# Simulation of Flow Around Hypersonic Blunt-Nosed Vehicles for the Calibration of Air Data Systems

Ian Alexander Johnston, B.E. (Hons)

Department of Mechanical Engineering
The University of Queensland

A thesis submitted for the degree of Doctor of Philosophy
January 1999

# Abstract

Control of a hypersonic vehicle in flight requires knowledge of the vehicle state to sufficient accuracy. Particularly important state data are the air data which describe the ambient atmosphere, and its interaction with the moving vehicle. Such information includes vehicle angle of attack, angle of sideslip, and dynamic pressure. Flush air data systems are commonly installed on blunt-nosed hypersonic vehicles to infer air data through consideration of pressure measurements made at the vehicle surface. An accurate pressure model, describing the relationship between air data parameters and surface pressure distribution, is required for such inferences to be made.

In this work, a new computational fluid dynamics code is developed for the primary purpose of calibrating surface pressure models for use with flush air data systems. In particular, the hypersonic flight experiment (HYFLEX) vehicle is used as a case study. When tested with flight data, results from the calibrated HYFLEX flush air data system exhibit approximately doubled accuracy in estimating vehicle angle of attack and dynamic pressure, compared to an uncalibrated system. Also, the numerical calibration procedure has accuracy and efficiency advantages, compared to traditional, experiment-based calibrations.

The selection and development of algorithms for use in the computational fluid dynamics code are described in detail in the thesis. Particular attention is paid to assessing the suitability of different numerical approaches and models for simulating the flow pressure field around blunt bodies at high speed. A largely original shock fitting formulation is used to improve the accuracy and robustness of the computational fluid dynamics program. The advection upwind splitting method, combining difference and vector splitting, is found to be a good technique for computing fluxes in blunt-body flow simulations. Also, a new modification to the van Albada limiter and monotone upwind scheme for conservation laws is used to provide accurate solution reconstruction. A range of validation and verification test cases are presented, to ensure that simulation results are credible. Some common failings of numerical simulation techniques are investigated, and suppressed or avoided. Such failings include excessive numerical dissipation, shock instability, shock smearing, spurious numerical noise and oscillation, and odd-even decoupling.

# Originality

The work presented in this thesis is, to the best of the author's knowledge and belief, original and the author's own, except as acknowledged by reference or otherwise. None of the material contained in this thesis has previously been submitted, either in whole or in part, for a degree at the University of Queensland or any other institution.

All of the computational fluid dynamics and air data system computer code referred to in this thesis is original and was written by the thesis author, except as acknowledged.

Ian Johnston.

# Related Publications


Johnston, I. A. and McGhee, A., "A CFD Study of Flight in a Martian Atmosphere," *International Aerospace Congress 1995 Proceedings, PICAST 2 – AAC 6*, Vol. 1, The Institution of Engineers, Australia, Melbourne, Mar. 1995, pp. 361–365.

Johnston, I. A. and Jacobs, P. A., "Hypersonic Blunt Body Flows in Reacting Carbon Dioxide," *Twelfth Australasian Fluid Mechanics Conference*, edited by R. W. Bilger, Vol. 2, Sydney, Dec. 1995, pp. 807–810.

Johnston, I. A. and Jacobs, P. A., "SF2D: A Shock Fitting and Capturing Solver for Two Dimensional Compressible Flows," Technical Report 6/96, Department of Mechanical Engineering, University of Queensland, Jul. 1996.

Smith, A. L., Johnston, I. A., and Austin, K. J., "Comparison of Numerical and Experimental Drag Measurement in Hypervelocity Flow," *The Aeronautical Journal*, Vol. 100, No. 999, Nov. 1996, pp. 385–388.

Johnston, I. A., Tuttle, S. L., Jacobs, P. A., and Shimoda, T., "The Numerical and Experimental Simulation of Flow Around the HYFLEX Vehicle Forebody," *Proceedings of the 21st International Symposium on Shock Waves*, edited by A. Houwing, Vol. 1, Canberra, 1997, pp. 453–511.

Johnston, I. A., "An Air Data System Simulator," Technical Report 8/97, Department of Mechanical Engineering, University of Queensland, Aug. 1997.

Johnston, I.A., Jacobs, P. A., and Shimoda, T., "A Study of Flush Air Data System Calibration Using Numerical Simulation," *A Collection of Technical Papers, AIAA 8th International Space Planes and Hypersonic Systems and Technologies Conference*, Norfolk, Apr. 1998, pp. 532–543.

Johnston, I.A., Jacobs, P. A., and Shimoda, T., “Flush Air Data System Calibration Using Numerical Simulation,” *Journal of Spacecraft and Rockets*, Vol. 35, No. 6, Nov.–Dec. 1998, pp. 812–820.

Johnston, I. A., Tuttle, S. L., Jacobs, P. A., and Shimoda, T., “The Numerical and Experimental Simulation of Flow Around the HYFLEX Vehicle Forebody,” *Shock Waves*, to appear in Vol. 9, No. 1, 1999.

# Acknowledgements

To my friend and thesis supervisor Peter Jacobs, I extend thanks for guidance, encouragement, and the benefit of an uncommonly broad span of knowledge. Peter told me that a Doctor of Philosophy degree was an academic apprenticeship; I am pleased to have learned the trade from him.

The partnership of numericist and experimentalist yields results and insight which surpass any that could be achieved in isolation or competition. A number of experimentalists conducted work in parallel to this thesis, and I would like to express my appreciation to them. Sean Tuttle conducted the HYFLEX shock tunnel tests, and contributed to the analysis presented in Section 6.5. Adrian Smith performed blunt-body drag experiments in the expansion tube, which are used to assist validation in Section 5.9. The University of Queensland laser physics group produced interferogram and shadowgraph images of some of the experiments.

For fruitful discussions, the elucidation and criticism of new and old ideas, and plenty of conversation far removed from fluid dynamics, I sincerely thank the past and present postgraduates in our research group: Kevin Austin, Chris Craddock, Andrew McGhee, and Paul Petrie-Repar. I am also grateful for input from Klaus Hannemann, David Mee, and the Centre for Hypersonics research group.

The Australian government afforded me financial support through an Australian postgraduate award and a Department of Mechanical Engineering scholarship, without which this research would not have been completed. Likewise, the University of Queensland High Performance Computing Unit is thanked for making their supercomputing facilities available. The Japanese space agencies NAL and NASDA graciously supplied the HYFLEX flight data.

I am indebted to my family, for their ever present support from thesis inception until the last sentence was penned; my father is surely the greatest engineer of them all. To my friends, including fellow postgraduates, housemates, colleagues, and collegiates, thanks for the laughs and good times that never seemed to end. It's been fun.

I. A. J.

# Contents

# List of Figures

# Nomenclature

## Symbols

| | |
|---|---|
| $a$ | Speed of sound, m/s |
| $A$ | Area, $m^2$; Shock stability variable |
| $A_{i,j}$ | Thermodynamic curve fit coefficient |
| $\mathbf{A}$ | Vector area, $m^2$; Design matrix |
| $b$ | Min-mod biasing parameter |
| $\mathbf{b}$ | Residual vector |
| $B$ | Total variation bound |
| $c_p$ | Specific heat at constant pressure, J/kg K |
| $c_v$ | Specific heat at constant volume, J/kg K |
| $C$ | Species mass fraction; CFL number |
| $\mathbf{C}$ | Vector of species mass fractions |
| $e$ | Intensive internal energy, J/kg |
| $E$ | Intensive total energy, J/kg; Residual pressure error, Pa |
| $f$ | Pressure correction function |
| $F$ | Force, N; Fringe shift; Generalized surface pressure model, Pa |
| $\mathbf{F}$ | Vector of fluxes; Force vector, N |
| $g$ | A component of $f$ |
| $G$ | Mass flow rate per unit area, kg/$m^2$ s |
| $\Delta G_r$ | Gibb's free energy of reaction, for reaction $r$, J/mol |
| $h$ | Cell width, m; Intensive enthalpy, J/kg |
| $h^{\mathrm{f}}$ | Intensive enthalpy of formation, J/kg |
| $\mathbf{h}$ | Vector spanning a cell, m |
| $H$ | Enthalpy flux, N/m s |
| $i$ | Generic counter; Grid index; Pressure port index |
| $\hat{\mathbf{i}}$ | Basis vector in the $x$ direction |
| $I$ | Intensity |
| $j$ | Generic counter; Grid index |

| | |
|---|---|
| $\hat{\mathbf{j}}$ | Basis vector in the $y$ direction |
| $k$ | Grid index; Boltzmann's constant, $1.38 \times 10^{-23}$ J/K; Thermal conductivity, W/m K |
| $k_{b,r}$ | Backward reaction rate for reaction $r$ |
| $k_{f,r}$ | Forward reaction rate for reaction $r$ |
| $\hat{\mathbf{k}}$ | Basis vector in the $z$ direction |
| $K$ | Gladstone-Dale constant, $m^3/kg$; Shock detection threshold |
| $K_{eq,r}$ | Equilibrium constant for reaction $r$ |
| Kn | Knudsen number |
| $L$ | Characteristic body dimension, m; Momentum flux, Pa; Norm |
| $\mathbf{L}$ | Momentum flux vector, Pa |
| $m$ | Min-mod function; Molecular mass, kg |
| $M$ | Mach number; Molecular mass per mole, kg/mol |
| $n$ | Number density, $1/m^3$; Timestep; Normal vector component |
| $\hat{\mathbf{n}}$ | Unit normal vector |
| $NR$ | Number of chemical reactions |
| $NS$ | Number of chemical species |
| $p$ | Grid convergence order |
| $P$ | Pressure, Pa |
| $P_{atm}$ | Standard atmospheric pressure, 101.3 kPa |
| Pr | Prandtl number |
| $q$ | Dynamic pressure, Pa; Heat flux, $W/m^2$ |
| $\mathbf{q}$ | Heat flux vector, $W/m^2$; Air data state vector |
| $Q$ | Generalized flow variable |
| $\mathbf{Q}$ | Vector of source terms |
| $r$ | Radius, m; Ratio of differences; Reaction number |
| $R$ | Gas constant, J/kg K; Pressure ratio |
| $R$ | Universal gas constant, 8.31435 J/mol K |
| $s$ | Intensive entropy, J/kg K; van Albada limiter term |
| $s^c$ | Entropy constant of integration, J/kg K |
| $S$ | Surface area, $m^2$ |
| $t$ | Time, s |
| $t_c$ | Characteristic time, s |
| $\hat{\mathbf{t}}$ | Unit tangent vector |
| $T$ | Temperature, K |
| $\mathbf{T}$ | Matrix of viscous stresses, Pa |
| $u$ | Velocity, m/s |

