\partial t}, \quad (4.12)$$

$$\mathbf{U}^{(n+1)} = \mathbf{U}^{(n)} + \Delta \mathbf{U}^{(1)}. \quad (4.13)$$

Incorporating the spatially split discretised form of the temporal derivative $\frac{\partial \mathbf{U}^{(n)}}{\partial t}$, for a cell at level l , this scheme can be written

$$\mathbf{U}_i^{t+\Delta t} = \mathbf{U}_i^t - \alpha \left(\sum_{if} \mathbf{F}_{if} \right) + \Delta t \mathbf{Q} \quad (4.14)$$

where α takes the form

$$\Delta t^{(l)} \frac{A}{V}. \quad (4.15)$$

Physically, this multiplication factor determines the change in the conserved intensive property due to a flux through an interface with area A over time Δt . Once all such contributions to the right hand side of equation (4.14) have been calculated, the state vector, \mathbf{U} , for the cell can be updated.

Rather than calculating all such contributions for a given cell at once, computations during the integration procedure are performed on an interface by interface means. For each interface, fluxes are evaluated and the neighbouring cells' state vectors are modified to account for their incremental effect. The fluxes are then discarded, and the procedure repeated for another interface. Since data is stored on a cell by cell basis, every cell at the integration level l is visited during this process. Each cell will either interface with another cell at the same refinement level l or with a cell at a refinement level one greater ($l + 1$) or one less ($l - 1$) than its own. Fluxes through interfaces between two cells at the integration level l are used to modify the state vectors for both cells. Since both these

cells will be visited during the integration procedure at the level l , care must be taken to prevent the flux being calculated through the same interface twice. This is achieved by only considering a cell's interfaces in the positive coordinate direction when the adjoining neighbour is at the same level of refinement according to

$$\begin{aligned} \mathbf{U}_{cell} &= \mathbf{U}_{cell} - \alpha \mathbf{F} \\ \mathbf{U}_{nb} &= \mathbf{U}_{nb} + \alpha \mathbf{F}. \end{aligned}$$

If a temporally adaptive strategy were employed, cells at level $(l - 1)$ would be updated only once for every two advancements at the integration level l , and as such two flux contributions through the interface between such cells would have to be summed during the integration step on the coarser level. In addition to this, the greater cell volume of the coarser cell must be taken into account by modification of the coefficient α

$$\begin{aligned} \mathbf{U}_{cell} &= \mathbf{U}_{cell} \mp \alpha \mathbf{F} \\ \mathbf{U}_{nb} &= \mathbf{U}_{nb} \pm \frac{\alpha}{4} \mathbf{F}. \end{aligned}$$

In contrast, cells at level $(l + 1)$ would be updated twice for every advancement at the integration level l . Hence two flux contributions through the interface with a more refined neighbour would have already been calculated during the integration procedure at the finer level. Hence no new fluxes need be calculated at interfaces with finer cells. The integration procedure is summarised by the following pseudo code.

```

determine maximum time step from CFL limit
for levels =  $l_{max}$  to  $l_{min}$  {
  if cell at level  $l$  is a leaf {
    for  $i = 1$  to number of dimensions {
      positive direction {
        modify state vectors if neighbour cell is equally or
        less refined
      }
      negative direction{
        modify state vectors if neighbour cell is less refined
      }
    }
  }
}
for levels =  $l_{max}$  to  $l_{min}$  {
  calculate flow properties for leaf cells at level  $l$ 
  calculate state vector for parents at level  $l$  as a
  conservative average of children
  calculate flow properties for parents at level  $l$ 
}

```

4.2.2 Reconstruction on Adaptive Grids

As discussed in section 2.2.1, reconstruction, consisting of interpolation and limiting, aims to obtain an improved estimate of the flow state either side of an interface. Such schemes require information from the cells sharing the interface as well as their neighbours as shown in Figure 2.2. The MUSCL schemes implemented in this thesis are third order accurate on uniform meshes; however, problems occur at transitions between fine and coarse regions. At the edges of these transitional regions there exists two cell configurations that must be considered, shown in Figure 4.8 for interpolation of the right hand interface state. Figure 4.8a considers the case of a transition from fine to coarse cells. This represents a transition from a region with large flow fluctuations to one containing small

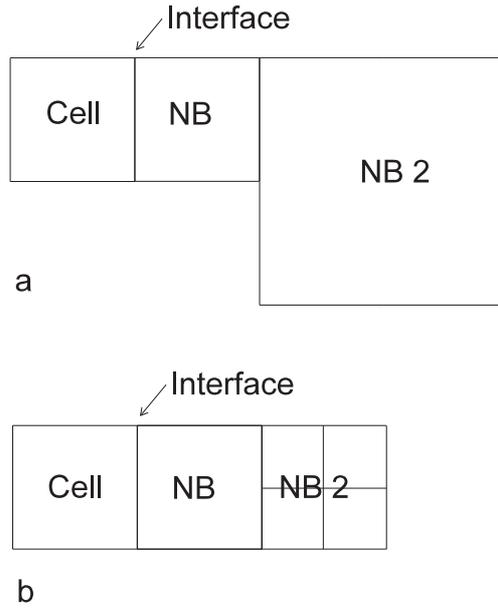


Figure 4.8: At the beginning of a transition between regions of differing refinement level, there are two possible reconstruction configurations to consider.

linear variations. In this case the van-Albada limiter introduced in Chapter 2 is used to consider only variations in the interpolation direction. In order to account for the varying cell size Johnson's [29] modification is used to alter Δ_i^- and Δ_i^+ such that

$$\Delta_i^- = \frac{2x_i(Q_i - Q_{i-1})}{x_i + x_{i-1}} \quad (4.16)$$

$$\Delta_i^+ = \frac{2x_i(Q_{i+1} - Q_i)}{x_i + 1 + x_i}. \quad (4.17)$$

Using the definition of α introduced earlier to express the length of a neighbour cell $x_{i\pm 1}$ in terms of the cells length x_i the modifications become

$$\Delta_i^- = \frac{2(Q_i - Q_{i-1})}{1 + \alpha_{i-1}} \quad (4.18)$$

$$\Delta_i^+ = \frac{2(Q_{i+1} - Q_i)}{\alpha_{i+1} + 1}. \quad (4.19)$$

Figure 4.8b considers the case of a transition from coarse to fine cells. For adaptive grids allowing several levels of refinement, such cells at intermediate levels have been refined so as to provide a smooth transition between fine and coarse levels. As such the

flow variations within these cells are usually still small and the reconstruction scheme is applied using the properties of the parent cell; an average of the children cells' properties such that conservation of mass, momentum and energy is preserved.

Within the transitional region the case of fine-coarse interfaces provides problems in accessing data from the tree as well as maintaining accuracy during the reconstruction procedure. As shown in Figure 4.9 the coarse cell neighbour required for improved accuracy cannot be determined quickly using the threaded tree structure. Since cells are refined

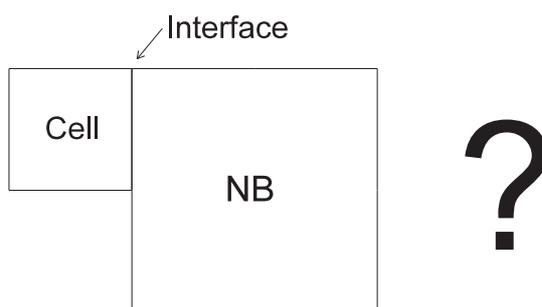


Figure 4.9: At an interface with a coarse cell (NB), it is difficult to determine the second neighbour cell required for reconstruction (indicated by ?).

based on second order variations in properties, such cells represent a linearly varying flow state of relatively small slope. Hence we can either revert to the low order scheme that assumes a constant flow state or reconstruct the right hand interface by linear interpolation. It should be noted that all cells required to determine the left hand state are obtainable from the tree structure, allowing standard reconstruction to be applied. The performance of this reconstruction technique will be discussed later with reference to specific test cases.

4.2.3 Boundary Conditions

The nature of partial differential equations is such that it is the imposed boundary conditions that determine the final solution. One technique for their implementation on non-adaptive grids is by storing ghost-cells around the perimeter of the computational domain [23]. Cell information from neighbours inside the computational domain is copied at the beginning of every time step, allowing fluxes into all computational cells to be determined irrespective of their location. This would become quite cumbersome on adaptive meshes due to the split time stepping employed. Additionally, the refinement level of ghost-cells would change in accordance with that of their adjoining computational neighbour. To

exploit the same advantages during the reconstruction/flux evaluation steps, a considerable amount of 'book keeping' would have to be done. As well as this, the refinement of ghost-cells would result in many more being present than are actually needed to describe the boundary. For this reason, all boundary conditions are implemented through the creation of a ghost-cell whenever a boundary interface is considered. Since flux evaluations are performed on an interface by interface basis, both the computational cell and the newly created ghost cell are sent to the flux evaluation routine as if they were both interior cells. The ghost-cell is then destroyed and another interface examined.

Four types of boundary conditions are currently available for simulations; inflow, outflow, walls and periodic. All boundary quads are identified by a null neighbour pointer in the direction of the boundary. Additionally, the boundary condition type is stored for this cell. For inflow boundary conditions, the prescribed inflow state is copied directly to the ghost-cell. Outflow boundary conditions involve the assumption of zero variation in stream-wise derivatives. Implementation is achieved by making the ghost-cell a direct copy of the interior cell. For supersonic outflow, no error exists since characteristics cannot propagate upstream; however, in the subsonic case the resultant error is the penalty that must be paid for isolating the computational domain from everything downstream. Wall boundary conditions involve copying the interface cell in a similar manner but reversing the direction of the normal velocity component. Finally, the use of a linked data structure allows periodic boundary conditions to be implemented in a straight forward manner. At the beginning of a simulation when neighbour links are imposed, periodic boundary cells are linked as if they were spatially co-incident. Hence the boundary is treated implicitly without the creation of ghost cells.

4.3 Test Cases

The implementation of an adaptive code requires the tuning of several additional problem specific parameters governing the development of the computational domain. For this reason, the code was first validated with the adaption subroutines turned off to ensure correct implementation of the fluid dynamics algorithms. Concurrent to this, the grid adaption functions were validated by performing several hypothetical refinement and coarsening procedures alone. Once both aspects of the code had been validated, several test cases

were re-visited to investigate the influence of several adaption related parameters.

4.3.1 Sod's One-dimensional Shock Tube Problem

The first such test case considered is Sod's one-dimensional shock tube problem [55] discussed in chapter 2.

$$\begin{aligned} x \leq 0.5 \text{ m} & : \rho_l = 1.0 \text{ kg/m}^3, P_L = 10^5 \text{ Pa}, U_L = 0 \text{ m/s} \\ x > 0.5 \text{ m} & : \rho_r = 0.125 \text{ kg/m}^3, P_R = 10^4 \text{ Pa}, U_R = 0 \text{ m/s} . \end{aligned}$$

The initial computational grid comprises 100 cells in the streamwise direction as in the quasi-one-dimensional simulation in Chapter 3. Due to the 'quad data structure' employed, the grid starts with two computational cells in the spanwise direction. Four levels of grid refinement are permitted subject to the adaption parameters $\epsilon_r = 0.08$, $\epsilon_c = 0.03$ and $\epsilon_s = 0.3$. The sensitivity parameter η is set at 0.04 and the reference density ρ_{ref} to that at the high pressure end of the tube, 1.0 kg/m^3 . The blending and compression parameters κ and β are set at $1/3$ and 2 respectively giving an upwind biased scheme that is nominally 3rd order accurate. Initially, the reconstruction options are set to linearly interpolate the interface state for the coarse cell side of an interface between cells differing in refinement level. The diaphragm is removed at $t = 0$, and the Euler equations are integrated in time using a first order Euler method subject to the stability condition $\text{CFL} = 0.5$. Refinement is considered at every time step since the CFL number permits shocks to cross a cell every 2 timesteps.

It can be seen from Figure 4.10 that a good agreement is found with the theoretical results. The expansion is captured very well. The contact surface appears smeared over several (~ 6) cells characteristic of the reconstruction scheme employed. The shock has been captured over 3 cells and the correct speed is calculated. Whilst both shock and contact surface appear smeared over several cells, the physical distance has been reduced considerably due to refinement in these regions. There appear to be slight glitches on both the leading and trailing edges of the shock and contact surface.

The final computational domain is shown in Figure 4.10 and the value of the refinement indicator ϵ in Figure 4.11. The indicator correctly detects curvature within the expansion; however, only the values at the extremities are sufficient to cause mesh refine-

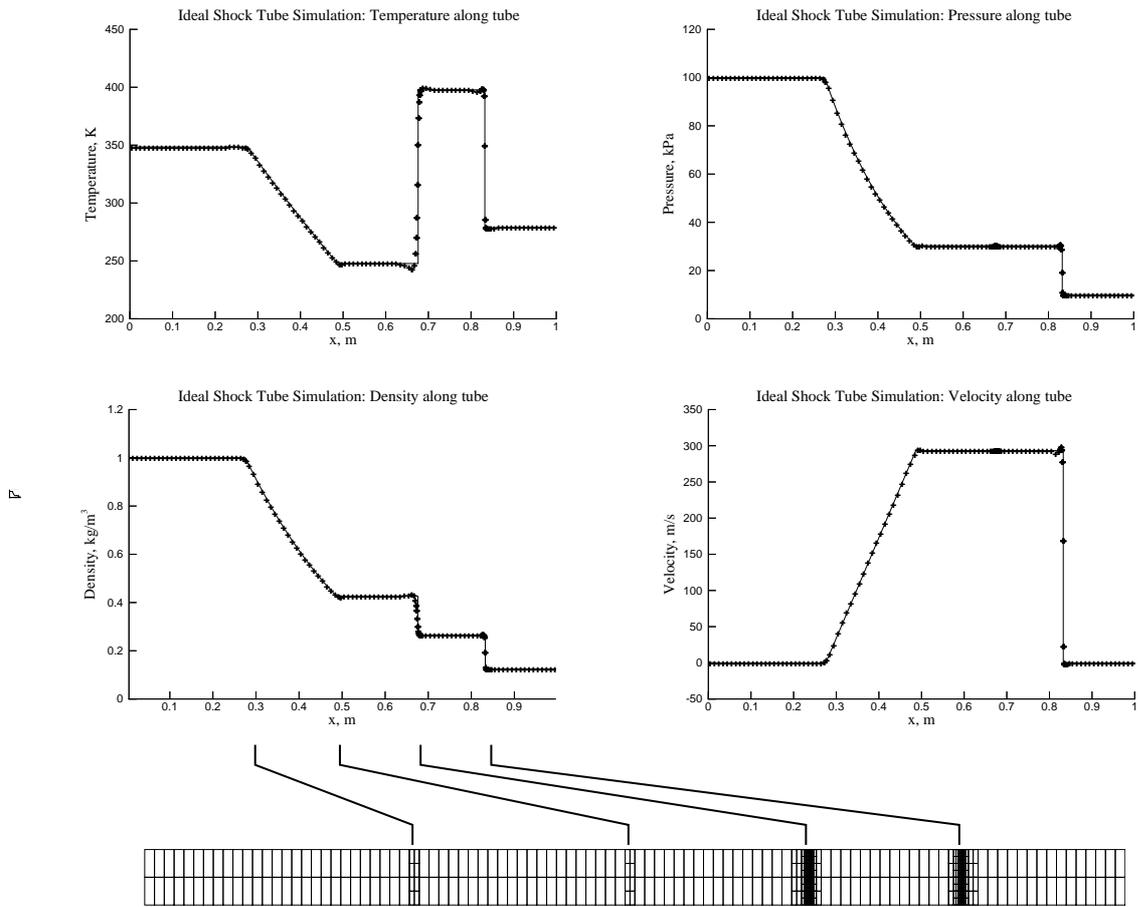


Figure 4.10: Comparison between numerical and analytical solution for Sod’s shock tube problem as well as the computational mesh ($t=0.6$ ms).