| | |
|---|---|
| $\mathbf{u}$ | Velocity vector, m/s |
| $\mathbf{U}$ | Vector of conserved variables |
| $\mathbf{v}$ | Tangential velocity vector, m/s |
| $V$ | Volume, $m^3$ |
| $\dot{w}_i$ | Density production rate of species $i$, $kg/m^3$ s |
| $w$ | Cell interface or control volume surface velocity, m/s |
| $\mathbf{w}$ | Cell interface or control volume surface velocity vector, m/s |
| $\mathbf{W}$ | Vector of species production rates, $kg/m^3$ s |
| $x_i$ | Mole fraction of species $i$ |
| $x$ | Cartesian coordinate, m |
| $X_i$ | Molal concentration of species $i$ per unit volume, $mol/m^3$ |
| $y$ | Cartesian coordinate, m |
| $z$ | Cartesian coordinate, m |
| $Z_{i,r}$ | Third body efficiency for species $i$, reaction $r$ |
| $\alpha$ | Angle of attack, deg; Grid compression parameter; Shock speed blending parameter |
| $\alpha_{i,r}$ | Stoichiometric coefficient for reactant $i$, reaction $r$ |
| $\beta$ | Angle of sideslip, deg; Grid compression parameter |
| $\beta_{i,r}$ | Stoichiometric coefficient for product $i$, reaction $r$ |
| $\gamma$ | Ratio of specific heats $c_p/c_v$ |
| $\delta$ | Fringe shift offset; Increment |
| $\Delta$ | Difference; Increment; Standoff distance, m |
| $\epsilon$ | Parameter for the van Albada limiter |
| $\theta$ | Cone angle, deg; Local flow incidence angle, rad |
| $\kappa$ | Blending parameter |
| $\lambda$ | Bulk viscosity, Pa s; Mean free path, m; Pressure correction scale factor; Wavelength, m |
| $\mu$ | Collision frequency, 1/s; Mach angle, deg; Molecular viscosity, Pa s |
| $\nu$ | Specific volume, $m^3/kg$ |
| $\rho$ | Density, $kg/m^3$ |
| $\sigma$ | Standard deviation in pressure measurement, Pa |
| $\tau$ | Viscous stress, Pa |
| $\phi_{ij}$ | Term in Wilke's viscosity law |
| $\psi$ | Limiter function |
| $\omega$ | Shock speed upwinding function |
| $\Omega$ | Collision cross section, $m^2$ |

## Subscripts

| | |
|---|---|
| $\circ$ | Standard condition |
| $\infty$ | Free stream value |
| a | Amplitude |
| b | Body |
| c | Curvature |
| ctr | Cell centre |
| D | Flux difference splitting |
| e | Exact solution |
| g | Gauge |
| i | Inner |
| if | Cell interface |
| *k* | Polynomial coefficient index |
| L | Left |
| n | Normal |
| o | Outer |
| R | Right |
| S | Shock |
| t | Tangential; Total (stagnation) |
| V | Flux vector splitting |
| vtx | Cell vertex |

## Superscripts

| | |
|---|---|
| *j* | Iteration number |
| L | Left |
| R | Right |
| $*$ | Best estimate |
| $\hat{}$ | Unit vector |

# Acronyms

| | |
|---|---|
| AUSM | Advection upwind splitting method |
| AUSMDV | AUSM combining flux difference and vector splitting |
| CFD | Computational fluid dynamics |
| EFM | Equilibrium flux method |
| ENO | Essentially non-oscillatory |
| FADS | Flush air data system |
| HYFLEX | Hypersonic flight experiment |
| IMU | Inertial measurement unit |
| MUSCL | Monotone upwind scheme for conservation laws |
| NAL | National aerospace laboratory of Japan |
| NASA | National aeronautics and space administration of the United States of America |
| NASDA | National space development agency of Japan |
| SF3D | Shock fitting and capturing three-dimensional Navier-Stokes solver |
| SVD | Singular value decomposition |
| TLNS | Thin layer Navier-Stokes |
| TVB | Total variation bounded |
| TVD | Total variation diminishing |

CHAPTER 1

# Introduction

Supersonic and hypersonic flight vehicles are commonly designed and manufactured with blunt noses. Heat and pressure loadings are often most extreme at the vehicle bow, and a large nose radius helps to withstand, distribute, and dissipate these loads. For the particular case of atmospheric entry and re-entry vehicles, high bluntness contributes to the drag production that is necessary to decelerate from superorbital to subsonic speed. In contrast, supersonic and hypersonic cruise vehicles need low drag to efficiently maintain velocity, and this requirement is optimally satisfied with a small but finite nose bluntness.[1] Understanding, analysing and predicting high speed flow around blunt bodies thus poses a practical and important engineering problem; faster and better design of new flight vehicles depends on it.

Gas flow around the forebody of blunt-nosed vehicles is typically clean, and subject to few upstream disturbances. By sampling the pressure distribution at the nose of a craft during flight, it is thus possible to accurately infer vehicle air data parameters, such as angle of attack, angle of sideslip, and dynamic pressure. The onboard instrumentation used to perform such estimates in real time, is termed an air data system (ADS). A flush air data system (FADS) is an air data system where all pressure measurements are made at tappings flush with the vehicle surface. The new X-33[2,3] and X-34[4] reusable launch vehicles will both be fitted with flush air data systems.

A pressure model describing the relationship between air data and measured nose pressures is needed before air data parameters can be estimated. The development of such models is called air data system calibration, and is a task traditionally undertaken using empirical or approximate theory, flight data, and ground-based experiments. Some of these methods were used in the calibration of existing flush air data systems on the space shuttle orbiter,[5] and an F-18 supersonic aircraft.[6]

The expense, time, and risks associated with building and flying a test vehicle, coupled with the difficulty of measuring its flow-field, restrict the use of full flight experiments for the investigation of hypersonic blunt body flows and FADS calibration. Also, the iterative nature of investigative experimentation tends to further preclude the frequent use of flight experiments. The chief advantage of flight experiments, however, is that flow data of high quality is generally obtained.

A less expensive and, in some ways, more flexible arrangement for investigating blunt-body flow is ground-based testing. Available ground-based test facilities include projectile ranges and wind tunnels. Realistic simulation of high speed flight in a wind tunnel requires the production of high speed test flow, which is difficult to sustain for extended periods. Most hypersonic wind tunnels are impulse facilities, such as shock tunnels or expansion tubes, which produce short, infrequent, bursts of test flow. Problems with obtaining sufficient test time, correct test conditions, and a uniform flow, are characteristic of impulse facilities. It can also be difficult to make complete, nonintrusive measurements in the time available for an experiment. Projectile ranges, where the body itself is shot through the test gas, are effectively immune from flow nonuniformity problems. Projectile ranges are limited, though, by test time, instrumentation, and flow condition restrictions.

Referring to the role of wind tunnels in fluid dynamical studies, in 1946 von Neumann said[7]

> "Indeed, to a great extent, experimentation in fluid dynamics is carried out under conditions where the underlying physical principles are not in doubt, where the quantities to be observed are completely determined by known equations. The purpose of experiment is not to verify a theory but to replace a computation from an unquestioned theory by direct measurements. Thus wind tunnels, for example, are used at present, at least in part, as computing devices to integrate the partial differential equations of fluid dynamics."

Presently, there is no unquestioned theory that both completely and practically describes all facets of hypersonic flow. However, the fundamental conservation laws, combined with appropriate approximations about gas behaviour and flow physics, do comprise an adequate mathematical model in many circumstances. Computational fluid dynamics (CFD) is the process of solving these equations numerically, and in some circumstances is more convenient and accurate than using the wind tunnel as an integrator.

It was predicted last century that the development of a computing engine would necessarily guide the future course of science, with the operations of analysis being executed by machinery.[8] In the case of fluid dynamics, this is certainly true. CFD effectively opens up a flow field, allowing a full set of flow variables to be determined throughout the entire


Figure 1.1: Illustration of the HYFLEX beginning descent. (Source: Reference 9)

simulation domain. Additionally, the influences of experimental uncertainty and measurement error are nonexistent. A specific benefit arising from CFD in hypersonic aerodynamics, is that the range of available test conditions is only restricted by limitations of the physical models. Further, craft can be inexpensively tested at full scale, and design changes can be implemented very quickly. The simulation process is also conducive to the automatic optimization of vehicle design parameters. From a research perspective, CFD is useful for assessing the relative strength of competing or complementary physical processes. The effects of viscosity, turbulence, and chemical reaction on a blunt-body flow field, for example, can be viewed in isolation or combination. While simulations are usually precise and repeatable, their accuracy depends on the validity of the governing mathematical model, and the numerical technique employed to solve it. It is important that verification and validation of a CFD code is undertaken to ensure that credible results are produced.

The thesis of this work is that CFD is by itself up to the task of accurately and conveniently calibrating flush air data systems for hypersonic blunt-nosed vehicles. Through the use of CFD, the expense of flight testing and inaccuracy of approximate theory are avoided. Further, calibration data can be obtained to a fidelity not achievable in ground-based experimental facilities.

The thesis will be argued using the hypersonic flight experiment (HYFLEX) vehicle as a case study.[10] An illustration of the HYFLEX during re-entry is shown in Figure 1.1. The vehicle, which flew in 1996, was equipped with the pressure sensor infrastructure

required for a FADS. A complete FADS was not installed, however, and real time FADS-based estimates of air data were not produced during the flight. The pressure data recorded during flight, though, provide an ideal base to test FADS calibration techniques.

## 1.1 Blunt Body Flows

Since the HYFLEX FADS pressure sensors are located on its blunt nose, a brief introduction to the flow characteristics of a blunt body in a hypersonic stream is now presented. A more detailed discussion of blunt body flows is available in Reference 11, and the semantics of defining aerodynamic bluntness are dealt with in Reference 12.

An diagram of air flow around a cylinder at Mach 6, is shown in Figure 1.2. The initially uniform free-stream flow is processed by a detached bow shock (S), and subsequently enters the shock layer. The hypersonic free stream is undisturbed by the downstream obstacle, since the speed of information propagation in that region is slower than the flow speed. The shock wave is strongest at the point where it is normal to the free stream inflow (N). Away from this location, the bow shock becomes oblique to the inflow and weakens, due to relief afforded by the body curvature.

Inside the shock layer, the sonic surface (L) demarks the transonic interface between subsonic and supersonic flow. For lower speed supersonic inflow, the interface would occur further downstream than pictured. Within the subsonic region bounded by the sonic surface, shock, and body, flow information is everywhere propagated in all directions via pressure waves. The stagnation point (T), is located within the subsonic region, and is defined as the location where flow impinges on the body in the surface-normal direction. In the case of an ideal, calorifically perfect gas, and an adiabatic body surface, flow pressure and temperature are highest at the stagnation point. The viscous boundary layer (B) is initiated at the stagnation point, and grows along the body surface in the downstream direction. In the presence of adverse pressure gradients, particularly in the shadow region behind the body, the boundary layer may at some stage separate from the surface.

As gas advects out of the subsonic region, it expands (E) and accelerates into the increased volume between shock and body. The decreasing shock angle, combined with the effects of flow expansion, usually results in a decrease of both pressure and temperature. At points downstream of the subsonic region, the increased flow speed means that pressure waves can not travel back upstream. Hence, the state of the downstream flow field does not affect the subsonic region, except possibly via electromagnetic field or the boundary layer. Hence, the simulation of a complete hypersonic vehicle is not necessarily required to accurately reproduce the flow around its nose. This idea is later exploited, when simulating the HYFLEX forebody.

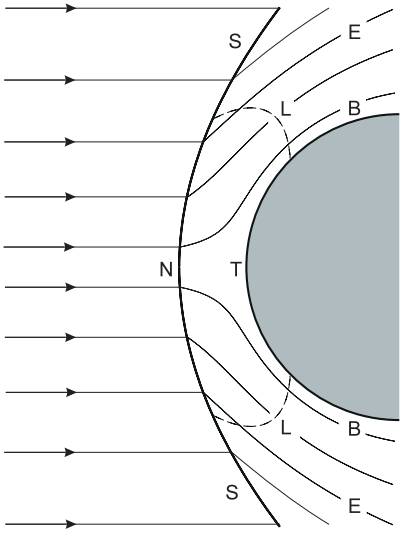



Figure 1.2: A blunt body in a hypersonic stream.

At the high temperatures within the shock layer, the gas may no longer behave in an ideal manner. In high temperature parts of the flow, gas component species may react and dissociate or ionize. Reaction catalysis and recombination of species can occur at the surface. Radiation from the hot shock layer or body may preheat the free stream flow, and the shock layer itself may be in thermal nonequilibrium.

## 1.2 Thesis Outline

The aims of this thesis are sixfold:

(i) To develop a new, robust, and accurate computational fluid dynamics code tailored for the hypersonic blunt-body problem, and air data system calibration.

(ii) To evaluate and verify the new code on various test cases.

(iii) To examine the influence of high temperature gas effects on blunt-body surface pressure.

(iv) To investigate the use of experiment and approximate theory for the calibration of flush air data systems.

(v) To develop a technique allowing computational fluid dynamics results to be used for a flush air data system calibration.

(vi) To show that a flush air data system on a hypersonic blunt-nosed vehicle may be accurately and conveniently calibrated using computational fluid dynamics simulations exclusively.

To achieve these aims, we first go thoroughly into the theoretical and mathematical background of CFD, and draw together the necessary algorithmic components required for the development of a good code. The new code, named SF3D, is then tested, and subsequently applied to the practical engineering problems of hypersonic vehicle simulation and FADS calibration. To assist the reader, a breakdown of the purpose and contents of following chapters is provided below.

**Chapter 2.** In this chapter, a general framework for the solution of the Navier-Stokes equations is presented. Finite-volume discretization of a flow field, numerical integration of the governing equations, and different thermodynamic gas models are described. Specific attention is paid to the use of finite-volume cells with moving boundaries.

**Chapter 3.** Reconstruction schemes and flux solvers, which are perhaps the two most important elements of a CFD code, are covered in the third chapter. Some desirable properties of these code elements are outlined. A range of different reconstructions and flux algorithms are surveyed, with particular emphasis placed on their associated advantages and failings when applied to blunt-body flows. The chapter concludes with a discussion of an instability that occurs when some of the flux solvers are used to simulate a blunt heat shield entering the atmosphere of Mars.

**Chapter 4.** In the fourth chapter, a shock-fitting technique is proposed to alleviate unphysical instabilities produced by otherwise accurate flux solvers. Shock fitting is shown to boost accuracy, efficiency, and robustness, and allow simulations to be performed without the addition of excessive artificial dissipation.

**Chapter 5.** A range of verification and validation test cases are used to demonstrate the credibility of results generated by the SF3D code. The geometric conservation property of the code is proven, code speed is tested, order of grid convergence is evaluated, and temporal convergence is demonstrated. Also, computed results are compared with those obtained from shock tunnel, expansion tube, and other experiments. Based on the results in this and preceding chapters, optimal numerical algorithms for blunt-body simulation are selected.

**Chapter 6.** Details of grid construction for simulations of the HYFLEX are presented in this chapter, and the selection of physical models for the simulations is justified. Also included are comparisons of CFD HYFLEX simulations with results obtained from shock tunnel experiments, flight data, and approximate theory. The accuracy of each technique is individually assessed to determine suitability for use in flush air data system calibration.

**Chapter 7.** In the seventh chapter, HYFLEX simulation results are used to calibrate its flush air data system. Air data estimates from the calibrated FADS are compared with values obtained from the onboard inertial measurement unit and an uncalibrated FADS.

**Chapter 8.** A summary of the thesis is presented in this final chapter. Conclusions are made, and some proposals and recommendations are drawn from the work.

CHAPTER 2

# A Navier-Stokes Solver

The primary application of the CFD code to be developed in this thesis is the simulation of the HYFLEX flight vehicle. We thus require the capability to resolve supersonic and hypersonic compressible flows with shocks, over a range of flight conditions encountered on the HYFLEX entry trajectory. In particular, we are most interested in obtaining an accurate description of the pressure field about the vehicle. Accurate pressure data is needed for the air data system calibration in Chapter 7, and for reconciling flight data and wind tunnel data with simulation results in Chapter 6. To perform the required simulations, an accurate and valid mathematical flow model is first needed.

Choosing a model with unnecessarily high accuracy can be counterproductive in practice, since the most accurate mathematical models are often the most complex and difficult to solve. Conversely, a formulation exhibiting poor accuracy may not suffice in some applications. The method of characteristics[13] is available for steady, inviscid, continuum, compressible flow, and is relatively computationally inexpensive. Potential flow theory[14] is also relatively inexpensive, and is capable of modelling these flows time accurately. For transonic flows and flows with strong shocks, however, solving the potential flow equations becomes more complicated. At a higher computational cost, the Euler equations[15] can be used to overcome these limitations and provide greater accuracy. Extra terms must be added to the Euler equations in order to model viscous effects and heat conduction; the resulting formulation is the Navier-Stokes equations. For rarefied gas flows, where intermolecular spacing is within a few orders of magnitude of a characteristic size of the flow field, the continuum assumption breaks down and the none of the preceding models are valid. Direct simulation methods and the Burnett equations are applicable in the rarefied gas and transition regimes.[16, 17]

Since only a small duration of the HYFLEX flight is spent in the rarefied flow regime, this thesis will be restricted to considering continuum flow only. Further, a time-accurate continuum model is required to properly simulate the starting processes in some of the ground-based blunt body experiments which shall be examined later. Also, the accurate reproduction of a hypersonic blunt body pressure field requires the simulation of viscous boundary layer effects (demonstrated in Chapter 6). Based on these considerations, the Navier-Stokes equations are chosen to model most of the flows in this thesis.

This chapter continues with a discussion of the generalized Navier-Stokes equations. A survey of techniques for discretizing and numerically solving the equations is subsequently presented, and the optimal methods for hypersonic blunt body flows are selected for inclusion in the SF3D computational fluid dynamics code. Various models for transport and thermodynamic gas properties are described, as are chemistry models for reacting gases.

## 2.1 The Navier-Stokes Equations

The Navier-Stokes equations describe the motion of continuum, viscous fluids. In integral form, the generalized Navier-Stokes equations for a control volume can be written as

$$\frac{\partial}{\partial t}\int_V \mathbf{U}\,dV + \int_S \mathbf{F}\,dS = \int_V \mathbf{Q}\,dV\ , \tag{2.1}$$

where **U** is a vector representing conserved flow quantities at points within the control volume, **F** is a vector of fluxes across the surface of the control volume, and **Q** is a vector of source terms. The scalars $V$ and $S$ refer to the volume and surface of the control volume respectively, and $t$ is the time variable. For the special case of a control volume with a porous surface moving with constant local velocity **w**, the vectors **U**, **F** and **Q** may be expressed as

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho\mathbf{u} \\ \rho E \\ \rho\mathbf{C} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \rho(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} \\ \rho\mathbf{u}(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} + P\hat{\mathbf{n}} - \mathbf{T}\hat{\mathbf{n}} \\ \rho E(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} + P\mathbf{u}\cdot\hat{\mathbf{n}} - \mathbf{u}\cdot(\mathbf{T}\hat{\mathbf{n}}) - \mathbf{q}\cdot\hat{\mathbf{n}} \\ \rho\mathbf{C}(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} \end{bmatrix}, \quad \text{and } \mathbf{Q} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \mathbf{W} \end{bmatrix}. \tag{2.2}$$

Here, **u** is the fluid velocity vector and $\hat{\mathbf{n}}$ is a unit normal to the control volume surface, pointing in the outward direction. The primitive variables $\rho$, $P$, and $E$ represent density, absolute static pressure, and intensive total energy respectively. Viscous flow stresses are contained in matrix **T**, and heat transfer rates in vector **q**. Species concentrations within

the fluid are represented by the vector **C**, while **W** is a vector of species production rates due to chemical reaction. Explicitly,

$$\mathbf{C} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_i \\ \vdots \\ C_{NS} \end{bmatrix}, \quad \text{and} \quad \mathbf{W} = \begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \\ \vdots \\ \dot{w}_i \\ \vdots \\ \dot{w}_{NS} \end{bmatrix}. \tag{2.3}$$

The mass fraction of species $i$ is denoted by $C_i$, with $i$ defined in the range $[1 \ldots NS]$, where $NS$ is the total number of species present in the control volume. Similarly, $\dot{w}_i$ is the density formation rate of species $i$.

With the definitions of Equations 2.2 and 2.3, the Navier-Stokes equations ensure conservation of mass, momentum, energy, and species. For a fluid containing only one species $NS = 1$, $\mathbf{C} = [1]$, and $\mathbf{W} = [0]$, and the conservation of mass and conservation of species conditions become tautologous. None of the physical flows studied in this thesis contain sources or sinks of mass, momentum or energy (other than those which enter through control volume surfaces). Thus every element of the source term, with the exception of **W**, is set to zero.

For most of the flows in this thesis it will be convenient to work in a three-dimensional right-hand Cartesian coordinate system. For such a system, we now define

$$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}, \ \mathbf{w} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix}, \ \hat{\mathbf{n}} = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}, \ \mathbf{q} = \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix}, \ \text{and } \mathbf{T} = \begin{bmatrix} \tau_{xx} & \tau_{yx} & \tau_{zx} \\ \tau_{xy} & \tau_{yy} & \tau_{zy} \\ \tau_{xz} & \tau_{yz} & \tau_{zz} \end{bmatrix}. \tag{2.4}$$

The $x$, $y$, and $z$ subscripts refer to coordinate axes pointing in the $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ directions respectively. Additionally, the first subscript on a viscous stress term $\tau$ indicates the normal direction to the plane over which the stress is acting, while the second subscript refers to the direction of action.