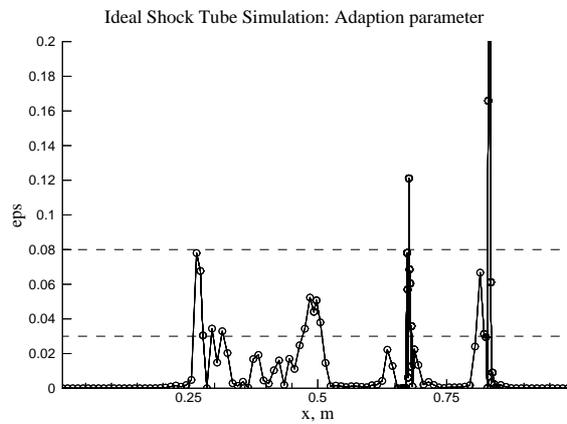


Figure 4.11: The refinement indicator at $t=0.6$ ms for Sod’s shock tube problem.

ment. High levels of refinement are evident around the shock where ϵ is large primarily due to the shock indicator. Large curvature around the contact surface is evident from the high ϵ values in that region; hence the mesh is refined accordingly. The glitches evident at the trailing edge of the shock and contact discontinuity (apparent also by moderately large values of the curvature parameter) coincide with a transition from coarse to fine mesh. These are most probably caused by the linear interpolation used during the reconstruction at interfaces between fine and coarse cells. Whilst this technique worked well within the expansion it would be expected to introduce spurious oscillations if an attempt were made to interpolate across discontinuities. For this reason, this noise is probably introduced early in the simulation when the flow has not yet developed into the clearly defined regions evident at 0.6ms [19].

In order to investigate the source of the noise generated around the shock and contact discontinuities, the simulation was repeated, however, this time first order reconstruction was used to determine the coarse interface between fine and coarse cells. The adaptation parameters were adjusted so that $\epsilon_r = 0.6$ and $\epsilon_c = 0.3$. Once again the initial conditions were integrated in time until $t=0.6$ ms. The results are shown in Figure 4.12. This time the shock and contact surface are captured free of the glitches present at the trailing edges. The solution does appear overly diffuse at the trailing edge of the discontinuity however, and there is a small amount of overshoot at the leading edge. Once again, the expansion is captured quite well, however, the high pressure end appears rather diffuse. Additionally, there are several small glitches within the expansion, This is also evident from the final computational grid and plot of ϵ along the tube (Figures 4.12 and 4.13) respectively.

Moderately high levels of curvature are detected within the expansion causing the computational mesh to be refined in this region. It appears as though this is a consequence of the first order reconstruction being used at fine-coarse interfaces. The noise generated by the inconsistency causes refinement which further amplifies the noise. The noise gets smoothed out locally, but propagated along the expansion; simulations stopped earlier than 0.6ms showed the same local mesh refinement, closer to the diaphragm location.

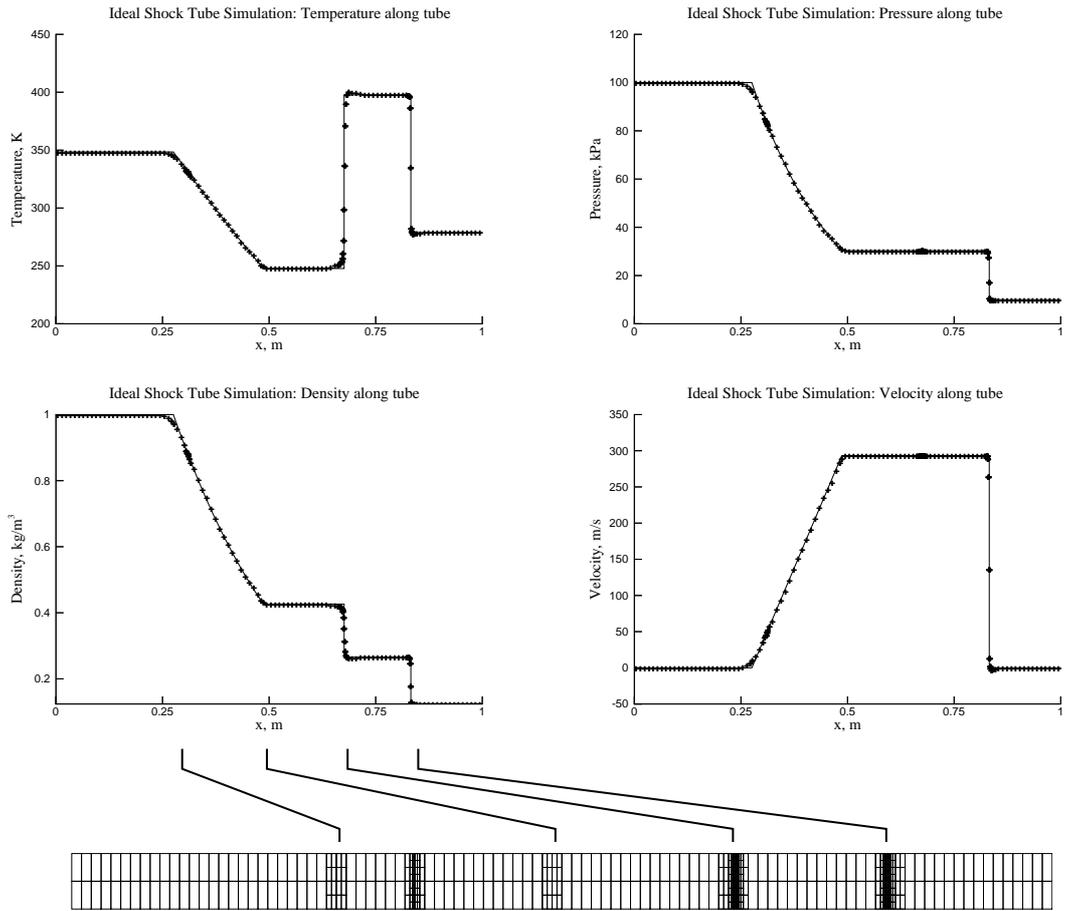


Figure 4.12: Comparison between numerical and analytical solution for Sod’s shock tube problem as well as the computational mesh ($t=0.6$ ms).

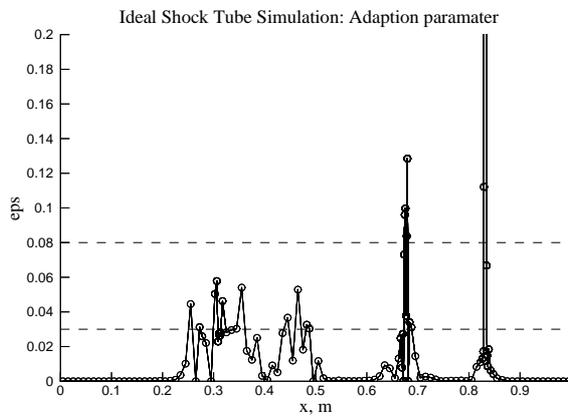


Figure 4.13: The refinement indicator at $t=0.6$ ms for Sod’s shock tube problem.

4.3.2 Mach 3 Flow over a Step

The second test case models uniform Mach 3 flow in a wind tunnel with a step. The tunnel is 3 units long, 1 unit wide, and assumed to have infinite width in the third dimension. A step 0.2 units high is located 0.6 units from the left hand side of the domain (Figure 4.14). The tunnel is filled with ideal air and impulsively started by specifying everywhere the artificial conditions

$$\rho = 1.4 \text{ kg/m}^3, P = 1.0 \text{ Pa}, u = 3 \text{ m/s}, v = 0 \text{ m/s}. \quad (4.20)$$

Gas at the same conditions flows in at the left of the domain and exits at the right hand plane. Reflecting boundary conditions are applied to all walls. This test case was first

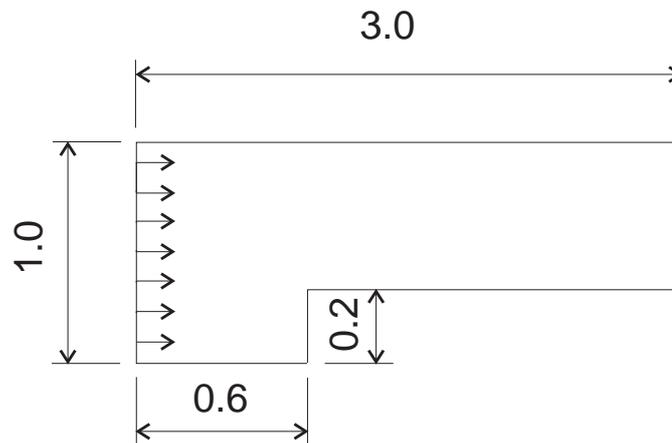


Figure 4.14: The geometry for the step (dimensions in m). The flow is initially assumed to be uniform throughout.

introduced in the 1960's [11] and has since been used by many others (see e.g. [63, 68]) as a benchmark problem for testing various schemes. As such, there exists a wealth of numerical data with which to compare results. The problem tests several features of the code including the two-dimensional formulation of the equations, strong shock interactions, and the ability to handle the singularity resulting from the expansion fan emanating from the corner of the step. The interactions between these features result in the formation of new, and in some instances unphysical features, further testing the numerical technique and adaptive algorithm.

To highlight the important features of the problem and to establish a benchmark for performance, uniform grid simulations were performed using both the first order Godunov

method (Figure 4.15) and the higher order MUSCL formulation (Figure 4.16). The initial flow field was integrated in time on a uniform mesh with $\Delta x = \Delta y = \frac{1}{80}$ to $t=4$ s subject to the stability criterion $CFL=0.3$. Both solutions are now discussed with reference to the computations of Woodward *et al* [68] which used a version of the piecewise parabolic method [5].

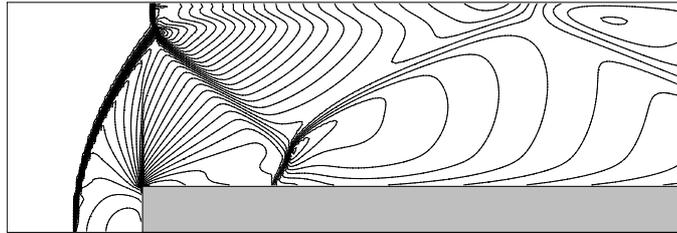


Figure 4.15: Flow field at $t=4.0$ s on a uniform mesh using Godunov's method

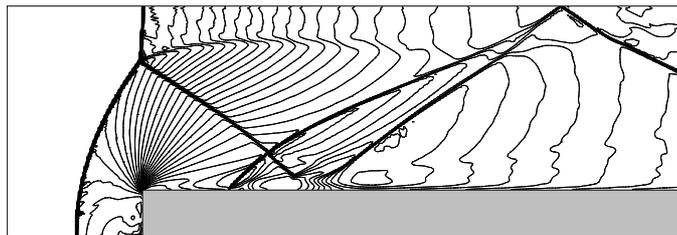


Figure 4.16: Flow field at $t=4.0$ s on a uniform mesh using the higher order MUSCL formulation

The first order Godunov method results in the computation of a flow field containing shocks whose position and shape is incorrect. A Mach stem forms on the upper wall; however, it is too short and slightly too far downstream. The contact surface originating from the lambda structure is captured well since it is nearly aligned with the computational grid. Shock waves, too, are captured well when aligned with the mesh; however, the lack of artificial smearing results in numerical instabilities as evident by the glitches in contours just downstream of such features. A second, spurious Mach stem forms due to the strong numerical boundary layer present on the lower wall of the duct. Finally, an *unphysical* expansion shock stems from the corner of the step. Whilst this phenomenon is a solution to the Euler equations, it produces a decrease in entropy which is not physically realisable. Other methods of flux calculation have been shown to also produce similar expansion shocks [29]. Some such methods employ an “entropy fix” to reduce the effect,

however it has been noted that the application of a reconstruction step eliminates the problem altogether [29] as can be seen from the higher order MUSCL results in Figure 4.16.

The higher order MUSCL simulation produces thin shocks whose position and shape are much more accurate. The Mach stem on the upper wall is in the correct location and has the correct length. It appears very thin since it is aligned with the mesh, however this once again results in some numerical noise. The effects of the numerical boundary layer on the lower wall appear worse than in the Godunov case. The entropy layer is the result of trying to capture the singularity at the corner with a single cell. Woodward *et al* suggest a fix: by modifying cells in this region, they were able to decrease its effect. Under such conditions, the MUSCL scheme produced results superior to the first order scheme in this region. The second contact surface is resolved, however it has incorrect position due once again to the numerical boundary layer. Finally, as mentioned earlier, an unphysical expansion shock no longer appears.

The inadequacies with the first order scheme noted earlier, which are only rectified by the introduction of a less dissipative technique provide a means of evaluating the performance of solution reconstruction on adaptive grids. Computations were performed using the adaptive algorithm subject to the parameters $\epsilon_r = 0.06$, $\epsilon_c = 0.05$ and $\epsilon_s = 0.3$. A sensitivity parameter η of 0.04 was used and several reference densities ρ_{ref} trialed to compare the development of the computational mesh. Four levels of refinement were permitted such that the resulting finest computational cells had $\Delta x = \Delta y = \frac{1}{80}$. Setting ρ_{ref} to the initial density of 1 resulted in unnecessary refinement downstream of the leading shock as shown in Figure 4.17 at $t=1.0$ s. By setting the reference density to 6, indicative of the maximum density in the flow, the amount of unnecessary refinement was reduced considerably.

Once again, the flow equations were integrated in time to $t=4$ using the first order temporal scheme with a CFL number of 0.3. The time progression of the computational solution is shown in Figures 4.18 and 4.19. MUSCL reconstruction is employed with $\kappa = \frac{1}{3}$ and $\beta = 2$. The reconstruction reverts to a first order scheme for determining the coarse side of an interface between fine and coarse cells. This should not introduce significant error as the unrefined cells should have low error indicators.

Figure 4.18 shows contours of density and the corresponding computational mesh dur-

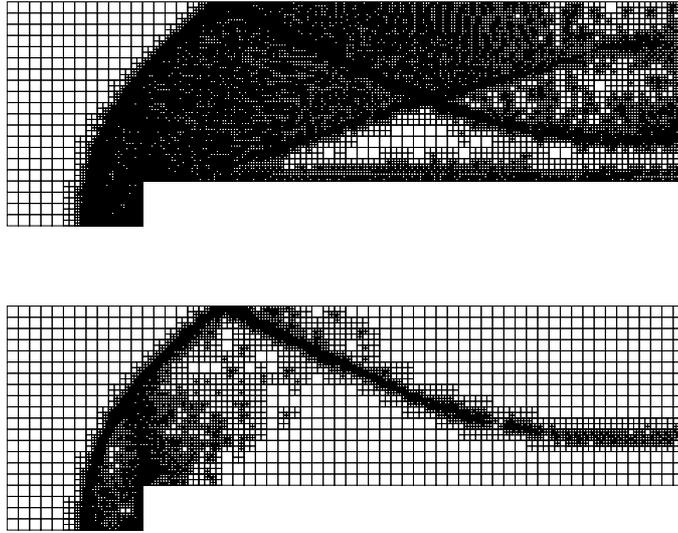


Figure 4.17: Computational mesh at $t=1.0$ s using reference density of 1 kg/m^3 (top) and 6 kg/m^3 (bottom).

ing the early stages of the simulation. The refinement indicators (as implicitly displayed in the refined mesh) appear to be distinguishing the shock and rarefaction corner well. At $t=2.0$ s, the Mach stem forms on the upper wall as is evident from Figure 4.19. The refinement indicators detect the contact surface emanating from the triple point and refine accordingly. At $t=3.0$ s the unphysical transition from regular to Mach reflection occurs on the lower wall. The theoretical shock angle at which this occurs for the given conditions is approximately 40 degrees, producing a deflection angle of 20 degrees. Since the Mach number decreases in the entropy layer, a two-shock regular reflection is incapable of deflecting the flow to this degree and hence the transition occurs. It is interesting to note that whilst the uniform MUSCL simulation resulted in a large Mach stem on the lower wall, computations on the adaptive domain suffered only slightly from the numerical boundary layer; this is probably a result of the cells along the lower wall being relatively coarse. Unfortunately, the second contact surface was not resolved since the refinement indicators were not sensitive enough to refine the mesh in this region.