In most practical fluid dynamics problems we are interested in the development of the flow field (specified by **U**) through time, or until steady-state is reached. That is, given $\mathbf{U}(t=0)$ we wish to integrate Equation 2.1 in time from $t = 0 \ldots t_{max}$, in order to solve for $\mathbf{U}(t = t_{max})$. Examining the definitions in Equations 2.2, it is seen that there are not enough unique relations to close the system and solve for $\mathbf{U}(t = t_{max})$. Additional relations describing the properties and behaviour of the fluid are needed. Over a broad range of

states, a continuum gas closely follows the functional relationships

$$P = P(\rho, e, \mathbf{C}), \tag{2.5}$$

$$T = T(\rho, e, \mathbf{C}), \tag{2.6}$$

$$\mathbf{T} = \mathbf{T}(\mu, \mathbf{u}), \tag{2.7}$$

$$\mathbf{q} = \mathbf{q}(\mu, T), \tag{2.8}$$

$$\mu = \mu(T), \quad \text{and} \tag{2.9}$$

$$\mathbf{W} = \mathbf{W}(\rho, e, \mathbf{C}), \tag{2.10}$$

where $T$ is the gas temperature, $e$ is the internal energy, and $\mu$ is viscosity. Equations 2.5 and 2.6 are often referred to as equations of state. Equations 2.7–2.9 model the transport properties of the gas, and Equation 2.10 models gas chemistry. There are some underlying assumptions and limits to the functional forms expressed in the Equations 2.5–2.10, and these will be detailed and justified later in the chapter.

To finally close the system of equations introduced in this section, we now express total energy in terms of internal energy and kinetic energy as

$$E = e + \tfrac{1}{2}|\mathbf{u}|^2. \tag{2.11}$$

In the above expression, gravitational potential energy is assumed negligible.

## 2.2 Solving the Navier-Stokes Equations

The most efficient and accurate method for solving the Navier-Stokes equations depends on the nature of the fluid flow problem. For very simple flow problems it is possible to develop analytical solutions to the Navier-Stokes equations. In general, however, the equations must be solved numerically.

The practical importance and range of applicability of the Navier-Stokes equations, coupled with the emergence and advancement of digital computer technology, have inspired the development of a wide range of numerical solution procedures over the last 40 years. Each solution technique has strengths and weaknesses, and it is important to select a technique well-suited to solving a particular flow problem. Some important considerations in selecting a numerical scheme are the boundary conditions, initial conditions and geometry of the flow domain, as well as the fluid type, accuracy requirements, computer speed, and problem dimension and scale. Most modern schemes rely on representing the continuum flow field by a number of discrete samples, or in other words, a discretization. We now group numerical solution methods into four broad classes, depending on the discretization technique.

**Finite-volume schemes.**[18–20] For schemes in this class, the desired solution domain is discretized into an array of small, finite, control volumes. The small control volumes are referred to as cells. The discretization is performed so that no two cells overlap, and so that no gaps exist between adjacent cell boundaries within the solution domain. Finite-volume schemes work by solving the integral form of the Navier-Stokes equations for the flow contained in each cell, so that eventually the entire flow field is advanced in time or towards a steady-state. Solving the Navier-Stokes equations for each cell is a simple task; the cells are made small enough so that the flow within them is close to homogeneous. The required fluxes through each cell surface are determined by considering the properties of neighbouring cells, or the boundary conditions of the solution domain.

The finite-volume philosophy, then, is to represent the flow as a conglomerate of discrete, locally uniform, sub-flows. Thus it is important that the length scale of flow features to be captured in the solution are significantly larger than the length scale of the finite volume cells. Finite-volume schemes are particularly useful for flows which contain discontinuities, such as contact surfaces and shock waves. The (theoretically) infinite gradient of flow properties at discontinuities presents few numerical problems for the integral Navier-Stokes formulation. Additionally, finite-volume schemes have the advantage of being inherently conservative over the whole solution domain; any flux of a conserved quantity out of one cell must enter the volume of a neighbouring cell. Complex geometries can be treated in a straight-forward manner by constructing an appropriate mesh of cells in physical space. There is no requirement for cells to be positioned in an ordered arrangement. Boundary conditions on edges of the solution domain are reasonably simple to implement.

**Finite-difference schemes.**[15,21,22] This class of schemes relies upon representing the flow field by a structured array of discrete points (or nodes). The differential form of the Navier-Stokes equations is used to update the flow properties at each node until convergence is reached. Taylor expansions are normally applied to estimate the gradients of flow properties that are required for solution of the differential equations. In practice, the Taylor expansions are evaluated by calculating the difference in flow properties between neighbouring nodes. Since finite-difference schemes require the calculation of derivatives, difficulties may be encountered if flow discontinuities are present within the solution domain. Also, these schemes are usually solved in a regular computational space, rather than directly in physical space. Metrics, which provide a mapping relation between the spaces, must then be calculated. Complex geometries require complex mappings, and can be difficult to handle. For finite-difference schemes, particular care must be taken to ensure that theoretically conserved flow quantities are actually conserved in physical

space. Finite-difference schemes are well-known, the oldest in use for computational fluid dynamics, and are usually easy to implement.

**Finite-element schemes.**[23,24] These schemes can be used to solve the Navier-Stokes equations in integral form. Discretization is performed by subdividing the problem domain into non-overlapping geometric elements of arbitrary shape and size. It is assumed that the solution to the Navier-Stokes equations for each element takes a particular functional form, and the equations are subsequently solved in a function space. Nodes, lying on or inside each element, are used for the evaluation of the function values and derivatives required for solution. While finite-element methods have been well developed for solving the incompressible Navier-Stokes equations at low Reynolds numbers, they are not in common use for compressible flow problems.

**Spectral methods.**[25,26] In these methods, the governing fluid dynamics equations are solved globally in wave space as a combination of polynomial basis functions. Compared to finite-difference and finite-volume methods, spectral schemes exhibit low dissipation and high accuracy and efficiency. Spectral methods, though, are currently limited to solving problems with relatively simple geometry and limited boundary conditions. Also, difficulties can be encountered in dealing with flow discontinuities. Considerable research work has been directed at alleviating these restrictions.[27]

For all methods, we should expect to get the same solution in the limit of an infinitely fine discretization and infinitely precise computer arithmetic. While each class of numerical schemes takes a different approach to solving the Navier-Stokes equations, definite similarities and relationships exist between the classes. Spectral methods, for example, can be viewed as very high order finite-difference methods.[25] Finite-element methods, meanwhile, are the analog of spectral methods applied locally. Finite-element and finite-volume methods also share properties; both are used to solve the integral form of the Navier-Stokes equations over an array of geometric entities, and the finite-volume method can be written as a finite-element weak formulation.[23] Further, Selmin (1993) describes a node-centred finite-volume technique that provides a spatial discretization equivalent to both cell-centred finite-volume and finite-element methods.[28]

This simulations performed in this thesis are concerned with compressible flows containing discontinuities, over a range of blunt-body geometric arrangements. Considering these specifications, a finite-volume scheme is the most attractive numerical method because (i) spectral and finite-difference solvers do not offer the geometric flexibility of the finite-volume method, (ii) it is usually more difficult to implement boundary conditions

using the spectral and finite-element techniques, compared to the finite-volume technique, (iii) the finite-volume technique guarantees physical conservation laws, (iv) flow discontinuities are easily captured with finite-volume methods, and (v) for compressible flows, current finite-volume methods are further evolved than finite-element methods.[24] The remainder of this chapter, then, is devoted to the development a new finite-volume CFD code. Although some of the presented treatment and algorithms are specific to finite-volume methods, much of it is applicable to the other classes of schemes also.

## 2.3 Volume Discretization

Before application of the finite-volume method, the entire flow domain must be discretized into a grid of non-overlapping control-volume cells. A finite-volume CFD code can operate on either unstructured, structured, or hybrid grids.

Unstructured grids are composed of polygonal cells interconnected in an unordered fashion. Triangular cells are normally used in two dimensional domains, with tetrahedrals commonly used in three dimensions. The cells, however, are allowed to be any shape and have any number of neighbouring cells. These properties make unstructured grids appealing for use with geometrically complicated flow domains. The flexibility of unstructured grids is particularly useful for adaptive-grid codes, where cells need to be added and deleted during the solution process. In terms of computer requirements, the irregular positioning of cells means that connectivity information between cell interfaces and nodes must be held in memory. Unstructured grid housekeeping requires dedicated code, and can become cumbersome.

In contrast, structured grids are composed of cells connected in an ordered, regular way. In two dimensions the cells must be quadrilateral, while in three dimensions the cells must be hexahedral. An example of a three-dimensional structured grid is shown in Figure 2.1. The regularity of the grid makes it possible to uniquely reference any cell with three indices. Structured grids, though, do not have the geometric flexibility of unstructured grids, and are more difficult to apply to complicated geometries. Using multiple blocks of structured grids can alleviate this problem to some degree. A strong advantage of structured grids is the ease of computer implementation — the regular cell ordering simplifies the storage and referencing of cells in memory. Since information about a cell and its neighbours is normally stored in close proximity, the benefits of computer cache memory is maximized. Additionally, structured grid codes are generally well-suited to vector and parallel processing.

A beneficial property of structured grids is that it is easy to align cells near the edges of the flow domain orthogonal to the boundary geometry. When the boundary represents a

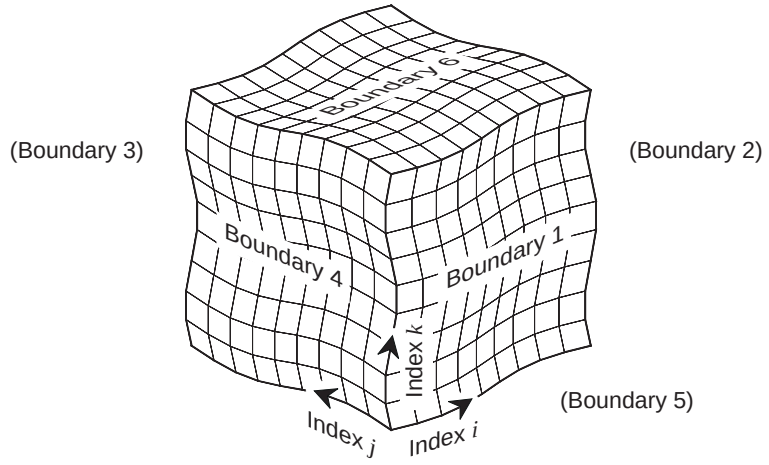



Figure 2.1: A structured discretization of the solution domain.

non-slip wall in a viscous simulation, this becomes an important property for the accurate simulation of boundary layers; it is a simple matter to arrange structured grid cells in a way that efficiently and accurately captures boundary layer gradients. It is more difficult to do the same with an unstructured grid.[29] Hybrid structured/unstructured grids combine the geometric flexibility of unstructured grids, with the boundary layer capturing properties of structured grids. Unfortunately hybrid grids also come with the expense of increased code complexity.