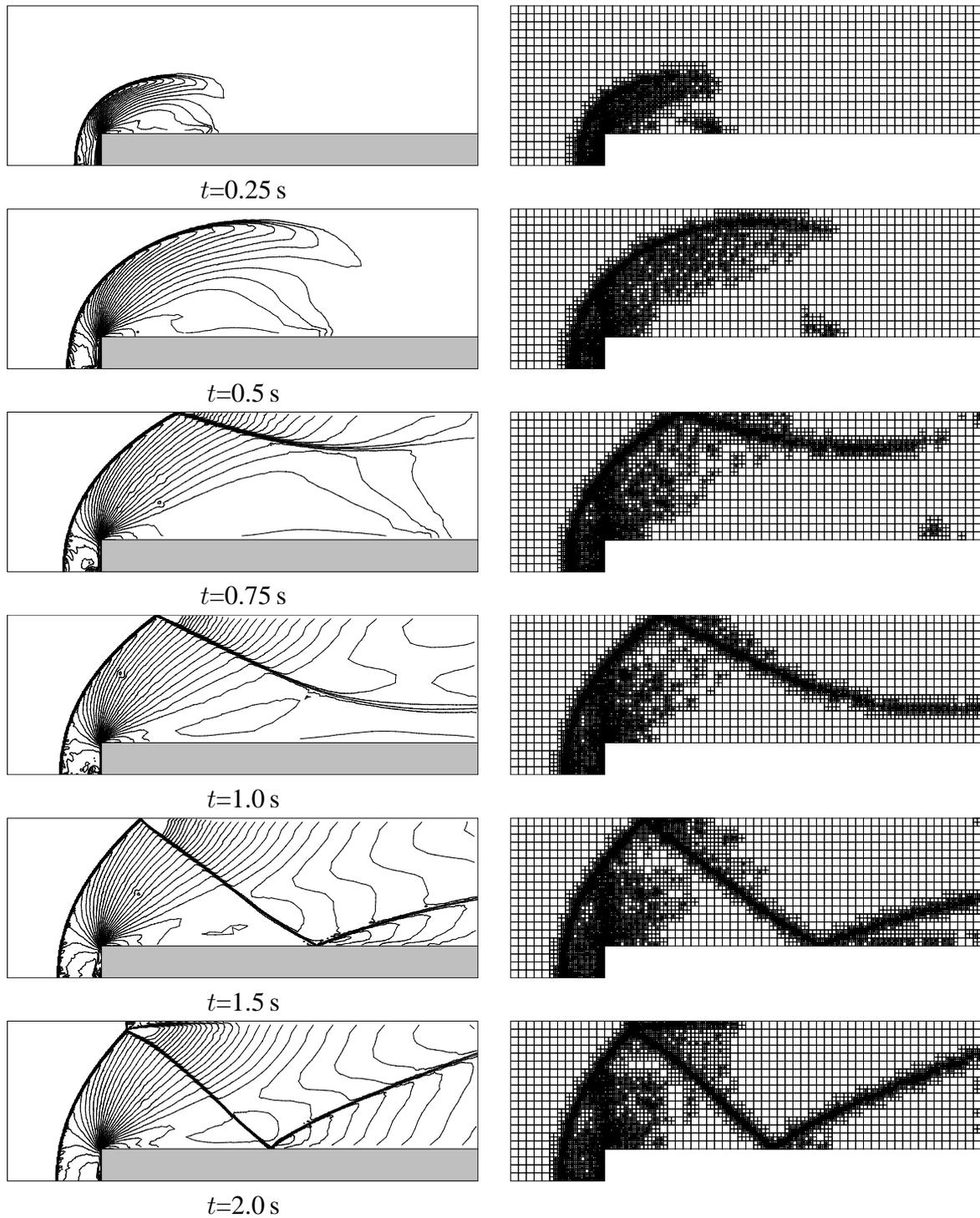


Figure 4.18: Evolution of the flow domain for the first two seconds. Contours of density (left) and the computational mesh (right).

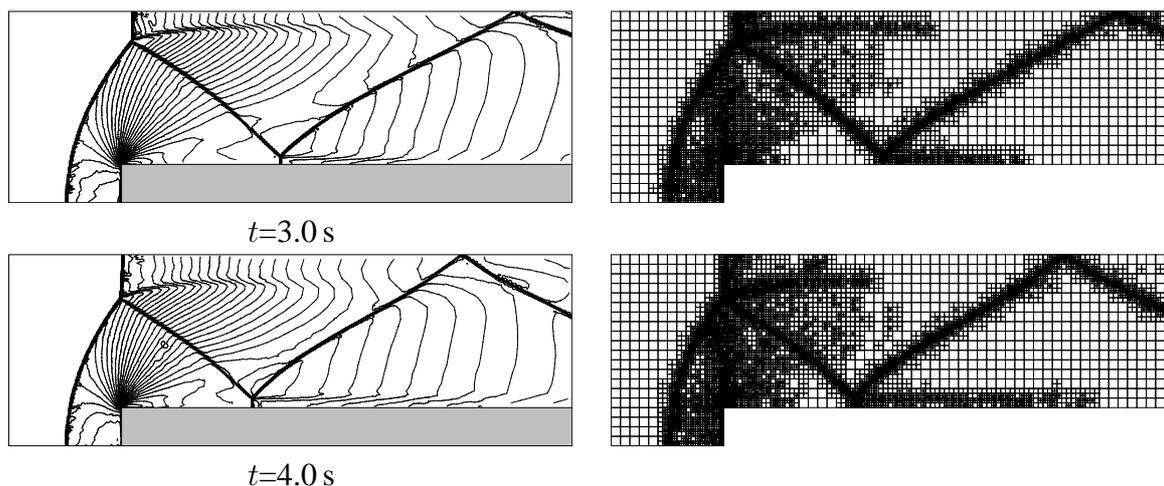


Figure 4.19: Evolution of the flow domain for the final two seconds. Contours of density (left) and the computational mesh (right).

4.3.3 Shock Wave Diffraction around a 90 degree Sharp Corner

The final test case considered concerns the diffraction of a shock wave around a 90 degree sharp corner. This test case was chosen as the benchmark problem for computational simulations at the 18th International Symposium on Shock Waves [58]. As such, a large amount of computational data from state of the art codes using a variety of numerical techniques is available. The geometry of the duct is shown in Figure 4.20

The incident shock wave has a Mach number of 1.5, the ambient medium being air at room temperature and atmospheric pressure. A large scale vortical structure is formed by the shear layer emanating from the corner. The high resolution simulation of Figure 4.24 shows that this shear layer rolls up forming discrete vortices due to the Kelvin-Helmholtz instability [70]. The expansion fan present on the corner is terminated by two shock waves which match the conditions of the expanded flow to that behind the diffracted shock wave. Within the large vortex there exists two distinct shocks (termed the left and right vortex shocks) whilst beneath it lies a contact surface which emanates from some point on the primary shock wave.

The ISSW problem involved computing density contours at the instant when the diffracted shock wave was 1 length unit from the 90 degree corner. Contours of density were chosen for the visualisation, with the official ISSW format calling for each isopycnic to correspond to an increase of 4% of the initial density.

The presence of these difficult-to-resolve flow features, particularly associated with

the internal structure of the vortex, provides a means of assessing the spatial accuracy of the code. Indeed, preliminary adaptive calculations using only the first order Godunov method showed that this scheme was too dissipative to resolve the vortex core as can be seen from Figure 4.20.

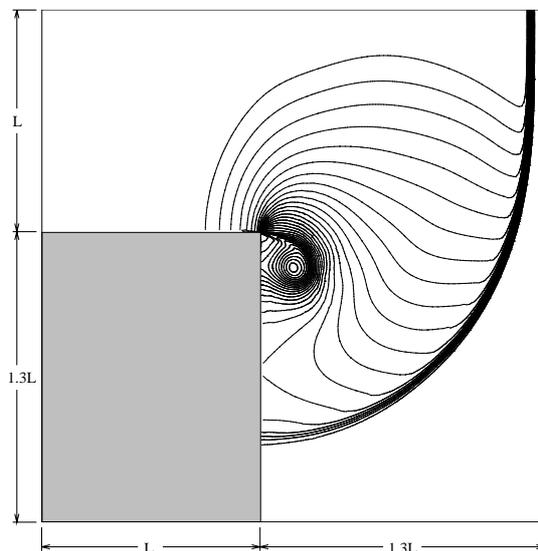


Figure 4.20: Contours of density in ISSW format using the first order Godunov method on a uniform mesh

The preliminary mesh was set with $\Delta x = \Delta y = \frac{1}{40}$ so as to provide adequate resolution for the plotting of contours behind the diffracted shock wave. Computations were performed using the adaptive algorithm subject to the parameters $\epsilon_r = 0.06$, $\epsilon_c = 0.05$ and $\epsilon_s = 0.3$. A sensitivity parameter η of 0.04 was used and the reference density was set as the cell centred value; this allows accurate calculation of the refinement indicator within the vortex core, where the value of density drops to a minimum. Simulations allowing two, three, four and six levels of refinement were performed subject to the *CFL* limit of 0.1. Contours of density as well as the final computational mesh are presented in Figures 4.21-4.24. With two levels of refinement it can be seen that the key elements of the flow field are captured. The diffracted shock is resolved well despite not being aligned with the computational mesh. The expansion fan at the corner is visible; however, the associated shock waves are not resolved well. The internal structure of the vortex is also not resolved. In addition, there exists some noise in the flow behind the primary shock wave as is evident from the computational mesh which has been refined spuriously in some regions. By allowing three levels of refinement, more detail of the interaction becomes apparent.

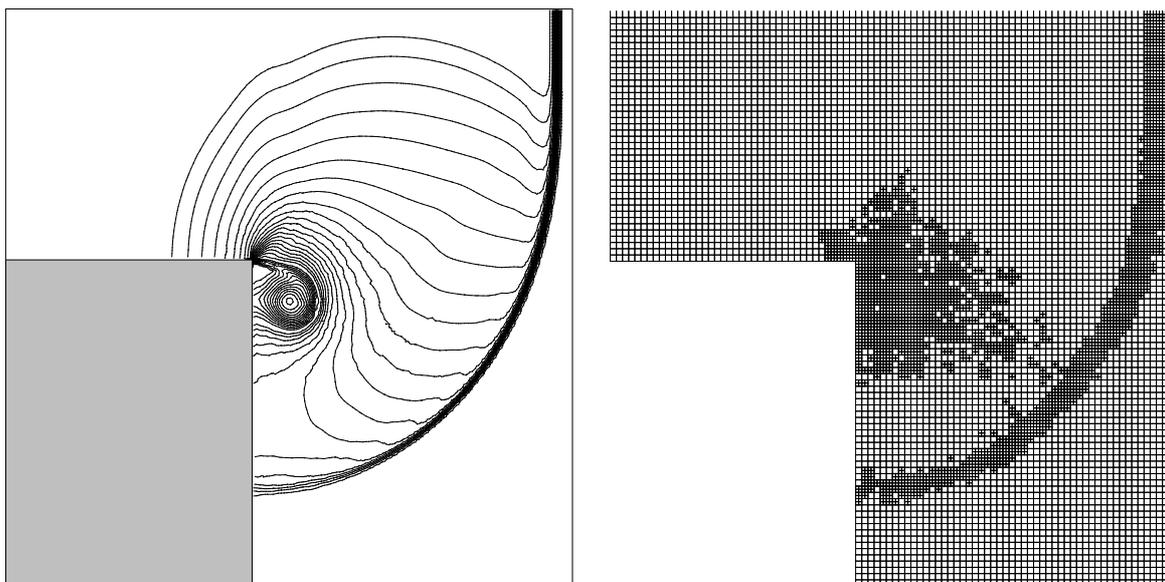


Figure 4.21: Two levels of mesh refinement. Contours of density in ISSW format

The noise behind the primary shock wave has been diminished somewhat and no spurious refinements are present. The two shock waves terminating the expansion fan are clearly visible. The corner vortex appears to have some internal structure to it; in particular, the right vortex shock can be distinguished. The primary shock wave appears more crisp because it is resolved over a smaller physical distance. When four levels of refinement are allowed, both vortex shocks are resolved. Additionally, the contact surface is becoming apparent. Finally, 6 levels of refinement were permitted in order to provide a high resolution simulation of the interactions. The internal structure of the vortex can clearly be seen, and both shocks are resolved well. Discrete vortices associated with Kelvin-Helmholtz roll-up of the shear layer are also apparent. The contact surface beneath the corner vortex, whilst present, is difficult to distinguish. This is most probably due to deficiencies in the solution reconstruction technique applied at fine/coarse boundaries. Such boundaries would be expected away from the centre of the vortex core as the mesh adjusts slowly to the more uniform flow behind the primary shock.