The blunt-body shapes to be simulated in this thesis are not complex enough to warrant the use of unstructured grids (and consequently incur the disadvantages associated with them). Thus we choose to employ a structured grid, with the form and nomenclature described by Figure 2.1.

Grid geometry can either be described directly in a physical coordinate system, or by metrics linking a computational domain to the physical system. The use of metrics, to some extent, allows the separation of geometry considerations from the core flow solver. However, metrics do add complexity to the numerical formulations, and make it more difficult ensure that the flow solver remains conservative. In this thesis, a physical coordinate system will be used throughout.

Grid quality can have a large bearing on solution accuracy. In fact, it is just as important to have a good grid as an accurate solver.[28] A desirable grid property is high resolution (more cells) in areas of the flow with strong gradients. Significantly distorted cells, cells with large aspect ratios, singularities, and neighbouring cells of disparate sizes should usually be avoided. Cell alignment relative to the flow direction is another important grid property, and can have either beneficial or detrimental effects. These effects are discussed in Chapters 3, 4, and 5.

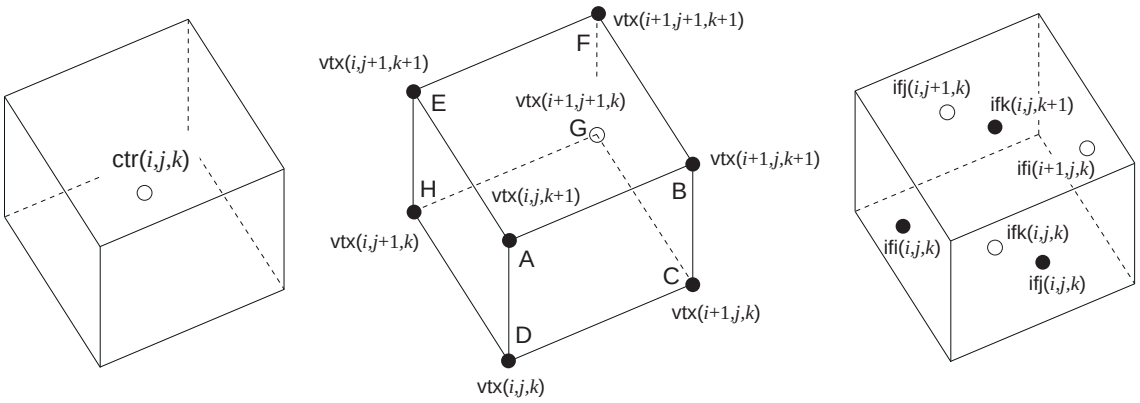



Figure 2.2: Nomenclature for the centre, vertices, and interfaces of cell$(i, j, k)$.

### 2.3.1 Cell Nomenclature

Figure 2.2 defines the nomenclature used to reference the components of a generalized hexahedral finite-volume cell. The geometric centre of cell$(i, j, k)$ is denoted by ctr$(i, j, k)$, and is associated with the average flow properties contained within that cell. The cell vertex shared between cell$(i, j, k)$ and cell$(i-1, j-1, k-1)$ is designated as vtx$(i, j, k)$. The geometry of a cell is completely defined by its eight vertices. Cell faces which are normal to the *i*-index direction are referenced using ifi, with ifj and ifk being similarly defined. The interface shared between cell$(i, j, k)$ and cell$(i-1, j, k)$ is denoted by ifi$(i, j, k)$. The geometric centre of an interface is associated with the average fluxes which pass through that interface.

Each cell — providing it is not located on the edge of the grid — shares geometry with 26 neighbouring cells. The entire grid, excluding boundaries, may thus be defined using just one set of vertex coordinates per cell. Likewise, each cell is uniquely associated with only three of its six interfaces. In practice, the elimination of duplicated geometry data reduces the computer memory required to perform CFD simulations.

### 2.3.2 Cell Geometry

Formulations for computing geometric properties of a nonorthogonal hexahedral cell are now presented. These properties include cell volumes, face areas, and centroids, and are required for the solution of the discretized Navier-Stokes equations.

It should first be noted that each face of a hexahedron contains an extra vertex than is required to define a plane. If the location of all cell vertices are arbitrarily chosen, it is not guaranteed (nor likely) that the four vertices on each cell face will be coplanar. Hence each face of a truly general hexahedron is actually a curved surface. While it is desirable to allow the arbitrary selection of cell vertices, we would at the same time

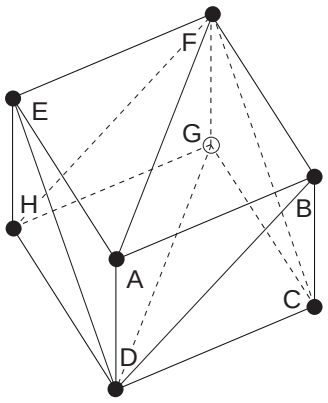



Figure 2.3: Triangulation of cell faces.

like to avoid the complexities associated with defining and managing curved cell faces. To simplify the volume discretization, we now decompose each hexahedron face into two (obviously planar) triangles. Thus the cells are no longer strictly hexahedra, but are instead hexahedroid dodecahedra.

The manner in which the hexahedron faces are decomposed is shown in Figure 2.3. Extra edges are defined by the lines **DB**, **DG**, **DE**, **FA**, **FH**, and **FC**. The regularity of the surface triangulation ensures that grid integrity is maintained when cells are stacked together.

The volume of the cell can be evaluated by considering the hexahedroid to be composed of six tetrahedra, with three tetrahedra located on each side of the internal boundary **ADFG**. With reference to Figure 2.3, the total signed cell volume is equal to the sum of the tetrahedron volumes:

$$V_{\substack{\mathbf{ABCD}\\\mathbf{EFGH}}} = \tfrac{1}{6}\left[\overline{\mathbf{AF}}\cdot(\overline{\mathbf{AB}}\times\overline{\mathbf{AD}})+\overline{\mathbf{AE}}\cdot(\overline{\mathbf{AF}}\times\overline{\mathbf{AD}})+\overline{\mathbf{DE}}\cdot(\overline{\mathbf{DF}}\times\overline{\mathbf{DH}})\right.$$
$$\left.+\overline{\mathbf{DH}}\cdot(\overline{\mathbf{DF}}\times\overline{\mathbf{DG}})+\overline{\mathbf{DG}}\cdot(\overline{\mathbf{DF}}\times\overline{\mathbf{DC}})+\overline{\mathbf{DF}}\cdot(\overline{\mathbf{DB}}\times\overline{\mathbf{DC}})\right]. \qquad (2.12)$$

The cell centroid is calculated as the volume-weighted mean of the tetrahedron centroids. The centroid of a tetrahedron is simply the average of its vertex position vectors.

It is easy to show that the area of each complete hexahedroid face is independent of the choice of triangulation vertices. The outward-pointing vector area of face **ABCD**, for example, is

$$\mathbf{A}_{\mathbf{ABCD}} = \tfrac{1}{2}\overline{\mathbf{AC}}\times\overline{\mathbf{DB}}, \qquad (2.13)$$

while the surface normal is

$$\hat{\mathbf{n}}_{\mathbf{ABCD}} = \frac{\mathbf{A}_{\mathbf{ABCD}}}{|\mathbf{A}_{\mathbf{ABCD}}|}. \qquad (2.14)$$

We deem the centroid of a hexahedroid face to be coincident with the area-weighted mean of the centroids of its two component triangles. Note that this definition does not mean that the face centroid will necessarily be coplanar with either of the triangles. The centroid of a triangle is just the average of its vertex position vectors.

## 2.4 The Discretized Navier-Stokes Equations

The Navier-Stokes equations are now discretized for application to a finite-volume cell. Since cell centre flow properties represent the averaged values for the cell volume, the volume integrals in Equation 2.1 become trivial. Similarly, by assuming cell-interface fluxes to be uniform over each cell face, the surface integral may easily be evaluated. For a single cell, Equation 2.1 can thus be recast as

$$\frac{\partial}{\partial t}\left(\mathbf{U}_{\text{ctr}} V\right) = \mathbf{Q}_{\text{ctr}} V - \sum_{\text{if}} \mathbf{F}_{\text{if}} A_{\text{if}}. \tag{2.15}$$

The volume of a cell with moving surfaces is a function of time, and thus $V$ remains inside the time derivative. Integrating Equation 2.15 with respect to time, from $t_n$ to $t_{n+1}$, yields

$$\mathbf{U}_{\text{ctr}}^{n+1} V^{n+1} - \mathbf{U}_{\text{ctr}}^{n} V^{n} = \int_{t_n}^{t_{n+1}} \mathbf{Q}_{\text{ctr}} V \, dt - \int_{t_n}^{t_{n+1}} \left( \sum_{\text{if}} \mathbf{F}_{\text{if}} A_{\text{if}} \right) dt, \tag{2.16}$$

and, after rearrangement,

$$\mathbf{U}_{\text{ctr}}^{n+1} = \frac{1}{V^{n+1}} \left[ \mathbf{U}_{\text{ctr}}^{n} V^{n} + \int_{t_n}^{t_{n+1}} \mathbf{Q}_{\text{ctr}} V \, dt - \sum_{\text{if}} \int_{t_n}^{t_{n+1}} \mathbf{F}_{\text{if}} A_{\text{if}} \, dt \right]. \tag{2.17}$$

Given the initial state of the cell defined by $\mathbf{U}_{\text{ctr}}^{n}$, the flow properties $\mathbf{U}_{\text{ctr}}^{n+1}$ of the cell at a later time can be determined by evaluating the right-hand-side of Equation 2.17. Specifically, methods for evaluating the inviscid terms of the flux vector $\mathbf{F}_{\text{if}}$ are discussed in Chapter 3. Section 2.8 details a technique for evaluating the viscous flux terms. A procedure for determining the source term $\mathbf{Q}$ is described in Section 2.10, and the time-behaviour of the geometry-related terms, $A_{\text{if}}$ and $V$, is addressed in Section 2.7. Numerical techniques for performing the time integrals in Equation 2.17 are presented in Section 2.6.

## 2.5 Behaviour of the Navier-Stokes Equations

The most appropriate approach for solving the discretized Navier-Stokes equations is, in part, dictated by the mathematical behaviour of the equations in different flow regimes. We begin this section with a description of the properties of the Navier-Stokes equations at steady state.