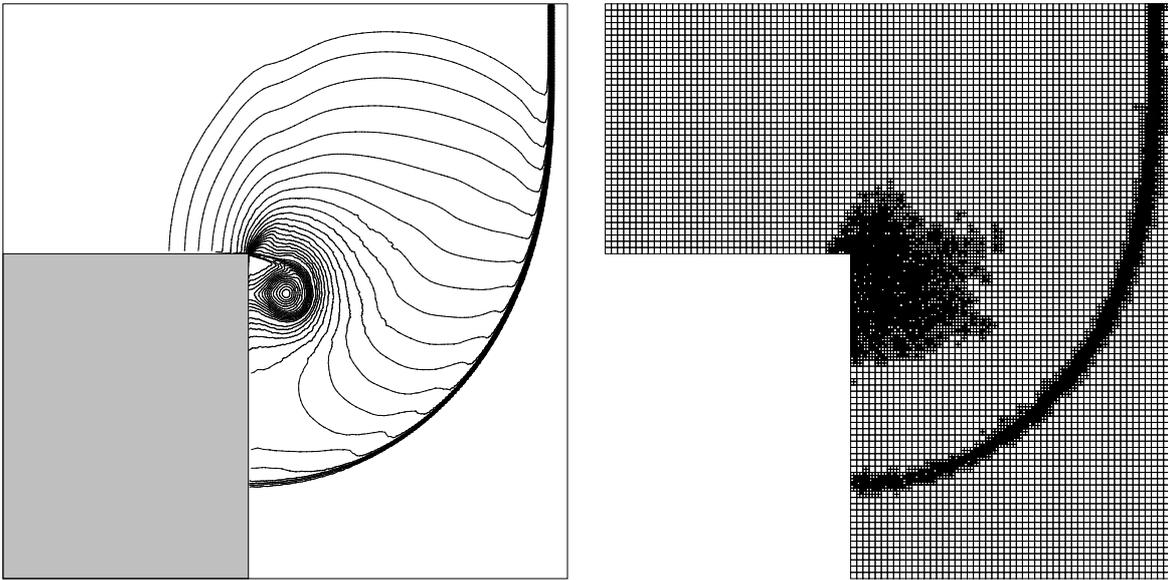


Figure 4.22: Three levels of mesh refinement. Contours of density in ISSW format

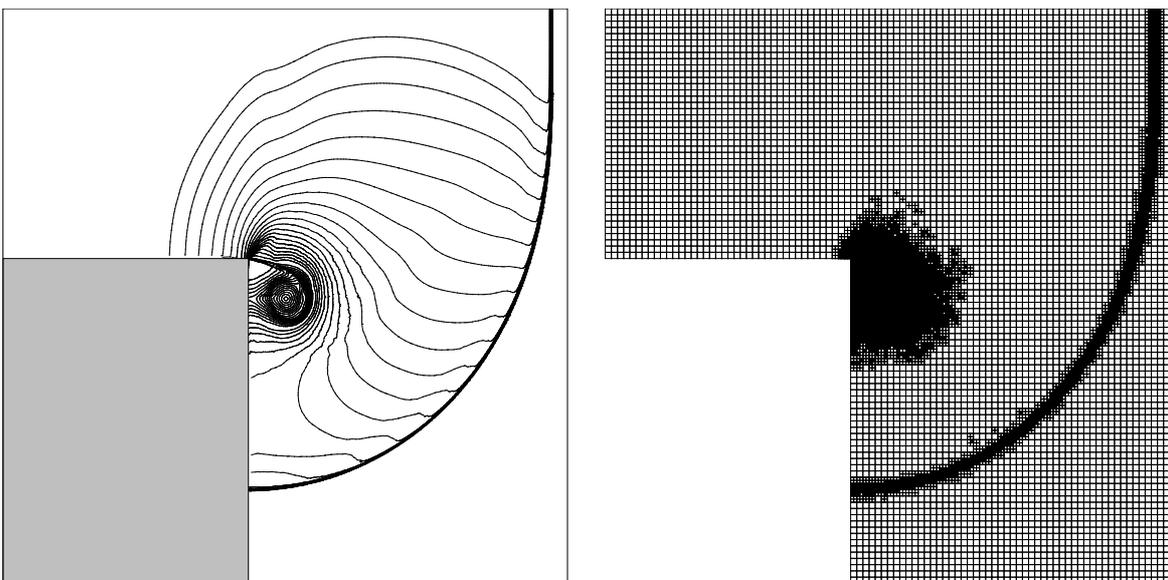


Figure 4.23: Four levels of mesh refinement. Contours of density in ISSW format

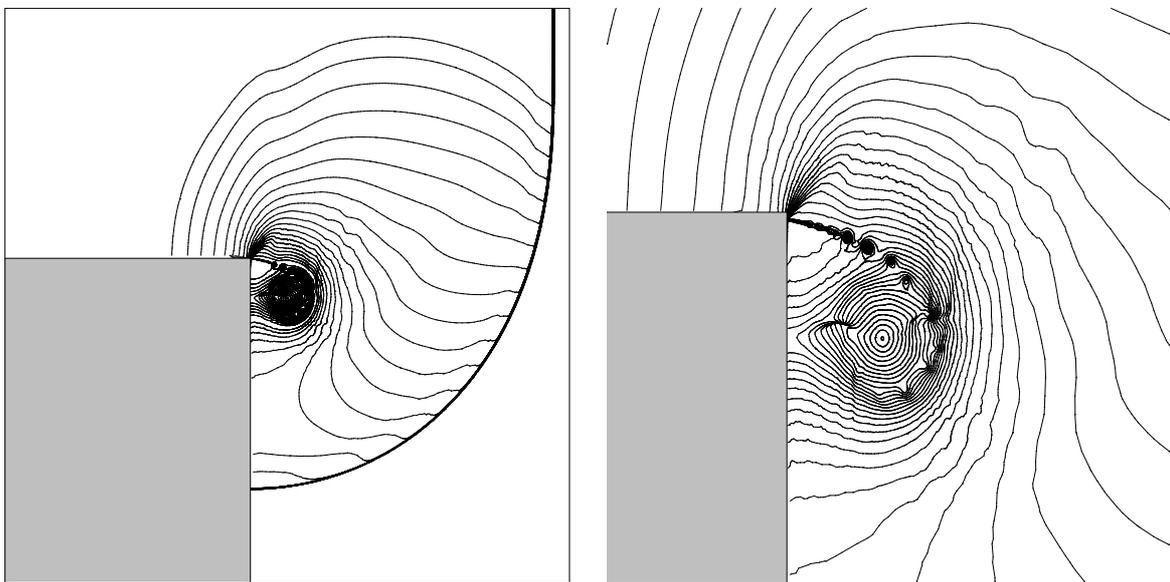


Figure 4.24: Six levels of mesh refinement. Contours of density in ISSW format

Shock-Vortex Interactions

Shock-vortex interactions are an important area of research both from a fundamental phenomenological perspective [35] and due to their vast range of interdisciplinary applications [66]. In the past much effort has been devoted to experimental work that carefully establishes a single vortex and focuses on its interaction with a single shock wave. Such experiments generally involve the diffraction of a shock wave to produce a patch of vortical flow. The vortex is processed either by the reflected primary shock wave on its return [53, 38] or by a secondary, delayed wave [54, 13]. Recently, Skews revisited his earlier work [53] monitoring the experiments for a longer period to investigate the effect of multiple shock-vortex interactions [52]. Under certain incident shock strengths and diffraction angles it was found that the second passage of the shock wave destroys the vortex completely, leaving a patch of turbulent flow. Skews' study provides the motivation for the current work and will now be described in greater detail. The propagation of a shock wave around a sharp bend causes a vortex to be shed from the corner. A portion of the shock wave reflects from the far wall and returns to interact with the vortex in a manner which has been well studied [46]. The characteristic flow features for this scenario are shown in Figure 5.1. The vortex is stretched to some degree by the first interaction, however it remains otherwise unaffected. The shock wave on the other hand is deformed considerably as it is "swept around" by the vortex. The shock then undergoes a second reflection from the near wall and returns to interact with the vortex once more. This time the passage of the shock wave through the vortex is more disruptive and in some instances the vortex is destroyed completely leaving a patch of turbulent flow. A detailed understanding of the mechanisms behind the vortex breakup could give insights into the development of compressible turbulence [52] and aid the development of turbulence models that capture its macroscopic effect.

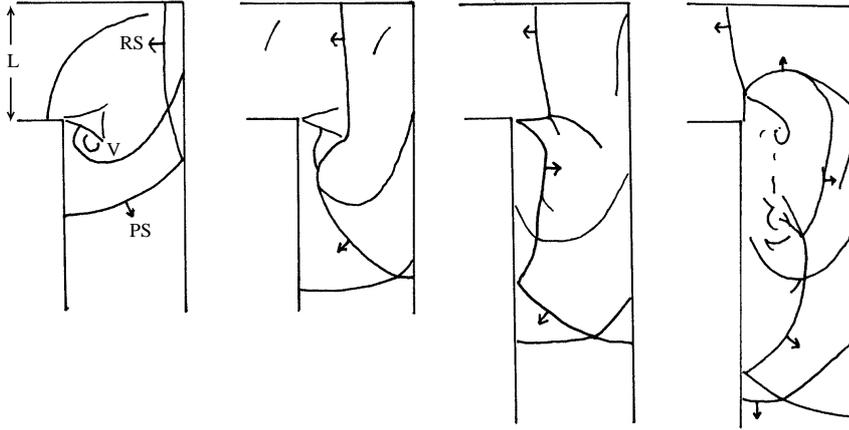


Figure 5.1: Flow development for the interaction between a Mach 2.0 shock wave and a vortex shed from the corner: vortex V, primary shock PS, reflected shock RS. Adapted from Figure 2 in [52]

5.1 Compressible Turbulence

In order to understand the nature of compressible turbulence it helps to consider further the governing equations presented in Chapter 2. The energy equation shows that the kinetic energy of the flow can be degraded into internal or thermal energy through viscous dissipation. As mentioned earlier an expression for this dissipation rate can be obtained by considering the stress-strain relation for a Newtonian fluid under Stoke's hypothesis of zero bulk viscosity. Under this assumption the deformation stress τ_d is related to the deformation strain ω_d by

$$\tau_d = 2\mu\omega_d \quad (5.1)$$

The total strain on a fluid element is made up of both ω_s spherical (non-deforming) and ω_d non-spherical (deforming) components. By examining a differential fluid element this strain rate can be expressed in tensor form as

$$\omega = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (5.2)$$

Since the spherical strain in any coordinate direction is one-third of the fluid divergence, an expression for the deformation strain ω_d can be obtained and substituted into 5.1.

Hence the rate of dissipation of kinetic-energy per unit mass is obtained [35].

$$\frac{d\epsilon}{dt} = \left\{ \nu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \nu \frac{\partial u_k}{\partial x_k} \delta_{ij} \right\} \frac{\partial u_i}{\partial x_j} \quad (5.3)$$

By considering the length, time and velocity scales

$$u_i^* = \frac{u_i}{u_\infty}, \quad x_i^* = \frac{x_i}{L}, \quad t^* = \frac{t}{L/u_\infty}, \quad \nu^* = \frac{\nu}{\nu_\infty}, \quad \epsilon^* = \frac{\epsilon}{u_\infty^2} \quad (5.4)$$

the dissipation rate can be non-dimensionalised as

$$\frac{d\epsilon^*}{dt^*} = \frac{1}{Re_L} \left\{ \nu^* \left(\frac{\partial u_i^*}{\partial x_j^*} + \frac{\partial u_j^*}{\partial x_i^*} \right) - \frac{2}{3} \nu^* \frac{\partial u_k^*}{\partial x_k^*} \delta_{ij} \right\} \frac{\partial u_i^*}{\partial x_j^*} \quad (5.5)$$

where Re_L is the Reynolds number defined as

$$Re_L = \frac{u_\infty L}{\nu_\infty} \quad (5.6)$$

Since viscosity is a small number it can be seen that for flows of practical interest, the Reynolds numbers are large and the viscous dissipation rates low except in regions characterised by velocity changes over very small length scales. The phenomenon of turbulence can then be seen as a mechanism for the energy cascade of large scale structures into small scale motions where viscous dissipation can become effective. The first term on the right of equation 5.5 represents this energy cascade to the smallest eddies (the so called Kolmogorov length scale) through vortex-vortex interactions. The second term represents a compressible mechanism for this phenomenon with the shock waves performing a similar role to Kolmogorov eddies [35].

In turbulent high speed compressible flow, shock waves as well as regions of small scale vortical motions must be present. Hence shock-shock, vortex-vortex as well as shock vortex interactions play an important role in the dissipation of kinetic energy. As observed by Skews [52] this latter interaction can degrade large structures into small scale motions in very short time-scales under certain circumstances. The underlying mechanism behind this represents a significant contribution to the energy cascade process and as such is of fundamental importance to the understanding of compressible turbulence.

Numerical modelling of turbulent flows requires a full three dimensional simulation

with the spatial and temporal resolution to capture all such scales. Specialised codes are required to perform these calculations in reasonable time frames, and for flows containing moderately strong shock waves, are only possible for very simply flow geometries [72]. Whilst LES can lead to significant decreases in computational effort it has been shown that meaningful results are obtained only with high resolution in the vicinity of shock waves [71]. Additionally, accurate turbulent simulations require the use of numerical schemes with a higher resolving accuracy and hence higher computational cost than those presented in this thesis. Hence, even an LES of the complicated flow structure that result from the propagation of a shock wave around a 90 degree bend cannot be achieved in a feasible time given present day computing power. The present two dimensional simulations aim to investigate only a single scale in the flow mechanism involved in the break-up of the large vortex shown in figure 5.1. It is hoped that the details of this process are, to a first approximation, independent of viscosity and the underlying fine scale interactions characteristic of a turbulent flow. For this reason, we now turn our attention away from the ensemble of interactions that characterise a turbulent flow and focus on the detailed modelling of the shock wave vortex interactions.

5.1.1 Previous Modelling Work

Much work in the early 1950's was focused at analysing shock-vortex interactions as a mechanism for noise production in supersonic aircraft. The idea was to represent the noise generated as turbulence passed through the shock pattern of a jet, by the passage of a shock wave through a single vortex. Analytical theories that modelled the resulting acoustic wave were devised [46] and improved upon [66] to the extent where a good prediction of the overall 'far-field sound emission' could be made. Improvements in computer power and numerical algorithms over the last two decades, has enabled computational simulations of shock-vortex interactions to be made.

At the 18th International Symposium on Shock Waves, a special poster session was held to compare various CFD codes. The benchmark problem for the comparison was the diffraction of a Mach 1.5 shock wave around a 90 degree bend. A variety of numerical schemes were used to model the problem, solving either the Euler or Navier-Stokes equations. Whilst the simulations involved only the early development of the vortex, ending before the reflection of the primary shock-wave, many were able to resolve its internal

structure [58].

In a later study, Uchiyama *et al* [61] allowed the simulation to progress further. They showed numerically that due to the growth of the Kelvin-Helmholtz instability, the shear layer emanating from the corner rolls up to form successive vortices. The interactions between these discrete vortices and the secondary shock (internal to the vortex) were shown to be responsible for the bifurcation of this shock in some instances. However, this study concerned the internal structure of the vortex and no attempt was made to model its interaction with the primary shock wave.

Matsuura *et al* [36] conducted a numerical and experimental study of a similar situation in which the outlet (the lower boundary of the duct in Figure 5.1) was replaced with a wall. Their Navier-Stokes simulations agreed well with experimental data although little could be concluded about the second shock-vortex interaction since, by the time of this interaction, waves had reflected from the bottom wall and influenced the flow near the vortex.

In a bid to explain the experimental observations of Skews, Timofeev *et al* [59] conducted a two-dimensional Euler simulation of the shock diffraction problem, allowing the shock to process the vortex twice. Whilst coarse grid simulations appeared to result in vortex break up after two transits of the reflected shock, finer simulations showed the problem to be grid-dependent; vortex break-up after a single passage (only) was observed in some instances. This early breakup is not consistent with the experimental data. Several ideas for the deficiency of these simulations have been proposed, including the applicability of the Euler analysis as well as neglect of three dimensional effects.

This thesis aims to return to this problem and take another look at the 2 dimensional interactions using a new simulation code. It is hoped that the present study will serve as a precursor to a fully three dimensional study utilising the same code structure and grid adaption techniques.

5.2 Interaction with a Mach 1.5 Shock Wave

The interaction of a corner vortex with a shock wave was investigated numerically using the experimental conditions of Skews. Simulations were performed on a computational

domain, non-dimensionalised by the duct width of $L = 50.8$ mm and the shock speed of 519 m/s. The characteristic time scale being $L/u_s = 50.8 \times 10^{-3}/519$ s. The domain extended 1 characteristic length (L) upstream and 2 characteristic lengths downstream of the corner. Initially a Mach 1.5 shock wave was established within the duct by specifying, along the upstream entrance, post shock conditions determined from the shock Hugoniot. An outflow condition imposes zero streamwise derivatives along the exit plane. Whilst the shock induced flow is subsonic allowing information from this boundary condition to corrupt the upstream flow state, it is deemed sufficiently far from the corner to play a negligible role in the time frames under consideration; the most interesting part of the shock-vortex interaction is completed before any ‘spurious’ waves from the outflow boundary can influence it.

MACS2d is used to advance the two-dimensional inviscid flow equations subject to a CFL limit of 0.1. The refinement parameters ϵ_r and ϵ_c are set to 0.06 and 0.05 respectively with the sensitivity parameter η set to 0.03. The initial cell size was set as $\frac{1}{40}$. Simulations were performed using both 4 and 6 levels of refinement as shown in Figures 5.2 through 5.6 and Figure 5.8 respectively. The first ‘snap-shot’ of the evolving flow domain was made at time $t = 38.746$. Contours of density as well as a plot of the computational flow domain are shown in Figure 5.2(top). The primary shock wave appears to be captured quite well as is the characteristic corner vortex. For clarity, only a limited number of contours are shown; however, the structure and computational domain resembles that shown in the previously computed test case.

The next flow-instant is captured at $t = 48.197$ and shown in Figure 5.2(bottom). A portion of the primary wave has reflected from the far wall and is returning towards the vortex. The corresponding computational mesh is adjusting to the post-reflected-shock flow state which should be nearly stagnant against the far wall. Small regions appear to remain refined and this may be due to inconsistencies in the flux reconstruction step at boundaries between cells that differ in volume. Simultaneously, the head of the corner expansion wave is propagating slowly upstream since the post shock flow is subsonic.

The third flow-instant is captured at time $t = 64.262$. Figure 5.3(top) shows that, by this time, the reflected shock has returned to process the vortex. The lower portion of the reflected wave is swept around under the vortex due to the higher velocity in the propagation direction associated with the clockwise rotation of the vortex. In contrast to the case

of the interaction of a shock wave with an isolated vortex, the upper portion of the shock wave is not affected to nearly the same degree. This is due to the slipstream emanating from the corner; the high gas velocity above the slipstream is isolated from the effects of the vortex. Also, the transition from regular to Mach reflection has occurred where the diffracted shock meets the far wall. By $t = 68.042$, the reflected shock has processed the vortex and a portion of it has reflected from the near wall. Another portion of the wave has been swung around by the vortex to the point where it is about to interact with the slipstream. Meanwhile the upper portion of the primary shock wave is developing a slight bend as the lower end interacts with the flow just downstream of the corner expansion.

Figure 5.4(top) shows the flow state at $t = 79.382$. By this time the portion of the shock reflected from the near wall has passed through the vortex for the second time resulting in a complicated flow field. The interaction between the shock wave segment swung around by the vortex and the slipstream has led to the formation of several new shocks. The upper portion of the primary shock wave has propagated further upstream however it has still not reached the corner. Meanwhile part of the diffracted shock has exited the flow domain. The flow domain after the interaction is shown in Figure 5.4(bottom). The computational mesh exhibits high levels of refinement around several key features that have evolved during the course of the simulation. It can be seen from the evolution of this mesh that all areas of the flow domain have undergone the transition from coarse to fine cells at least once, making the computational savings of the adaptive technique apparent.