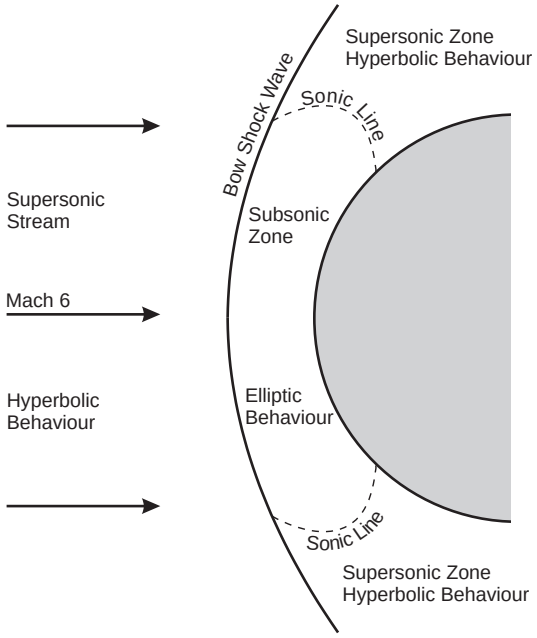



Figure 2.4: Steady inviscid flow around a cylinder in air at Mach 6.

The spatially hyperbolic nature of the steady (time-independent) supersonic Navier-Stokes equations can be exploited to expedite a solution. For wholly supersonic flow, the characteristic signals are everywhere real and always propagated in the downstream direction. In other words, the flow properties at a single spatial point in the flow are only dependent on upstream conditions, and, in turn, will influence only the downstream flow. Thus, by solving an upstream portion of the flow field first, and then continuing to march downstream, we automatically ignore any downstream influences and utilize the dependency properties. This technique is known as space-marching, and is an efficient way to solve steady supersonic flow problems.[29]

For steady subsonic flow the Navier-Stokes equations are spatially elliptic. Characteristics are everywhere imaginary, and signals are propagated in every direction.[30] The properties at a single point in the flow will thus influence the state of the entire flow field, and conversely. A range of iterative methods can be applied to solve elliptic problems, including the Jacobi and Gauss-Seidel relaxation schemes, as well as the alternating direction implicit scheme. These iterative methods are described in Reference 31 and will not be detailed here.

In many aerospace applications, mixed flow fields containing subsonic, transonic, and supersonic regions are common. Figure 2.4 shows typical flow around a blunt body in a supersonic stream. A space marching technique could not be used to solve the Navier-Stokes equations for this flow, since it would be physically incorrect in the subsonic zone and exhibit instability. Similarly, spatially iterative methods would become unstable in

the free stream and post-shock supersonic zones.[32] It is not usually a viable option to mix these solvers over the solution domain, since the exact location of sonic transition in mixed flows can not always be deduced beforehand.

A solution to the mixed flow problem is to solve the unsteady Navier-Stokes equations — even if only a steady-state solution is required. Since the state of the flow field at a given instant has no physical influence on earlier states of the flow field, the equations always exhibit hyperbolic properties with respect to time. Thus, by marching through time (rather than space) it is possible to solve the Navier-Stokes equations in a stable manner over a wide range of flow conditions. To boost solution efficiency for steady flows, however, it may still be desirable to solve portions of the flow that are guaranteed to be supersonic using a dedicated space-marching method. This technique is particularly useful for long, slender blunt bodies.

## 2.6 Time Integration

The time integrals in Equation 2.17 define the temporal evolution of the solution domain. In general it is not possible to evaluate these integrals analytically, and a form of numerical iteration must be used to march the flow field solution forward in time.[33] The optimal numerical integration method is dictated by a number of considerations, including the required temporal accuracy of solution, the characteristics of the flow field (which are in turn reflected by the characteristics of the Navier-Stokes equations), whether a steady-state or time-accurate solution is desired or needed, and the availability and type of computer resources. With these issues in mind, we now describe two approaches to iterative time integration and discuss their relative merits.

Explicit methods use an approximation to the exact time integral that requires knowledge of just the current state of the flow field. The approximation, however, is usually only valid and accurate for a small subsequent time period. Explicit methods thus rely on incrementally progressing a flow field solution in small timesteps, until the desired simulation time or steady-state convergence is reached. Implicit time integration methods formulate the flow solution at a subsequent time in terms of the flow field state at subsequent, current, and (sometimes) previous times. The new flow field is thus partially expressed in terms of unknowns, and solution requires the inversion of matrices containing large sets of simultaneous relations. For non-linear problems, implicit methods commonly employ timestepping in a similar manner to the explicit methods.

Although implicit and explicit methods both use temporal discretization, their mathematical behaviour differs significantly. In explicit schemes, the magnitude of the timestep is restricted by numerical stability considerations and is highly dependent on the nature of

the flow field and grid geometry. Most implicit schemes, however, retain stability at larger timesteps, with some implicit schemes being unconditionally stable.[32] Larger timesteps are usually desirable, since fewer iterations are required to integrate over a given time period.

The improved stability and larger timesteps of implicit schemes come at the cost of increased code complexity, and higher per-step computational effort. Before selecting a time integration method it is important to consider the balance between the number of timesteps required to complete a simulation, and the number of operations required to complete each step. For Euler flows of ideal gases, an implicit scheme will generally converge to a steady-state solution sooner than an explicit scheme. The larger timesteps of implicit schemes, however, may cause the time history of flow development to be poorly resolved in the interim. For an implicit method, the timestep must be reduced to get an accurate picture of unsteady flow development, thereby negating its efficiency advantage to an extent. Also, in stiff systems containing a range of disparate timescales (due to non-core flow processes such as viscous and chemical nonequilibrium effects) the timestep must necessarily be reduced anyway, further reducing the advantage of implicit methods.[34] In practice, explicit schemes are relatively simple to code and implement, and are easily cast in a form suitable for efficient parallelization.[15,34] They are less memory intensive than implicit schemes, since only one flow field solution needs to be stored at a time. Explicit schemes are compatible with any flux solver, and do not require linearized flux operators.

The simulation work carried out in this thesis is mainly concerned with steady and unsteady viscous flows in chemical nonequilibrium. Here, the efficiency gains afforded by implicit methods for particular steady flows do not outweigh the versatility and simplicity of explicit methods for all flows. Explicit time integration is thus implemented in SF3D. The steady flows investigated in this thesis are still solved by SF3D in a time accurate manner using the explicit integrator; this technique is referred to as pseudo-unsteady calculation.[33]

### 2.6.1 Euler Integration

A simple, explicit approximation to the time integrals in Equation 2.17 is given by the first order Taylor series expansion

$$\mathbf{U}_{\mathrm{ctr}}^{n+1} = \frac{1}{V^{n+1}} \left[ \mathbf{U}_{\mathrm{ctr}}^{n} V^{n} + \left( \mathbf{Q}_{\mathrm{ctr}}^{n} V^{n} - \sum_{\mathrm{if}} \mathbf{F}_{\mathrm{if}}^{n} A_{\mathrm{if}}^{n} \right) \delta t \right], \tag{2.18}$$

which is often referred to as the Euler method. Here, the incremental timestep $\delta t$ is defined as

$$\delta t = t_{n+1} - t_n. \tag{2.19}$$

Second and higher order terms have been neglected in Equation 2.18, and the approximation has local error of order $\delta t^2$. After integration over many timesteps, the error terms accumulate and a global error of order $\delta t$ occurs.[35]

### 2.6.2 Runge-Kutta Integration

Numerical integration using the Euler method can be made arbitrarily accurate by reducing the timestep size. To an extent, though, it is more efficient to achieve greater accuracy by using a higher order scheme, rather than choosing a smaller timestep. High order Runge-Kutta methods were first applied to finite-volume problems by Jameson,[36] and have been widely used since.[35,37] Runge-Kutta methods for any order of accuracy are available, with the first-order scheme being identical to the Euler method. Dick[19] cites a modified fourth order accurate Runge-Kutta scheme as giving the best compromise between allowable timestep (defined by stability considerations) and computational effort per step. In numerical experiments, a six stage Runge-Kutta scheme was found to be most efficient in solving a steady aerofoil flow.[38]

A disadvantage of high order methods is the increased computer memory requirements. A second order Runge-Kutta scheme, for example, requires double the storage capacity of the Euler method. Similarly, third and fourth order schemes require at least triple the storage capacity of the Euler method. For the large grids used to solve the three-dimensional, viscous flows investigated in this thesis, memory usage is a significant consideration. Excessive memory access can result in cache misses and cause code performance degradation, while a lack of available memory limits grid resolution and simulation size. A second order Runge-Kutta timestepping scheme was implemented in the SF3D code, to keep memory requirements manageable. Additionally, since second order spatial reconstruction is used within the code (see Chapter 3), a higher than second order time integration scheme would be wasteful.

The conservative, second order Runge-Kutta method used in SF3D incorporates two stages. First, the Euler method is invoked to determine a first order approximation to the flow field at a half timestep,

$$\mathbf{U}_{\text{ctr}}^{n+\frac{1}{2}} = \frac{1}{V^{n+\frac{1}{2}}} \left[ \mathbf{U}_{\text{ctr}}^{n} V^{n} + \left( \mathbf{Q}_{\text{ctr}}^{n} V^{n} - \sum_{\text{if}} \mathbf{F}_{\text{if}}^{n} A_{\text{if}}^{n} \right) \frac{\delta t}{2} \right], \tag{2.20}$$

and the fluxes at half-time are calculated and subsequently used to step the original solution forward by the full time increment,

$$\mathbf{U}_{\text{ctr}}^{n+1} = \frac{1}{V^{n+1}} \left[ \mathbf{U}_{\text{ctr}}^{n} V^{n} + \left( \mathbf{Q}_{\text{ctr}}^{n+\frac{1}{2}} V^{n} - \sum_{\text{if}} \mathbf{F}_{\text{if}}^{n+\frac{1}{2}} A_{\text{if}}^{n} \right) \delta t \right]. \tag{2.21}$$

### 2.6.3 Stability Limits

To retain stability during explicit time integration, we restrict the maximum allowable timestep by the condition

$$\delta t \leq C t_c, \tag{2.22}$$

where $C$ is the Courant-Friedrichs-Lewy (CFL) number[39,40] and $t_c$ is the smallest characteristic timescale of physical processes occurring in the flow. For the flows investigated in this thesis, the physical processes include both inviscid fluid advection and viscous shear. We thus define

$$t_c = \min(t_{\text{inviscid}}, t_{\text{viscous}}). \tag{2.23}$$

A convenient, approximate choice for the inviscid timescale is the minimum time taken for a wave signal to cross a finite volume cell in the solution domain. Hence,

$$t_{\text{inviscid}} = \min_{\text{if}} \left( \frac{|\mathbf{h}_{\text{if}}|}{|\mathbf{u}_{\text{ctr}} \cdot \hat{\mathbf{h}}_{\text{if}}| + a_{\text{ctr}}} \right). \tag{2.24}$$

where $a$ is the local sound speed and $\mathbf{h}_{\text{if}}$ is the vector spanning a cell between opposite face centroids. The viscous timescale will be discussed later.

If it is desired to march all cells forward with a uniform timestep, the limiting timestep will be defined by the cell with the smallest characteristic timescale. If time accuracy is not required, however, it is possible to accelerate convergence to a steady state solution by using the local timestep limit for each cell.