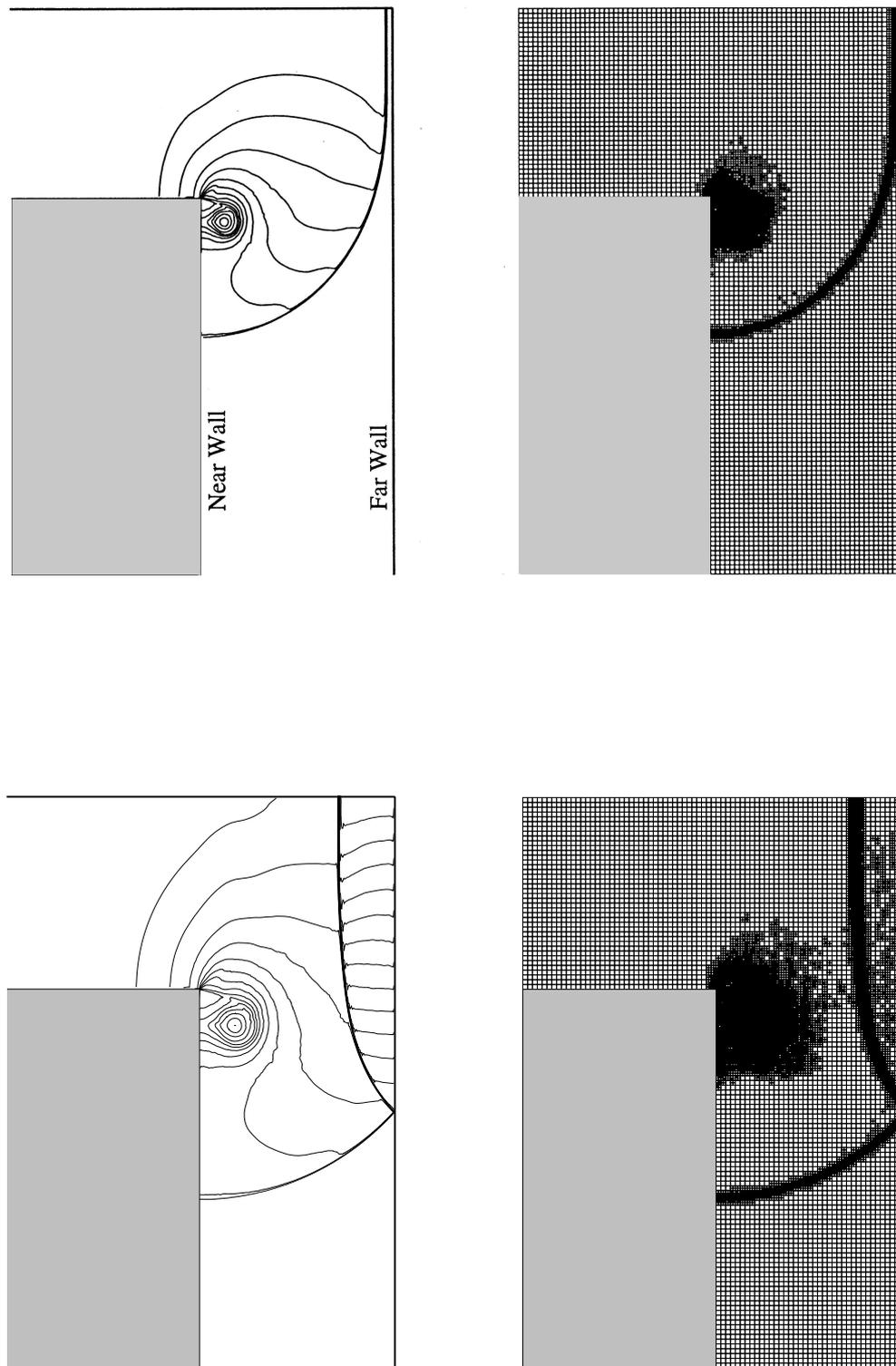


Figure 5.2: Contours of density (left) and the computational mesh (right) for the diffraction of a Mach 1.5 shock wave around a 90 degree bend. Simulation time $t = 38.746$ (top) and $t = 48.197$ (bottom).

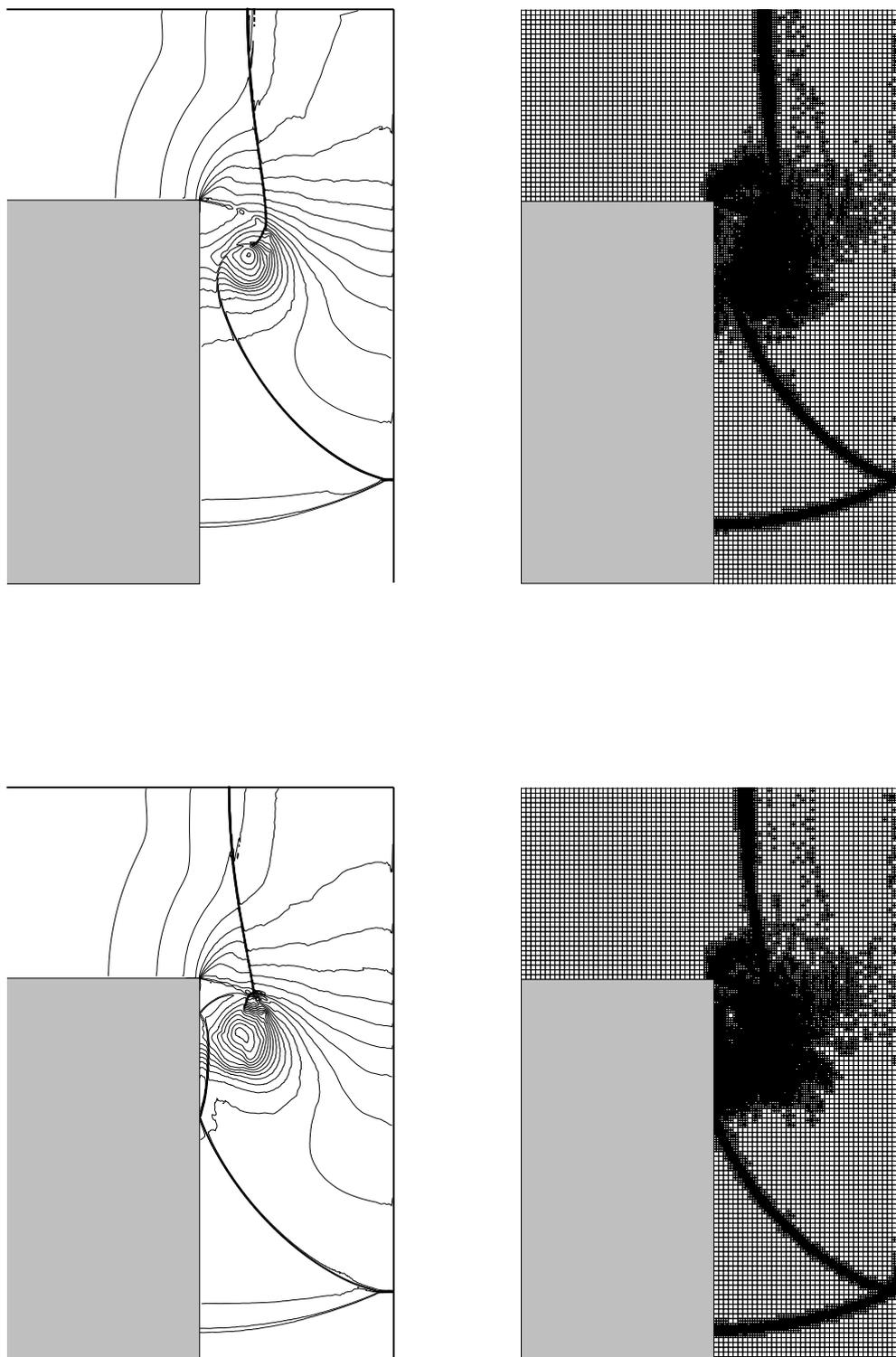


Figure 5.3: Contours of density (left) and the computational mesh (right) for the interaction of a Mach 1.5 shock wave with a corner vortex. Simulation time $t = 64.262$ (top) and $t = 68.042$ (bottom).

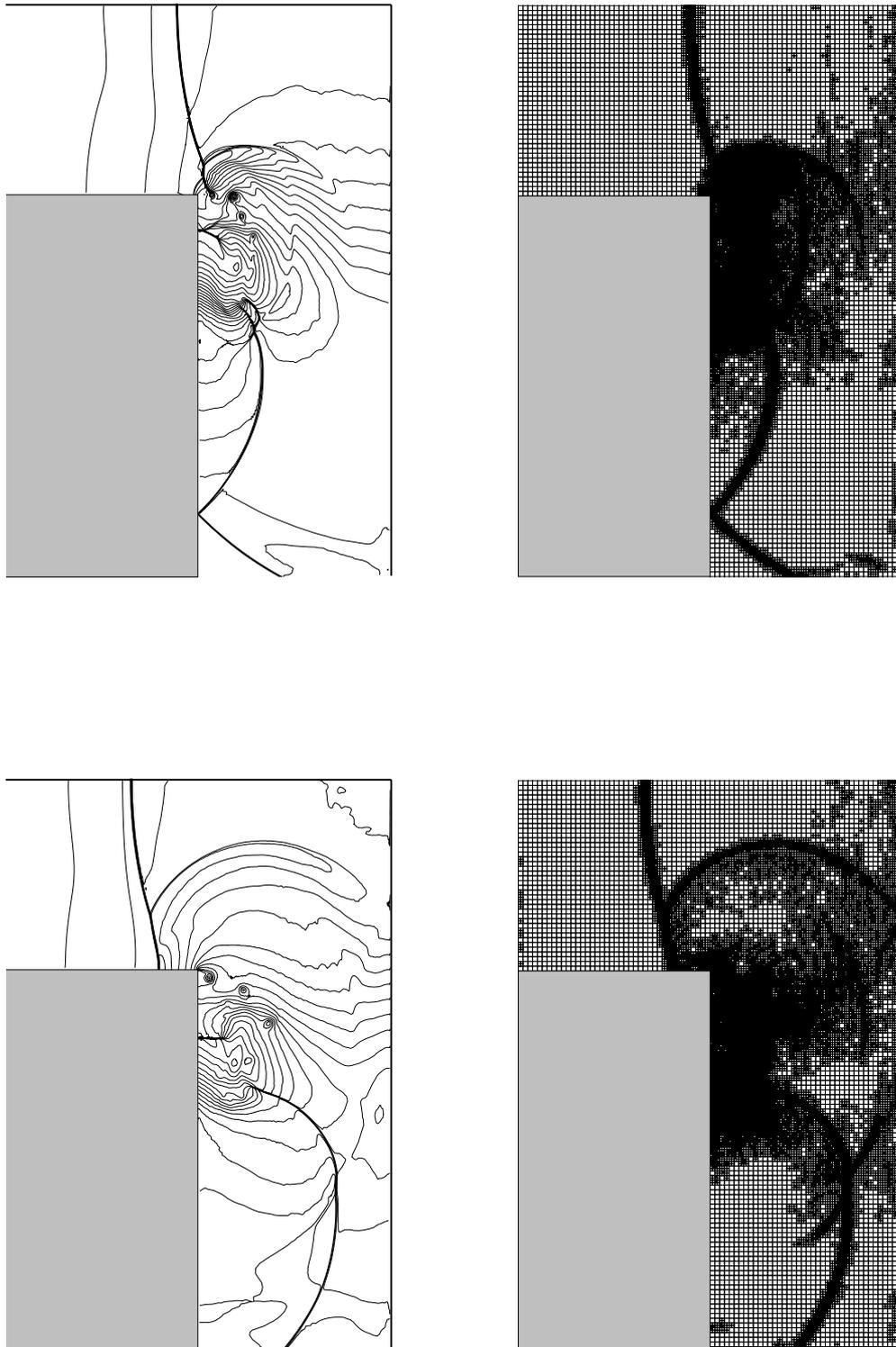


Figure 5.4: Contours of density (left) and the computational mesh (right) for the second interaction of a Mach 1.5 shock wave with a corner vortex. Simulation time $t = 79.382$ (top) and $t = 90.723$ (bottom).

To enable a more detailed analysis of the interaction of the Mach 1.5 shock wave with the corner vortex, density and vorticity contours are plotted for the region just downstream of the corner. Vorticity is calculated as

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}.$$

Visualisation using vorticity allows the various scales of motions to be more easily distinguished since only vortices and curved shocks are resolved in the plots. Eight time instants are shown for computations performed using the initial mesh with 4 levels of refinement in Figure 5.5 and 6 levels of refinement in Figure 5.6 in order to examine the effect of grid resolution on the interaction. The initial frame shows the reflected shock wave approaching the vortex for the first time ($t = 58.592$). Of particular interest is the difference in vorticity within the shear layer between the two grid resolutions. Because the Kelvin-Helmholtz instability responsible for the roll up of the layer is viscosity dependent, the numerical dissipation inherent to the computational scheme plays an important role. For a very fine mesh, the simulation code will produce a very fine shear layer emanating from the corner. Hence, the grid resolution determines the time and length scales that characterise the break-up of the shear layer into discrete vortices.

As the shock wave passes through the vortex ($t = 62.372$ and $t = 66.152$), there is a marked increase in the amount and intensity of the small scale vortical structures within the large corner vortex. This appears to be a result of the interaction between the reflected shock wave and the small vortices within the shear layer. The vorticity is more concentrated and a bifurcated shock structure also results.

At time $t = 69.932$ the reflected shockwave has reflected again, this time from the near wall, and is about to process the vortex. The portion of the shock wave swung around by the vortex has impacted on the shear layer resulting in a complex flow pattern containing several shock waves. At this stage, the large vortex appears to have been stretched slightly by the interaction but is otherwise relatively unaffected. In contrast, the internal structure of the vortex has been altered considerably, comprising several new shock structures, and regions of highly concentrated vorticity.

The second interaction between the shock and the corner vortex ($t = 73.712$ and $t = 77.492$) appears to be much more disruptive. Once again the smaller scale motions

within the vortex are amplified; however, this time the whole vortex is elongated in the downstream direction. The final instants ($t = 81.273$ and $t = 85.053$) show that the vortex continues to be stretched. This elongation is probably due to a mixture of stretching in the downstream direction and compression in the cross-stream direction associated with the shock interaction. The stretching component in the downstream direction appears only to be present during the second interaction with the shock wave. This is possibly due to the fact that the upper portion of the reflected shock wave has started to interact with the shear layer emanating from the corner. Hence the lower part of the corner vortex would become weaker and more easily convected downstream.

Figure 5.7 shows that as observed by Skews [52], the vortex appears to retain its integrity in this case. Whilst the large scale shock vortex interaction did not result in the vortex degrading into small scale motions, interactions on a smaller scale resulted in the generation of much vorticity. Several discrete fine scale vortices embedded around the stretched corner vortex can be seen from the vorticity contours. These are a result of the interaction between the discrete vortices associated with the Kelvin-Helmholtz roll up of the shear layer, and the multiple shock structures produced by the larger shock vortex interaction.

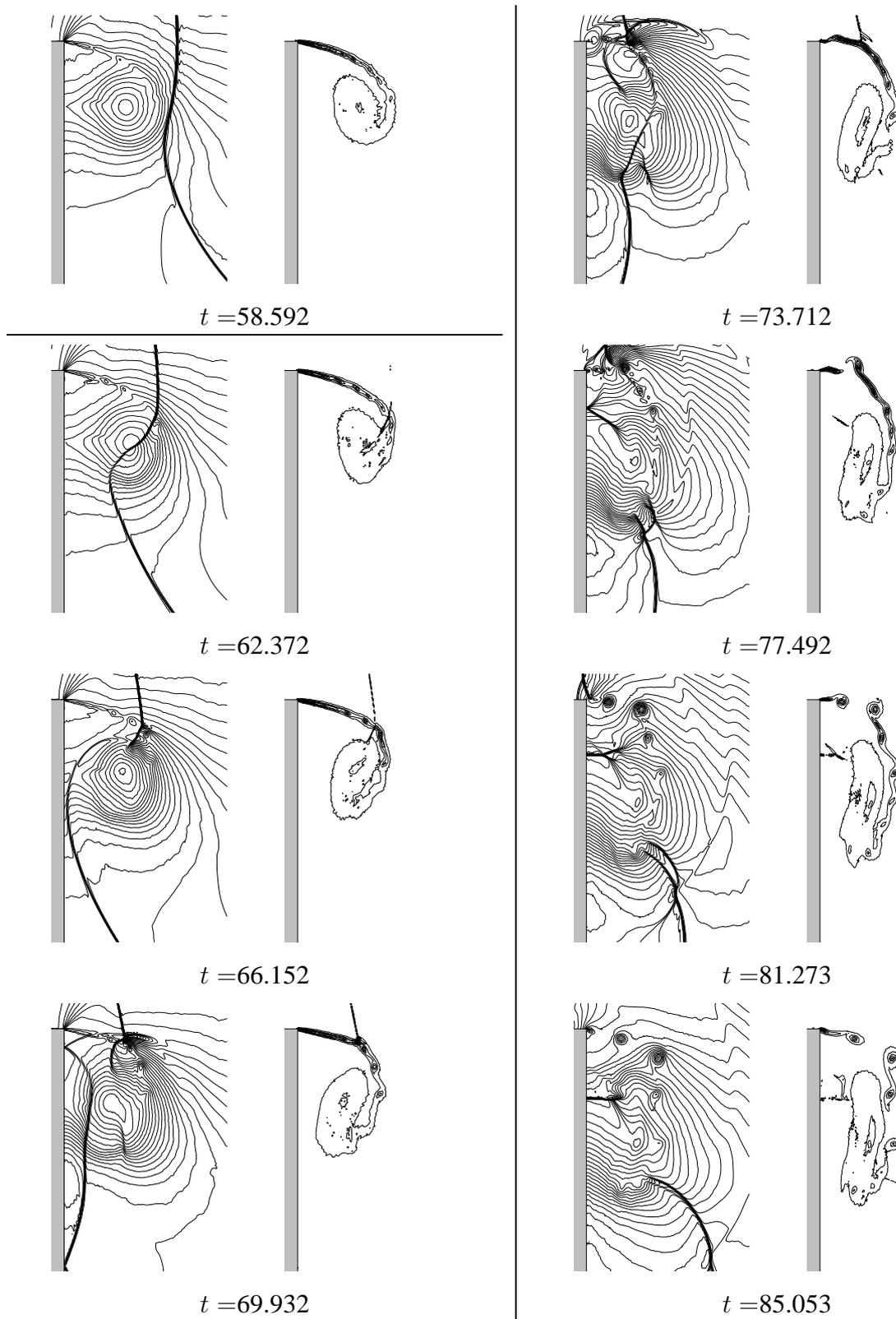


Figure 5.5: Flow development for a Mach 1.5 shock wave using 4 levels of refinement. Contours of density (left) and contours of vorticity (right). Flow progresses from top to bottom, left to right.

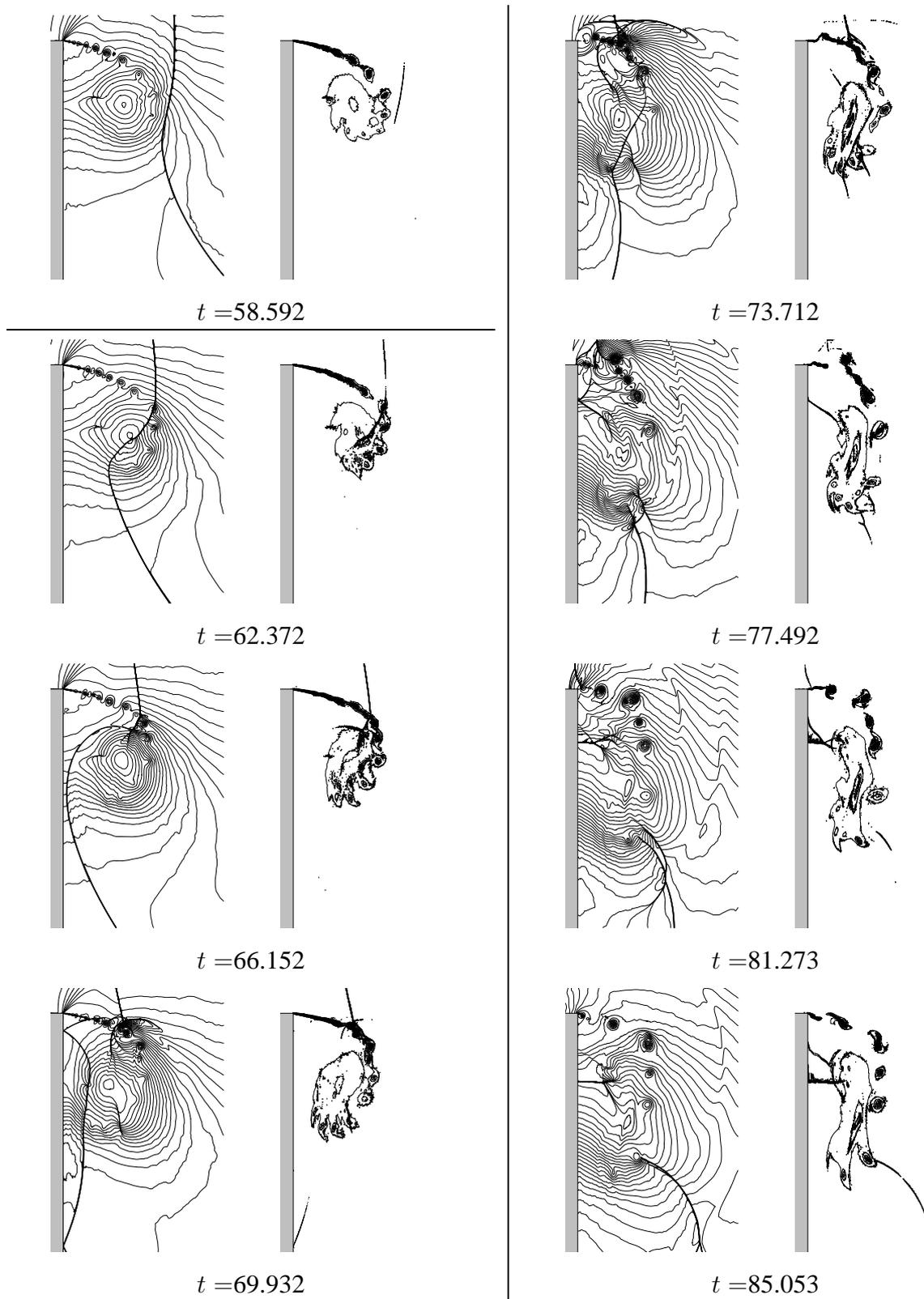


Figure 5.6: Flow development for a Mach 1.5 shock wave using 6 levels of refinement. Contours of density (left) and contours of vorticity (right). Flow progresses from top to bottom, left to right.

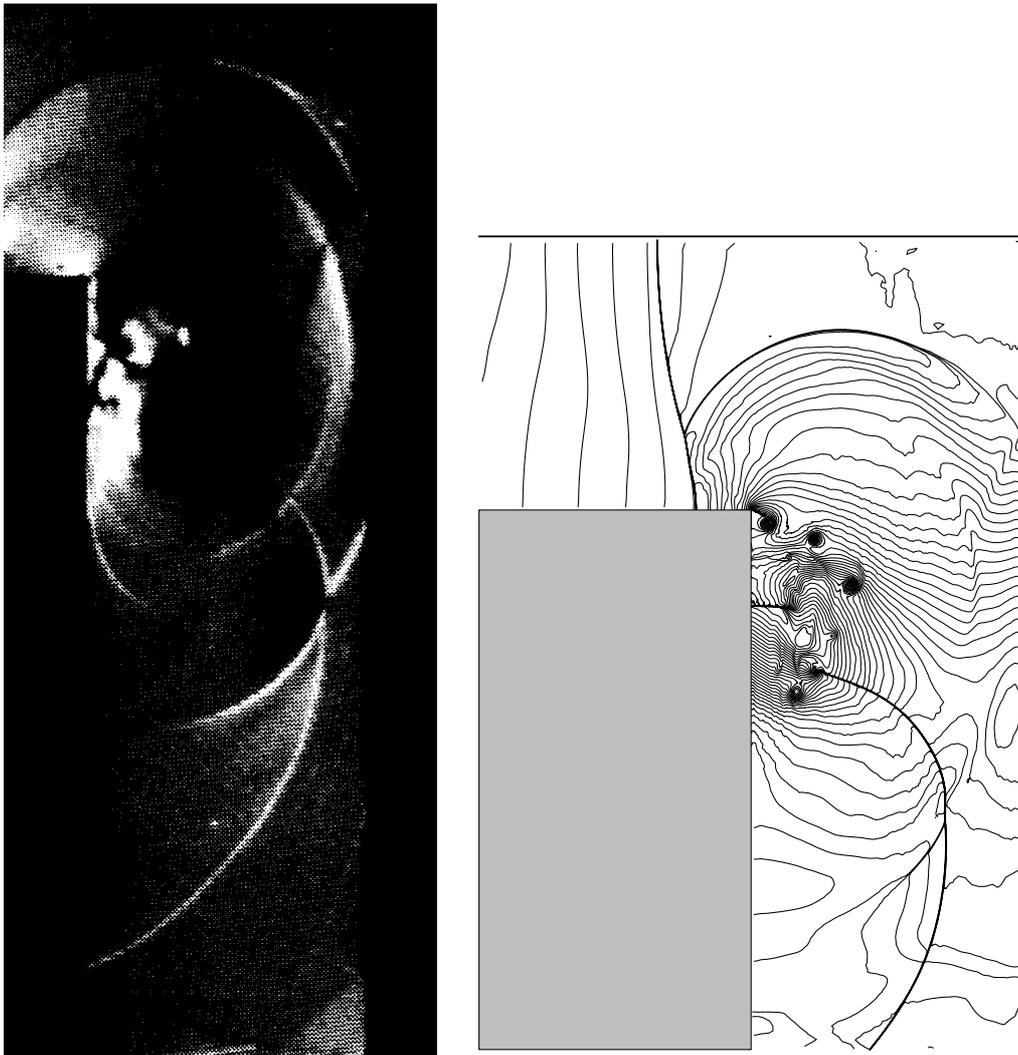


Figure 5.7: Comparison between the experimental Schlieren of Skews [52] (left) and computational density contours (right) for a Mach 1.5 shock wave.

5.3 Interaction with a Mach 2.0 Shock Wave

The interaction of a Mach 2.0 shock wave was investigated numerically subject to the same geometric and computational parameters discussed in the previous section. The characteristic length remained the same, whilst for a Mach 2.0 shock wave the characteristic shock speed is 692 m/s. By non-dimensionalising all times by the characteristic time scale $50.8 \times 10^{-3}/692$ flow instants can be compared with their Mach 1.5 counterparts. Again, the inflow conditions were specified by computing post shock flow conditions determined from the shock Hugoniot. Whilst this represents the exact jump conditions across a Mach 2.0 shock wave, the dissipative nature of the computational technique causes shocks to be smeared over several cells. The difference between the exact and (smeared) computational jump conditions results in the generation of a weak wave. Whilst this wave was insignificant in the previous case, the computational mesh in Figure 5.8 shows that the associated error is now causing slight variations in the post shock flow. Hillier [19] suggests a fix for these starting transients by first running a computational simulation of a shock wave of the desired strength and then specifying shock conditions for the full simulation based on the numerical jump conditions. Because the density contours in Figure 5.8 seem relatively unaffected by this problem, the current technique is deemed satisfactory for this study.

It can be seen from Figure 5.8 that the flow features seem more distinctly defined than for a Mach 1.5 shock wave. Even with four levels of refinement both shocks internal to the large vortex are distinctly visible, as is the contact surface beneath it. In addition to this, the expansion fan emanating from the corner is stronger, and the computational mesh has been refined accordingly in this region.

Once again density and vorticity contours in the region just downstream of the step have been plotted to enable visualisation of the interaction (Figure 5.9). As with the lower Mach number case, the first passage of the shock wave results in the the generation of considerable structure within the large vortex as a result of the interaction with the fine scale discrete vortices associated with the Kelvin-Helmholtz roll up of the shear layer. Several bifurcated shock structures result from this interaction. The large vortex appears to be compressed slightly in the cross stream direction by the first transit of the shock wave. The portion of the shock wave swung around by the vortex hits the slip stream at

time $t = 69.932$. About this time, the vortex begins to stretch in the downstream direction.

The second passage of the shock wave through the vortex results in an increased amount of detail within the large vortex. As well as this, several intersecting shock structures are apparent. These are most probably associated with the interaction between the reflected shock wave and the small scale internal shock and vortex structures. Whilst stretched considerably in the downstream direction, and compressed in the cross stream direction, the vorticity contours show that the vortex appears generally to have retained its integrity. Despite this however, it is evident that the small vortical structures concentrated within the vortex are becoming more dominant over the larger structure. Figure 5.10 shows a comparison between the experimental Schlieren of Skews [52] and computational density contours, after the second transit of the shock wave.

Unfortunately simulation results on a mesh using 6 levels of refinement were not available at the time of submitting this thesis due to the large computational time associated with the highly refined mesh. The earlier Mach 1.5 calculations suggest that the number and intensity of the fine scale vortices increases with grid resolution since the shear layer rolls up into finer structures. It is not unreasonable, then, to predict that higher resolution would result in additional fine scale structures internal to the large vortex. The complexity of the resulting flow field, particularly when viewed using contours of density, would make it difficult to distinguish the large scale vortex from the numerous small scales within it. It would be beneficial to produce numerical Schlieren plots¹ of the final flow state to compare with the experimental Schlieren of Skews. It would be interesting to see to what extent this visualisation technique, which is sensitive to density gradients in one direction, permits distinction between the two scales of motion.

There are two different scales of motions associated with the interaction. The large scale vortex and the reflected shock wave constitute the larger motions present which alone have been shown to interact in a well defined manner [46]. On a finer scale, the discrete vortices associated with the shear layer, and the internal vortex shocks are specific to the creation of a vortex by shock wave diffraction. If a fine enough numerical grid is used, the mutual interaction between these two has been shown to result in a bifurcated shock structure [61]. In any case, whilst the first transit of the shock results in only minor

¹unfortunately this is not possible at the post-processing stage and there was insufficient time to re-run the simulations calculating the required density derivatives.

compression of the large vortex, the interaction with the small scales results in bifurcated shocks and intensified vorticity, the effect being more pronounced for stronger shocks. The interaction between the portion of the shock wave swung around by the vortex and the shear layer, seems to produce a weakening in the large vortex allowing it to be more affected by the second transit of the shock wave; a combination of compression in the cross-stream direction and stretching in the downstream direction results in an elongated vortex. At the same time, the fine scale internal structure comprising shocks and vortices is further intensified by the second transit of the primary wave. These conflicting phenomena, namely the weakening of the large vortex and the grid dependent initiation and intensification of the fine scale internal motions, provides one explanation for the conflicting results obtained during the previous computational study [59]. They found that whilst relatively coarse grid simulations appeared to result in the vortex breaking up after the second transit of the shock wave, finer grids resulted in the vortex appearing to break-up earlier. On very-fine grids, the vortex even broke-up after the first transit of the shock wave.

A viscous model would be needed to determine the exact state of the shear layer emanating from the corner of the step. Without such a simulation, however, one hypothesis for the observations of Skews is that for flow states where the fine scales become stronger, it becomes increasingly more difficult to distinguish the large vortex, particularly by the Schlieren technique.