For both the Euler and second order Runge-Kutta schemes, experience reveals that stability is usually achieved when the CFL number is in the range $0 \leq C \leq 0.5$. Additionally, Gottlieb and Shu have shown that these schemes exhibit the total variation diminishing (TVD) property for hyperbolic systems, so long as the CFL number is kept at, or less than unity.[41] The TVD property guarantees that new extrema will not be exhibited within the solution domain, thus helping avoid the development of unphysical oscillations. It is interesting to note that no basic fourth order Runge-Kutta methods are TVD.[41]

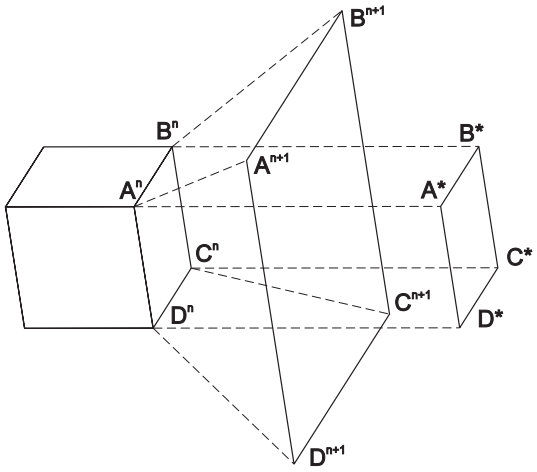



Figure 2.5: A moving cell interface.

## 2.7 The Geometric Conservation Law

In any computational fluid dynamics problem, and especially those in which a moving grid is used, it is important that consistency between the properties of geometrical entities is maintained. One important condition is that the sum of the cell volumes must always equal the volume of the entire solution domain, for example. For a moving grid, this implies that any extra volume accumulated by one cell must be accounted for by the loss of the same volume from other cells or movement of the domain boundaries. Thus, any volume change of a single cell must exactly correspond to the volume swept by its moving surfaces. This, in essence, is the Geometric Conservation Law (GCL). Any violation of the GCL translates to violation of physical conservation laws also.

Thomas and Lombard[42] have developed GCL expressions for the grid metrics of finite-difference flow computations performed in the computational domain. Gaitonde *et al.*[43] derive analogous expressions for two-dimensional finite-volume schemes in physical space. We now develop the GCL for finite-volume cells in three dimensions. With reference to Equation 2.1, the general expression for geometric conservation on a control volume is

$$\frac{\partial}{\partial t}\int_V dV + \int_S \mathbf{w}\cdot\hat{\mathbf{n}}\,dS = 0. \tag{2.25}$$

Performing a spatial discretization for a finite volume cell, and integrating with respect to time, yields

$$\delta V = -\sum_{\text{if}} \int_n^{n+1} A_{\text{if}}\,\mathbf{w}_{\text{if}}\cdot\hat{\mathbf{n}}\,dt. \tag{2.26}$$

where, without loss of generality, the velocity of each interface $\mathbf{w}_{\text{if}}$ is nominally taken as the average of its four vertex velocities. A first order approximation to the time integral gives

$$\delta V' = -\sum_{\text{if}} A^n_{\text{if}} \mathbf{w}'_{\text{if}} \cdot \hat{\mathbf{n}} \delta t. \tag{2.27}$$

where, in this case, $\mathbf{w}'_{\text{if}} = \mathbf{w}_{\text{if}}$ and is constant over $[n, n+1)$. The higher order terms missing from Equation 2.27 means that $\delta V'$ will not necessarily equal $\delta V$. For exact compliance with the GCL, the volume quantities must be equal. We now define a condition for a new interface velocity which guarantees geometric conservation:

$$\mathbf{w}'_{\text{if}} \cdot \hat{\mathbf{n}} = \frac{1}{A^n_{\text{if}} \delta t} \int_n^{n+1} A_{\text{if}} \mathbf{w}_{\text{if}} \cdot \hat{\mathbf{n}} \, dt. \tag{2.28}$$

The integral in Equation 2.28 is just the swept interface volume between timesteps. For the cell interface **ABCD** at timestep $n$ (shown in Figure 2.5) the new interface vertex positions at step $n+1$ are exactly

$$\mathbf{A}^{n+1} = \mathbf{A}^n + \mathbf{w}_{\mathbf{A}} \delta t, \tag{2.29}$$

$$\mathbf{B}^{n+1} = \mathbf{B}^n + \mathbf{w}_{\mathbf{B}} \delta t, \tag{2.30}$$

$$\mathbf{C}^{n+1} = \mathbf{C}^n + \mathbf{w}_{\mathbf{C}} \delta t, \text{ and} \tag{2.31}$$

$$\mathbf{D}^{n+1} = \mathbf{D}^n + \mathbf{w}_{\mathbf{D}} \delta t. \tag{2.32}$$

Thus Equation 2.28 simplifies to

$$\mathbf{w}'_{\text{if}} \cdot \hat{\mathbf{n}} = \frac{V_s}{A^n_{\text{if}} \delta t}, \tag{2.33}$$

where $V_s$ is the swept volume defined by the vertices $[\mathbf{ABCD}]^n [\mathbf{ABCD}]^{n+1}$. A similar GCL argument can be made for the second order Runge-Kutta scheme presented in Section 2.6.2, yielding the same result.

Equation 2.33 may be heuristically interpreted as compensation for neglecting the effect of the change in interface area between timesteps. It is this effect which would ordinarily be encompassed in the higher order terms absent from Equation 2.27. The formulated expression for $\mathbf{w}'_{\text{if}}$ ensures consistency between the actual swept interface volume and the swept volume computed under the approximation of a constant-area interface (denoted by the asterisked points in Figure 2.5).

In practice, the corrected interface velocity $\mathbf{w}'_{\text{if}}$ is usually very close to $\mathbf{w}_{\text{if}}$. Because the volume calculations required for Equation 2.33 add to computational expense, there is an option in SF3D to turn GCL adherence off. It should also be noted that the GCL interface velocity $\mathbf{w}'_{\text{if}}$ is only required for substitution into terms in Equations 2.18 and 2.21; elsewhere $\mathbf{w}_{\text{if}}$ is used as the interface velocity.

## 2.8 Treatment of Diffusive Terms

According to Stokes, the viscous stresses in a Newtonian fluid can be expressed by

$$\mathbf{T} = \begin{bmatrix} 2\mu\dfrac{\partial u_x}{\partial x} + \lambda\nabla\cdot\mathbf{u} & \mu\left(\dfrac{\partial u_x}{\partial y} + \dfrac{\partial u_y}{\partial x}\right) & \mu\left(\dfrac{\partial u_x}{\partial z} + \dfrac{\partial u_z}{\partial x}\right) \\ \mu\left(\dfrac{\partial u_y}{\partial x} + \dfrac{\partial u_x}{\partial y}\right) & 2\mu\dfrac{\partial u_y}{\partial y} + \lambda\nabla\cdot\mathbf{u} & \mu\left(\dfrac{\partial u_y}{\partial z} + \dfrac{\partial u_z}{\partial y}\right) \\ \mu\left(\dfrac{\partial u_z}{\partial x} + \dfrac{\partial u_x}{\partial z}\right) & \mu\left(\dfrac{\partial u_z}{\partial y} + \dfrac{\partial u_y}{\partial z}\right) & 2\mu\dfrac{\partial u_z}{\partial z} + \lambda\nabla\cdot\mathbf{u} \end{bmatrix}. \tag{2.34}$$

In the expressions, $\mu$ is the molecular viscosity coefficient and $\lambda$ is the second (or bulk) viscosity coefficient. The Stokes hypothesis

$$3\lambda + 2\mu = 0 \tag{2.35}$$

is used to relate the two coefficients.

Assuming that the heat transfer rate within the fluid is proportional to temperature gradient, Fourier's law is used to write the heat conduction term as

$$\mathbf{q} = k\left[\frac{\partial T}{\partial x} \quad \frac{\partial T}{\partial y} \quad \frac{\partial T}{\partial z}\right]^T. \tag{2.36}$$

The thermal conductivity $k$ is evaluated by assuming a Prandtl number and applying the definition

$$\mathrm{Pr} = \frac{c_p\mu}{k}, \tag{2.37}$$

where $c_p$ is the specific heat of the fluid at constant pressure. Physically, the Prandtl number gives an indication of the relative magnitudes of the diffusion of momentum and heat in the fluid. For a specific gas the Prandtl number is not strongly pressure dependent, but does vary somewhat with temperature.[44] For air, the Prandtl number remains within $\pm 0.05$ of 0.72 over the temperature range $100-2500$ K. Since the variation is not large, we will make the assumption of constant Prandtl number. A Prandtl number of 0.72 is also reasonably accurate for the homogeneous gases $H_2$, $O_2$, $N_2$, and $CO_2$.

To solve the finite-volume discretized Navier-Stokes equations (2.17) we need to evaluate the viscous stress and heat transfer terms at cell interfaces: $\mathbf{T}_{\mathrm{if}}$ and $\mathbf{q}_{\mathrm{if}}$. Examination of Equations 2.34 and 2.36 shows that we first need to know the corresponding velocity and temperature gradients at the interfaces. To solve for these quantities, an auxiliary mesh is constructed. The auxiliary mesh is defined by constructing cells with vertices coincident with the primary mesh cell centres. Figure 2.6 depicts the geometry of a single

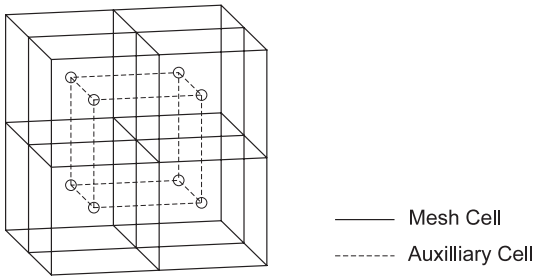



Figure 2.6: Auxiliary cell geometry.

auxiliary cell. For the special case of auxiliary cells near the primary mesh boundaries or corners, the auxiliary cell vertices are defined by the centroids or vertices of the primary mesh boundary interfaces respectively. The flow properties at the auxiliary cell vertices are set equal to those at the corresponding primary mesh cell centres, and the properties at the auxiliary cell interface centroids are taken as the average of the auxiliary interface vertex values. Next, Gauss' theorem is applied to determine the average flow gradients within the auxiliary cells. The local gradient of the vector velocity field can be written as

$$\nabla \cdot \mathbf{u}_{\text{ctr}} = \frac{1}{V} \sum_{\text{if}} \mathbf{u}_{\text{if}} \cdot \mathbf{A}_{\text{if}}, \tag{2.38}$$

while the partial derivative of any scalar quantity $s$ with respect to $x$, for example, is just

$$\frac{\partial s}{\partial x} = \frac{1}{V} \sum_{\text{if}} s_{\text{if}} (\mathbf{A}_{\text{if}} \cdot \hat{\mathbf{i}}). \tag{2.39}$$

Since the auxiliary cell volumetric centroids are in close proximity to the primary cell vertices, it is assumed that the gradients at both points are equal. Thus, the gradients at primary cell interface centroids may finally be evaluated by averaging the values at each of their four vertices.