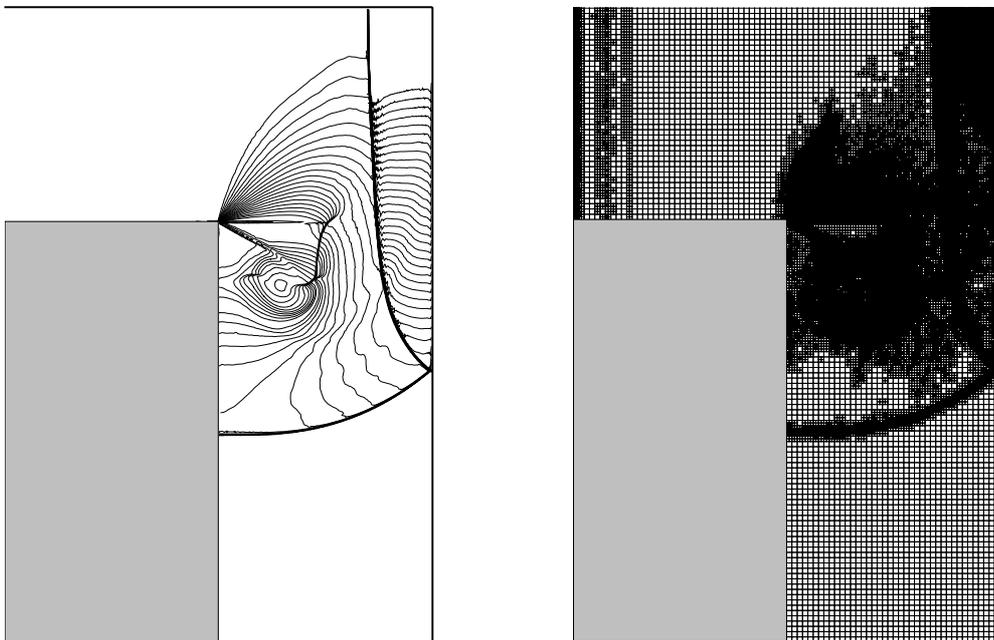


Figure 5.8: Contours of density (left) and the computational mesh (right) for the early stages of the interaction of a Mach 2.0 shock wave with a corner vortex. Simulation time=51.032

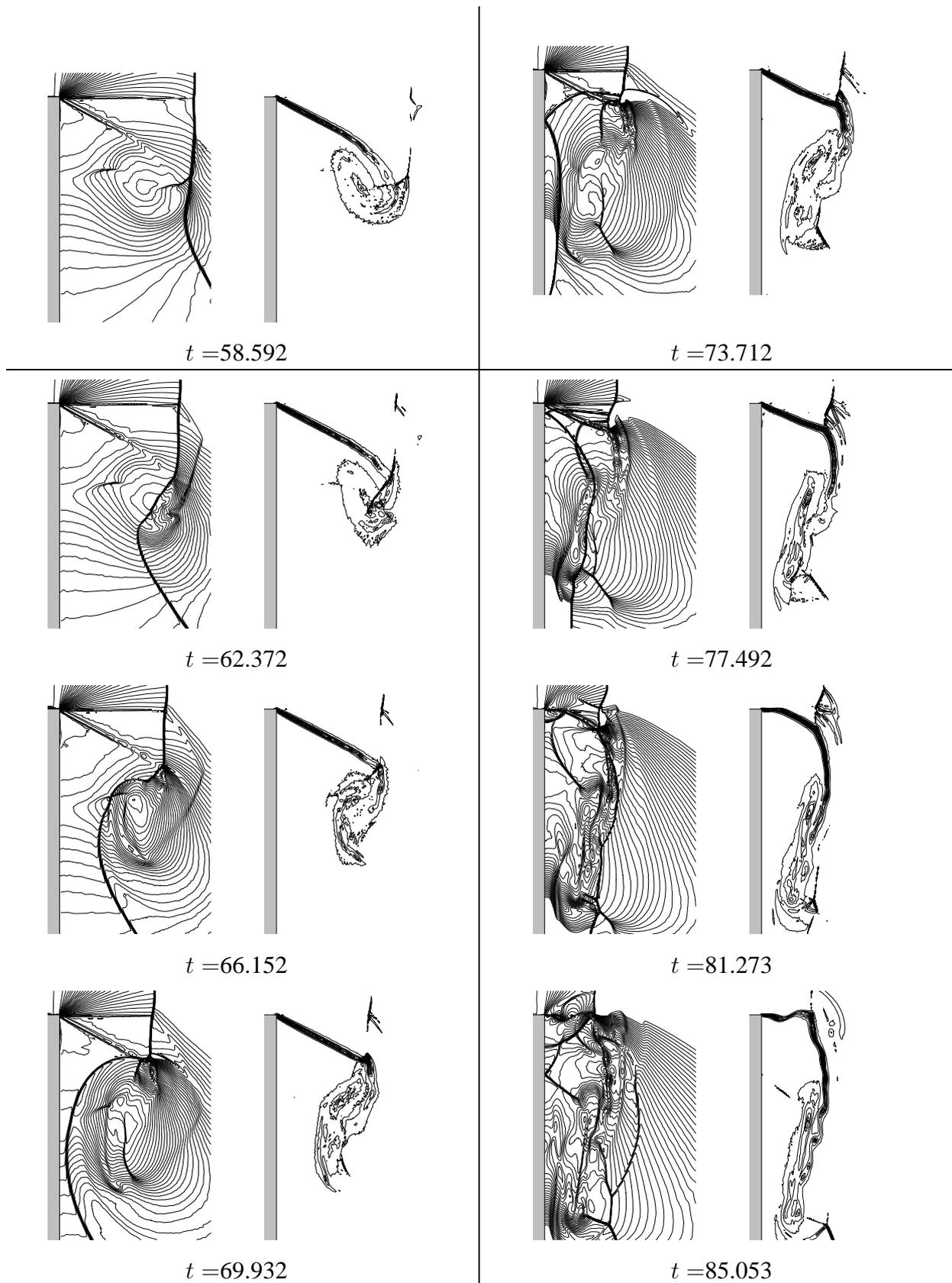


Figure 5.9: Flow development for a Mach 2.0 shock wave using 4 levels of refinement. Contours of density (left) and contours of vorticity (right). Flow progresses from top to bottom, left to right.

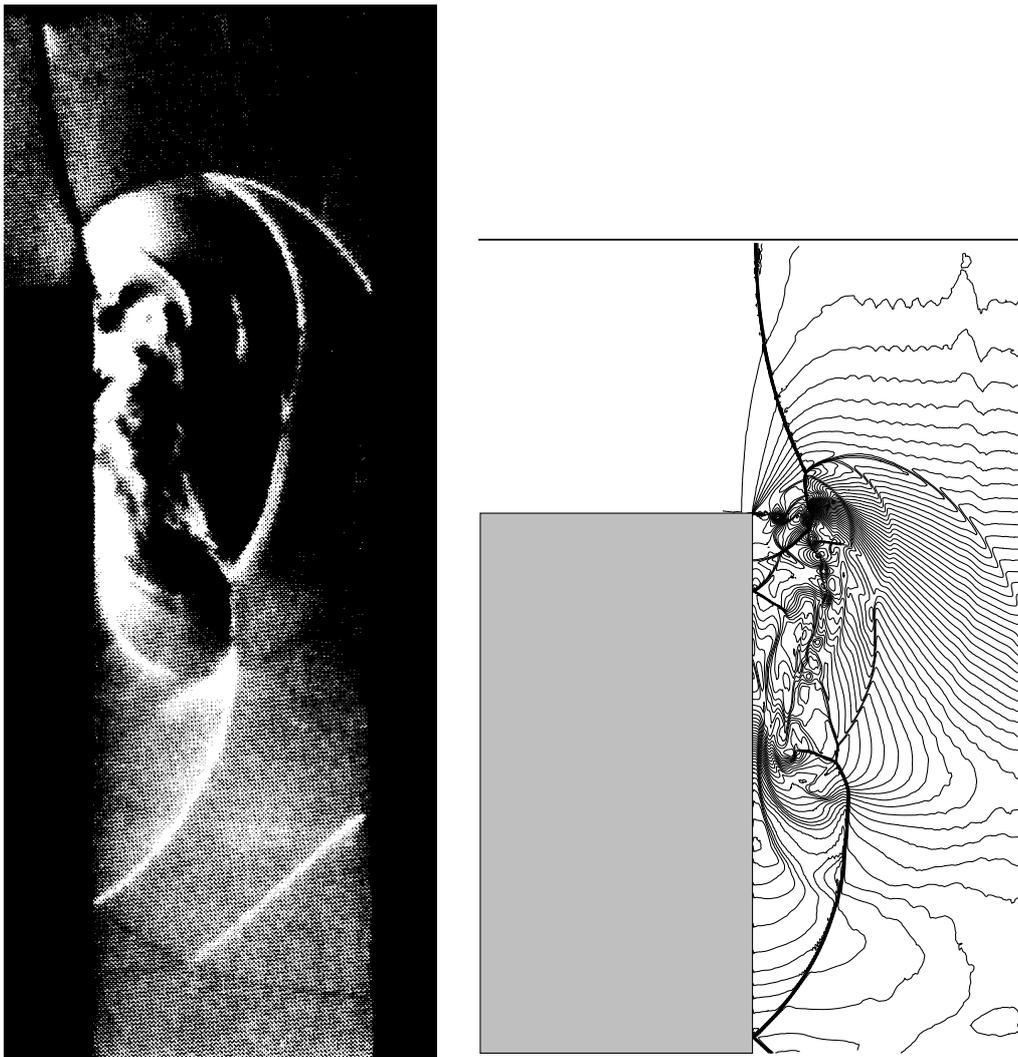


Figure 5.10: Comparison between the experimental Schlieren of Skews [52] (left) and computational density contours (right) for a Mach 2.0 shock wave.

Conclusion and Recommendations

This thesis was primarily concerned with the computational modelling of shock tube phenomena. Whilst direct numerical simulation of such phenomena incorporating complex chemical reactions could, in theory, provide an exact description of the processes taking place, computing power dictates that such a ‘brute force’ technique is currently not feasible nor will it become so in the foreseeable future. A consequence of this is that we are forced to limit the extent of our modelling to either very simplified flow models or detailed simulations of simple flow interactions. There are essentially three sections to this thesis: quasi-one-dimensional modelling of a full shock tube facility, the development of a two-dimensional flow simulation code, and the application of this code to shock wave vortex interactions.

6.1 Quasi-One-Dimensional Modelling

The design and operation of impulse facilities requires an accurate knowledge of the underlying flow processes. Since the flow is predominantly one-dimensional, a good approximation can be obtained in a reasonable time frame by employing a quasi-one-dimensional (q1d) analysis that incorporates approximate models for phenomenon such as viscous losses and heat transfer. Finite time diaphragm rupture is another mechanism that cannot be captured directly from the q1d assumption. In this thesis, two models that aim to capture the macroscopic effect of slowly opening diaphragms were designed. An Eulerian q1d simulation code was formulated and used to incorporate the models.

Whilst both models appear to produce some appropriate changes in the downstream flow properties, they do not capture the full nature of slowly opening diaphragms. It may be that a more detailed analysis of the wave pattern downstream of the driver re-

stricter might help in formulating a model that has a stronger effect than that produced by either of the current models. Ultimately, however, the underlying mechanisms are three-dimensional in nature and an attempt to model the complex interactions in a one-dimensional analysis will always be an approximation.

6.2 Two-Dimensional Flow Simulations

The second part of the thesis concerned the development of a two-dimensional Euler code that included solution adaptive grid refinement. Emphasis was placed on designing a code that could efficiently simulate detailed flow features such as shocks and vortices within simplified geometries; however, a fine computational grid is required to minimise the discretisation error. The distribution of discretisation error across the domain is not uniform, hence, computational effort is wasted in some regions. Compressible flows typically contain sharp transitions between uniform flow regions and, as such, considerable effort would be wasted if a uniform mesh was employed. The simulation code developed here provides the grid resolution where it is needed whilst avoiding the computational expense of using a uniformly fine grid.

The design of an efficient algorithm depends to a large extent on the type of flow problem to be solved. Due to the relatively straightforward geometry considered in this thesis, it was decided that square cells would form the basis for the computational mesh. This offers considerable simplifications to be made to the amount of data stored and to the complexity of the data structure needed to access it because many inter-cell relations are known implicitly. The restriction of the problem to an Euler analysis allows further optimisation since only first order derivatives are required at interfaces; the Navier-Stokes equations include viscous terms, and requires flow information beyond immediate cell neighbours. A modified threaded quad tree data structure was selected that implicitly contains many of the cell-neighbour links needed for the computation of first derivatives. Emphasis was placed on a technique that can refine and unrefine the mesh in a flexible manner so as to capture the many evolving flow features on all scales. It is believed that the subdivision of existing cells achieves this.

A Taylor series expansion approximation to the discretisation error was chosen as the primary technique to mark cells for adaption. This indicator was used in conjunction with

a gradient indicator tuned to detect shock waves.

An explicit time marching technique was used to advance the simulations. It was decided that the stability and resulting speed benefits of an implicit technique would be wasted due to the need for small time-steps to adequately resolve the very small time-scales associated with shock-vortex interactions. A time adaptive strategy was not implemented since the associated ‘speed up’ was considered small in comparison to that obtainable from spatial adaptation. Never the less, a strategy is discussed and its implementation is strongly recommended for future three-dimensional work.

Flow reconstruction was performed using MUSCL like interpolation modified for non-uniform cells, and a van Albada limiter as the base scheme. Considerable problems were encountered when dealing with interfaces between fine and coarse cells. Whilst the modified reconstruction scheme can handle a cell compression ratio of 50%, data retrieval problems were encountered. Whilst the flow state at the fine side of the interface can be calculated in a straightforward manner, the ‘quad’ data structure does not allow the coarse cell’s neighbour to be determined. A linear interpolation technique was applied to determine the coarse interface state and seemed justifiable since coarse cells only exist in regions of low curvature of flow properties. Whilst this worked well for problems that are inherently one-dimensional in nature, it introduced large amounts of numerical noise for flows exhibiting highly two dimensional behaviour. For this reason a first order scheme was used to for the determination of the coarse cell interface state. This data access problem could easily be overcome by storing an additional pointer from each ‘quad’ to the parent cell above it. Such a pointer appears in the current data structures; however, it was only used for debugging purposes and is not updated during application of the adaption routines. Future work concerning flow reconstruction could use this pointer to obtain flow states at any cell in the domain. Whilst this becomes inefficient for locating cells ‘far away’, it would be justifiable for locating coarse cell neighbours as required for consistent high order interpolation. It is expected that this would reduce some of the numerical noise generated in the current version of the code. Additionally, this extra link would allow viscous terms to be incorporated into the code in a straightforward manner.

A series of tests were performed to ensure that the code was solving the Euler equations correctly. Comparison with experimental, theoretical, and a wealth of previous numerical data indicate that the expected performance level was being obtained. The

evolving computational mesh for each simulation was studied and the adaption parameters optimised to ensure computational efficiency. Simulations of the early evolution of a corner vortex verified that the internal vortex structure was being adequately resolved.

The original scope of the (PhD) project included a three-dimensional simulation code. Due to the reduction in scope of the project from a PhD to a Masters thesis this was not possible, however, the code was developed so as to allow extension to three dimensions in a straightforward manner. It is hoped that future work would involve extending the code to three-dimensions where the full benefit of the subdivision adaption technique would be seen. The extensions required to achieve this are discussed in the text where relevant; the most tedious aspect would involve the formulation of three-dimensional ‘look-up’ tables relating a cell to all possible neighbour configurations. The code was also constructed around subroutines that anticipated later parallelisation. Future three-dimensional work should take advantage of this and exploit the benefits of parallel computing.

6.3 Shock Wave Vortex Interactions

The two-dimensional simulation code was applied to the simulation of the interaction between shock waves of varying strength and a corner vortex in a bid to explain the experimental observations of Skews [52]. The very small time scales in which the large corner vortex appeared to degrade into small scale motions hinted at an inviscid mechanism being an important component of the flow dynamics. For this reason the simulation code MACS2d was formulated for the Euler equations and used to simulate the experiments.

Two main scales of motion were identified during the interaction. On a large scale, the reflected shock wave attempts to compress the large corner vortex during its transit. On a smaller scale, the Kelvin-Helmholtz roll up of the shear layer emanating from the corner produces discrete vortices. The time and length scales associated with the formation of these vortices is viscosity dependent. The governing equations are inviscid and hence the numerical viscosity inherent to the scheme determines the thickness of the shear layer and thus the wavelength of rollup.. Internal to the vortex, there exists two small shock structures. The mutual interactions between these structures represents one scale of the ensemble of energy cascade processes inherent to a turbulent flow.

During the first transit of the shock wave the small scale structures are intensified as

they ‘receive’ energy from the larger motions. The large scale vortex remains relatively unaffected by this first interaction; however, since it continues to ‘receive’ vorticity from the shear layer emanating from the corner. A portion of the shock wave is swung around during this first interaction and impacts directly with the shear layer.

The second transit of the shock wave through the vortex produces a further intensification of the small scale motions. This time the large scale vortex is weakened considerably by the energy transfer since the shear layer supplying it has been weakened. The resulting flow field comprises concentrated small scale vortices and bifurcated shock structures as well as a weakened primary vortex. The scale which appears dominant in the final flow field depends on the shock wave strength and intensity of the initial small scale vortices. If density contours are used to visualise the flow it is difficult to determine the large scale structure after the second transit of the reflected Mach 2.0 shock wave; however, vorticity contours show that it is still present.

One would expect initial small scale motions to become more concentrated for higher incident shock speeds as well as larger bend angles. Additionally, a consequence of the grid dependent nature of the shear layer roll up is that finer meshes will begin with more concentrated small scale motions. This makes the idea of energy transfer between the scales consistent with the experimental observations and with the grid convergence problems experienced by the previous computational study [59].

With this in mind there are several areas for further work to be done. The computation of numerical Schlieren would allow a greater understanding of how the experimental visualisation distinguishes between the two scales of motion. Unfortunately this is not possible at the post processor stage due to deficiencies in the commercial software that was available to the author. Calculation of density derivatives within the simulation code could be implemented in the same way that vorticity is calculated. Whilst this is a straightforward task, the simulations would need to be rerun. Unfortunately this is not possible in the present time frames. The grid dependent nature of the initial fine scale vorticity can only be resolved by incorporating a physical viscosity. Only by implementing the Navier-Stokes equations will the correct amount of vorticity be predicted within the shear layer; it may be that coarse grid Euler calculations are more realistic than fine grid ones with respect to this phenomenon; however, this remains speculation.

Finally, turbulence is a three dimensional phenomenon in which vortical structures can be aligned in any direction. The two-dimensional analysis restricts the flow field to vorticity aligned with the z axis. Perturbations in the third dimension would cause vortical structures of differing orientations to be present. These structures would be involved in the energy cascade process and so draw energy from interactions with the larger scales. Extension of the code to three-dimensions has already been discussed and with current high performance computers, it is feasible that results could be obtained in a reasonable time frame. Future work in this area should make use of advanced post processing tools and three-dimensional visualisation techniques.

Bibliography

- [1] Berger, M. J., "Data structures for adaptive grid generation," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 904–916.
- [2] Berger, M. J. and Colella, P., "Local adaptive mesh refinement for shock hydrodynamics," *Journal of Computational Physics*, Vol. 82, 1989, pp. 64–84.
- [3] Bockelie, M. J. and Eiseman, P. R., "A time-accurate adaptive grid method and the numerical simulation of a shock-vortex interaction," Technical paper, NASA, 1990.
- [4] Chiang, y., van Leer, B., and Powell, K. G., "Simulation of unsteady inviscid flow on an adaptively refined cartesian grid," *A.I.A.A. Journal*, 1992.
- [5] Colella, P. and Woodward, P. R., "The piecewise parabolic method (PPM) for gas-dynamical simulations," *Journal of Computational Physics*, Vol. 54, 1984, pp. 174–201.
- [6] Connet, W. C., Agarwal, R. K., and Schwartz, A. L., "An adaptive-grid algorithm for the Euler/Navier-Stokes equations," *A.I.A.A. Journal*, 1987.
- [7] De Zeeuw, D. and Powell, K. G., "An adaptively refined cartesian mesh solver for the euler equations," *Journal of Computational Physics*, Vol. 104, 1993, pp. 56–68.
- [8] Doolan, C. J. and Jacobs, P. A., "Modeling mass entrainment in a quasi-one-dimensional shock tube code." *A.I.A.A. Journal*, Vol. 34, No. 6, 1996, pp. 1291–1293.
- [9] Dukowicz, J. K., "Conservative rezoning (remapping) for general quadrilateral meshes," *Journal of Computational Physics*, Vol. 54, 1984, pp. 411–424.

-
- [10] Edwards, M., Oden, T., and Demkowicz, L., "An h-r adaptive approximate riemann solver for the euler equations in two dimensions," *SIAM Journal on Scientific and Statistical Computing*, Vol. 14, No. 1, 1993, pp. 185–217.
- [11] Emery, A. F., "An evaluation of several differencing methods for inviscid fluid flow problems," *Journal of Computational Physics*, Vol. 2, 1968, pp. 306–331.
- [12] Falcovitz, J. and Birman, A., "A singularities tracking conservation laws scheme for compressible duct flows," *Journal of Computational Physics*, Vol. 115, 1994, pp. 431–439.
- [13] Falcovitz, J. and Takayama, K., "Vortex generation in a partitioned shock tube," *Proceedings of the second international workshop on shock wave/vortex interactions*, October 6-8 Sendai 1997.
- [14] Fox, R. W. and McDonald, A. T., *Fluid Mechanics*, John Wiley and sons, 1994.
- [15] Gaitonde, A. L. and Fiddes, S. P., "A three-dimensional moving mesh method for the calculation of unsteady transonic flows," *The Aeronautical Journal*, 1995, pp. 150–160.
- [16] Godunov, S. K., "Reminiscences about difference schemes," *Journal of Computational Physics*, Vol. 153, 1999, pp. 6–25.
- [17] Goozee, R. and Jacobs, P. A., "Parallel computing for compressible flow cfd. 1. survey of hardware and software." Departmental report, University of Queensland, 2000.
- [18] Groth, C. P. T., Gottlieb, J. J., and Sullivan, P. A., "Numerical investigation of high-temperature effects in the UTIAS-RPI hypersonic impulse tunnel." *Canadian Journal of Physics*, Vol. 69, No. 7, 1991, pp. 897–918.
- [19] Hillier, R., "Computation of shock wave diffraction at a ninety degree convex edge," *Shock waves*, Vol. 1, 1991, pp. 89–98.
- [20] Ikui, T. and Matsuo, K., "Investigations of the aerodynamic characteristics of the shock tubes," *Bulletin of JSME*, Vol. 12, No. 52, 1969, pp. 774–792.

- [21] Itoh, K., S., U., Komuro, T., Sato, K., Takahashi, M., Miyajima, H., Tanno, H., and Muramoto, H., "Improvement of a free piston driver for a high-enthalpy shock tunnel," *Shock Waves*, Vol. 8, 1998, pp. 215–233.
- [22] Itoh, K., Tani, K., Tanno, H., Takahashi, M., Miyajima, H., Asano, T., Saso, A., and Takayama, K., "A numerical and experimental study of free piston shock tunnel," *19th International Symposium on Shock Waves, Marseille, France.*, Springer-Verlag, 1993, pp. 257–262.
- [23] Jacobs, P. A., "Single-block Navier-Stokes integrator." ICASE Interim Report 18, 1991.
- [24] Jacobs, P. A., "An approximate Riemann solver for hypervelocity flows." *A.I.A.A. Journal*, Vol. 30, No. 10, 1992, pp. 2558–2561.
- [25] Jacobs, P. A., "Quasi-one-dimensional modelling of a free-piston shock tunnel." *A.I.A.A. Journal*, Vol. 32, No. 1, 1994, pp. 137–145.
- [26] Jacobs, P. A., "Shock tube modelling with L1d." Department of Mechanical Engineering Report 13/98, The University of Queensland, Brisbane, Australia., November 1998.
- [27] Jacobs, P. A., "L1d: a computer program for the simulation of transient-flow facilities." Department of Mechanical Engineering Report 1/99, The University of Queensland, Brisbane, Australia., January 1999.
- [28] Jacobs, P. A., Morgan, R. G., Stalker, R. J., and Mee, D. J., "Use of argon-helium driver-gas mixtures in the T4 shock tunnel." *19th International Symposium on Shock Waves, Marseille, France.*, Springer-Verlag, 1993.
- [29] Johnston, I. A., *Simulation of flow around hypersonic blunt-nosed vehicles for the calibration of air data systems*, Ph.D. thesis, The University of Queensland, June 1999.
- [30] Kallinderis, Y. and Vidwans, A., "Generic parallel adaptive-grid navier stokes algorithm," *A.I.A.A. Journal*, Vol. 32, No. 1, 1994, pp. 54–61.

- [31] Kaufmann III, W. J. and Smarr, L. L., *Supercomputing and the transformation of science*, Scientific American Library, 1993.
- [32] Khokhlov, A. M., "Fully threaded tree-algorithms for adaptive refinement fluid simulations," *Journal of Computational Physics*, Vol. 143, 1998, pp. 519–543.
- [33] Korst, H. H., "A theory for base pressures in transonic and supersonic flow," *Journal of Applied Mechanics*, 1956, pp. 593–600.
- [34] Lappas, T., Leonard, A., and Dimotakis, P. E., "An adaptive Lagrangian method for computing 1D reacting and non-reacting flows," *Journal of Computational Physics*, Vol. 104, 1993, pp. 361–376.
- [35] Lee, J. H. S., "Shock/vortex interactions: its role in compressible turbulence and detonation structure," *Proceedings of the second international workshop on shock wave/vortex interactions*, October 6-8 Sendai 1997.
- [36] Matsuura, K., Fujita, K., K., F., and Abe, T., "Interaction of a corner vortex with a reflected shock wave," *21st international symposium on shock waves*, July 20-25 Great Keppel Island 1997.
- [37] Maus, J. R. and Rock, S. G., "Computational results for AEDC free-piston shock tunnel and comparison with calibration data," *A.I.A.A. Journal*, Vol. 32, No. 1, January 1995, pp. 137–145.
- [38] Minota, T., Kishige, H., and Nishida, M., "Shock/vortex ring interaction: effects of initial shock strength," *Proceedings of the Second International Workshop on Shock Wave/Vortex Interactions*, October, 6-8 Sendai 1997.
- [39] Misra, A. and Pullin, D. I., "A vortex based subgrid stress model for large-eddy simulation," *The Physics of Fluids*, Vol. 9, No. 8, August 1997, pp. 2443–2454.
- [40] Mitsuda, M., Oda, T., Kurosaka, T., Wakuri, S., and T., A., "One dimensional simulation of free piston shock tunnel/expansion tube," *Proceedings of the 19th International Symposium on Shock Waves, Marseille, France.*, Springer-Verlag, 1993, pp. 457–463.

- [41] Murman, M. M. and Cole, J. D., "Calculation of plane steady transonic flow," *A.I.A.A. Journal*, Vol. 9, No. 1, 1971, pp. 114–121.
- [42] Petrie, P. J. and Jacobs, P. A., "Numerical simulation of diaphragm rupture." *20th International Symposium on Shock Waves*, California Institute of Technology, Pasadena, July 1995.
- [43] Petrie-Repar, P., *Numerical Simulation of Diaphragm Rupture*, Ph.D. thesis, The University of Queensland, January 1997.
- [44] Piomelli, U., "Large-eddy simulation: achievements and challenges," *Progress in aerospace sciences*, Vol. 35, 1999, pp. 335–362.
- [45] Quirk, J. J., "A parallel adaptive grid algorithm for computational shock hydrodynamics," *Applied Numerical Mathematics*, Vol. 20, 1996, pp. 427–453.
- [46] Ribner, H. S., "Cylindrical sound wave generated by shock-vortex interaction," *A.I.A.A. Journal*, Vol. 23, No. 11, 1985, pp. 1708–1715.
- [47] Roberts, G. T. and East, R. A., "Shock tunnel operation with driver restrictor," *Proceedings of the 16th International Symposium on Shock Tubes and Waves, Aachen, West Germany*, 1987, pp. 629–635.
- [48] Rothkopf, E. M. and Low, W., "Diaphragm opening process in shock tubes," *The Physics of Fluids*, Vol. 19, No. 12, 1974, pp. 1169–73.
- [49] Salas, M. D., "Shock wave interaction with an abrupt area change," *NASA Technical Paper*, Vol. Paper 3113, 1991.
- [50] Sasoh, A., Ohnishi, Y., Koremoto, K., and Takayama, K., "Operation design and performance of a free piston-driven expansion tube," AIAA Paper 99-0825, 1999.
- [51] Schmidt, G. H. and Jacobs, F. J., "Adaptive local grid refinement and multi-grid in numerical reservoir simulation," *Journal of Computational Physics*, Vol. 77, 1988, pp. 140–165.
- [52] Skews, B. W., "Shock/vortex interactions in shock wave propagation around sharp-angled bends," *Proceedings of the Second International Workshop on Shock Wave/Vortex Interactions*, October, 6-8 Sendai 1997.

- [53] Skews, B. W., "An experimental study of the interaction of shock waves with bends in a duct," *Symposium on internal flows*, paper29, 1971.
- [54] Skews, B. W. and Barbosa, F. J., "Experimental investigations of shock wave/vortex interactions in a bifurcated tube," *Proceedings of the Second International Workshop on Shock Wave/Vortex Interactions*, October, 6-8 Sendai 1997.
- [55] Sod, G. A., "A survey of several finite difference methods for systems of non-linear hyperbolic conservation laws," *Journal of Computational Physics*, Vol. 27, 1978, pp. 1–31.
- [56] Stalker, R. J., "A study of the free-piston shock tunnel." *A.I.A.A. Journal*, Vol. 5, No. 12, 1967, pp. 2160–2165.
- [57] Sun, M. and Takayama, K., "Conservative smoothing on an adaptive quadrilateral grid," *Journal of Computational Physics*, Vol. 150, 1999, pp. 143–180.
- [58] Takayama, T. and Inoue, O., "Shock wave diffraction over a 90 degree sharp corner-posters presented at 18th issw," *Shock Waves*, , No. 1, 1991, pp. 301–312.
- [59] Timofeev, E. and Takayama, K., "Shock wave-vortex interactions in channel bends," *Presented at the third international workshop on shock wave/vortex interactions*, September 12-14 Hakone 1999.
- [60] Uchiyama, N. and Inoue, O., "On the performance of adaptive mesh refinement computation," *Shock Waves*, Vol. 2, 1992, pp. 117–120.
- [61] Uchiyama, N. and Inoue, O., "Shock wave/vortex interaction in a flow over a 90-degree sharp corner," *A.I.A.A. Journal*, Vol. 33, No. 9, 1995, pp. 1740–1742.
- [62] van Albada, G. D., van Leer, B., and Roberts, W. W., "A comparative study of computational methods in cosmic gas dynamics," *Astronomy and Astrophysics*, Vol. 108, 1982, pp. 76–84.
- [63] van Leer, B., "Towards the ultimate conservative finite difference scheme. V a second order sequel to Gudonov's method," *Journal of Computational Physics*, Vol. 32, 1979, pp. 101–136.

- [64] Venkatakrishnan, V., “Convergence to steady state solutions of the Euler equations on unstructured grids with limiters,” *Journal of Computational Physics*, Vol. 118, 1995, pp. 120–130.
- [65] Voinovich, P. and Timofeev, E., “On the formation and development of a toroidal vortex and a highly curved mach disk in the course of shock wave/axis-of-symmetry interaction,” *Proceedings of the second international workshop on shock wave/ vortex interaction*, 1997, pp. 112–119.
- [66] Weeks, T. M. and Dosanjh, D. S., “Sound generation by shock vortex interaction,” *A.I.A.A. Journal*, 1967, pp. 660–669.
- [67] White, D. R., “Influence of diaphragm opening time on shock-tube flows,” *Journal of Fluid Mechanics*, Vol. 4, No. 6, 1958, pp. 585–599.
- [68] Woodward, P. and Colella, P., “The numerical simulation of two-dimensional fluid flow with strong shocks,” *Journal of Computational Physics*, Vol. 54, 1984, pp. 115–173.
- [69] Yamaki, Y. and Rooker, J. R., “Experimental investigation of circular, flat, grooved, and plain steel diaphragms bursting into a 30.5-centimeter-square section,” *NASA Technical Paper*, Vol. TM X-2549, 1972.
- [70] Yih, C., *Fluid Mechanics*, West River Press, 1979.
- [71] Garnier, E., Sagaut, P., and Deville, M., “Large-eddy simulation of shock/homogenous turbulence interaction,” *Direct and large-eddy simulation 3*, 1999.
- [72] Lee, S., Lele, S. K., and Moin, P., “Interaction of isotropic turbulence with shock waves,” *journal of fluid mechanics*, Vol. 340, 1997, pp. 225–247.