Accurate calculation of the viscous effects in boundary layer flows requires sufficient mesh resolution. To achieve the correct resolution, it is common practice to perform grid compression near such regions. The compression function implemented in SF3D, for clustering in the $i$ index direction, is

$$c = \left( \frac{i_{\text{vtx}}}{i_{\substack{\text{vtx} \\ \text{max}}}} \right) \left[ \left( \frac{i_{\text{vtx}}}{i_{\substack{\text{vtx} \\ \text{max}}}} \right)^{\alpha} (1-\beta) + \beta \right], \quad \text{where } i_{\text{vtx}} = 0, 1, \ldots, i_{\substack{\text{vtx} \\ \text{max}}}. \tag{2.40}$$

The compression function $c$ varies between zero and one and, for a given $j$ and $k$, indicates the location of the $i$th cell vertex for a grid of span unity. The function is easily scalable for application on grids of varying dimension. The $\alpha$ parameter dictates the clustering severity, with a large value producing tight clustering towards the $i = 0$ boundary, and a value of zero producing an equispaced grid. The minimum cell size is limited to a fraction

$\beta$ of the cell size in an equivalent equispaced grid. The $\beta$ parameter is especially useful for generating grids with fine but nearly evenly spaced cells near the wall. By setting $\beta$ to an appropriate value, it is possible to avoid the generation of a single, overly thin cell at the boundary, which could severely limit the timestep through Equation 2.24. The clustering parameters should be selected in the range $\alpha \geq 0$, and $0 \leq \beta \leq 1$.

We now specify a characteristic viscous timescale for use with Equation 2.23, to maintain solver stability. An expression adapted from Reference 45 is used:

$$t_{\text{viscous}} = \frac{\rho \, \text{Pr}}{4\mu\gamma \left( \frac{1}{\mathbf{h}_{\text{ifi}}{}^2} + \frac{1}{\mathbf{h}_{\text{ifj}}{}^2} + \frac{1}{\mathbf{h}_{\text{ifk}}{}^2} \right)}. \tag{2.41}$$

In the above, $\gamma$ is the ratio of specific heats $c_p/c_v$.

### 2.8.1 Approximate Viscous Stress Models

The full Navier-Stokes equations for laminar flow are used throughout the SF3D code. For some specific flows, however, approximations to the viscous terms can be used to speed computation and minimize memory usage. These approximations are now briefly surveyed.

**Solution Matching Strategy.**[46,47] For some flows, such as that over a flat plate, viscous effects are only of importance near wall boundaries, and viscous effects elsewhere can be safely ignored with no significant accuracy degradation. Hence an Euler solver could be used to solve the bulk of the flow, while a boundary layer solver, analytical technique, or empirical approximation is applied to resolve the boundary layer. Matching solutions at the viscous/inviscid interface can cause problems such as entropy layer swallowing,[48] and the strategy may only be used in the absence of flow separation and recirculation.

**Thin Layer Navier Stokes.** The thin layer Navier Stokes (TLNS) equations are obtained by neglecting gradients in the wall direction from the viscous terms in the full Navier-Stokes equations. The simplification is usually valid, since grid compression (used to capture shear gradients normal to the wall) causes poor accuracy in the resolution of gradients in the wall direction anyway. Although the TLNS equations are cited as capable of resolving separated and reverse flow regions successfully,[49] it is still better to use the full Navier-Stokes equations when these regions dominate the flow.[50]

**Parabolized Navier Stokes.**[49] The parabolized Navier-Stokes (PNS) equations are most applicable to flows with one dominant flow direction. In the PNS equations, all streamwise derivatives are neglected from the viscous terms in the full Navier-Stokes equations.

Space marching can be applied to solve the PNS equations efficiently, at the cost of admitting only steady-state solutions. The PNS equations can not be used for streamwise separated flows, or blunt body flows with large sonic zones where the physical upstream transmission of information would be prevented.

**Viscous Shock Layer.**[48,51] The viscous shock layer (VSL) equations are another approximation of the Navier-Stokes equations, parabolized in both the streamwise and crossflow directions. Although more approximate than the PNS equations, the VSL equations can be used to compute blunt body flows containing sonic zones. The VSL equations, however, exclude time-accurate solutions and are not applicable to flows that are separated in either the streamwise or crossflow direction.

### 2.8.2 Viscosity Models

The viscosity of air at relatively low temperatures is well described by Sutherland's law,

$$\mu = 1.456 \times 10^{-6} \frac{T\sqrt{T}}{T + 110.4} \text{ Pa} \cdot \text{s}, \qquad T \leq 3000 \text{ K}, \tag{2.42}$$

where the air temperature $T$ is in Kelvin units. For higher temperature air or reacting air, or any other heterogeneous gas, we use Wilke's law to derive the viscosity of a gas mixture in terms of the viscosity of its individual component species.[52] Wilke's law is based on kinetic theory, and states that the mixture viscosity $\mu$ is approximately

$$\mu = \sum_{i=1}^{NS} \frac{x_i \mu_i}{\sum_{j=1}^{NS} x_j \phi_{ij}}, \tag{2.43}$$

where the viscosity and mole fraction of an individual gaseous species $i$ are given by $\mu_i$ and $x_i$ respectively. Appendix A gives curve fits for the viscosity of a range of species as a function of temperature. The species mole fractions can be easily expressed in terms of mass fractions by

$$x_i = \frac{C_i M}{M_i}, \quad \text{where } M = \sum_{j=1}^{NS} \frac{M_j}{C_j}. \tag{2.44}$$

The molecular weight of species $i$ is given by $M_i$, and $M$ represents the molecular weight of the mixture (that is, mass of mixture per mole of mixture). The $\phi_{ij}$ term in Equation 2.43 is defined as

$$\phi_{ij} = \frac{\left[1 + (\mu_i/\mu_j)^{1/2} (M_j/M_i)^{1/4}\right]^2}{\left[8(1 + M_i/M_j)\right]^{1/2}}, \tag{2.45}$$

with $\phi_{ii}$ clearly being equal to unity. The value of $\phi_{ji}$ can be efficiently computed from $\phi_{ij}$ using the identity

$$\phi_{ji} = \phi_{ij}\frac{\mu_j M_i}{\mu_i M_j}. \tag{2.46}$$

Computation can be further simplified by invoking the Herning and Zipperer approximation:[53]

$$\phi_{ij} \approx \sqrt{M_j/M_i}. \tag{2.47}$$

## 2.9 Thermodynamic Gas Models

Accurate thermodynamic descriptions of the fluid are required to close the Navier-Stokes equations, via the functional Equations 2.5 and 2.6. These equations specify that relations linking temperature and pressure to the fluid composition, density, and internal energy, are needed. To provide such relations, we now present two thermodynamic models of different complexity and accuracy. For the following treatment, the nomenclature lists the default units that the SF3D code assigns to the various thermodynamic quantities. All of the expressions are generalized, however, and should work with any consistent set of units.

### 2.9.1 Thermally Perfect, Ideal Gas

The ideal gas equation of state relates the pressure, temperature, and density state variables by

$$P = \rho RT, \tag{2.48}$$

where $R$ is the gas constant, and can be expressed in terms of the universal gas constant $R$ through

$$R = R/M, \quad \text{where } R = 8.31435 \text{ J/mol K}. \tag{2.49}$$

Equation 2.48 generally exhibits good accuracy at the high temperatures encountered in supersonic and hypersonic flow. The accuracy degrades, however, when intermolecular attractive forces become significant, such as those encountered at gas states of combined low temperature and high pressure. At 300 K, the ideal gas equation of state is valid for pressures up to 10 MPa.[54]

Before pressure can be computed through Equation 2.48, the gas temperature must be known. To find the temperature, we first consider the intensive mixture enthalpy:

$$h = \sum_{i=1}^{NS} C_i h_i. \tag{2.50}$$

The intensive enthalpy of species $i$ is denoted by $h_i$. The specific heat and enthalpy of a single species are related by the definition

$$c_{p,i} = \left(\frac{\partial h_i}{\partial T}\right)_P. \tag{2.51}$$

Here we have made the assumption that each species is in thermal equilibrium; in other words, the translational, rotational, vibrational, and electronic energy modes are taken to be equilibrated. For high speed flows at low density, the thermal equilibrium approximation breaks down when sudden changes in the gas state are introduced and the gas does not have sufficient time to relax (such as directly after shock processing). Typically, one hundred intermolecular collisions are required to sufficiently relax a species to thermal equilibrium.[55]

The enthalpy of a thermally perfect species is a function of temperature only, so that $h_i = h_i(T)$. We may thus simplify Equation 2.51 by replacing the partial derivatives with ordinary derivatives, yielding

$$dh_i = c_{p,i}\, dT. \tag{2.52}$$

Integrating, we have

$$h_i(T) = h_i^{\mathrm{f(0K)}} + \int_0^T c_{p,i}\, dT. \tag{2.53}$$

The constant of integration, $h_i^{\mathrm{f(0K)}}$, is the formation enthalpy of the species at 0 K and 100 kPa. To maintain consistency throughout the thesis, all thermodynamic quantities, where appropriate, will be referenced to this state unless otherwise noted. In practice, the species' specific heats can be conveniently defined by polynomial curve fits. A fourth order curve fit is sufficiently accurate for most species, and can be written as

$$c_{p,i} = R_i \sum_{j=0}^{4} A_{i,j} T^j, \tag{2.54}$$

where the polynomial coefficients for species $i$ are denoted by $A_{i,j}$, and the species specific gas constant is $R_i$. Appendix A lists the curve fit coefficients for selected species

in the temperature range from 300 K to 15,000 K. We now substitute Equation 2.54 into Equation 2.53, and evaluate the integral to obtain

$$h_i = h_i^{\mathrm{f(0K)}} + R_i \sum_{j=0}^{4} \frac{A_{i,j} T^{j+1}}{j+1} \tag{2.55}$$

for each species. The intensive internal energy and enthalpy of the mixture can be related by the thermodynamic definition

$$h = e + P/\rho. \tag{2.56}$$

Rearranging Equation 2.56, and combining with Equations 2.48, 2.49, and 2.55, we finally have

$$T = \left[ \sum_{i=1}^{NS} C_i M \left( \frac{h_i^{\mathrm{f(0K)}}}{R} + \frac{1}{M_i} \sum_{j=0}^{4} \frac{A_{i,j} T^{j+1}}{j+1} \right) \right] - \frac{eM}{R}. \tag{2.57}$$

Given the internal energy of the gas mixture and its composition, the secant method can be used to iteratively determine temperature from Equation 2.57. If the temperature of a cell from the previous timestep is used as a starting guess, only one or two iterations are typically required for an accurate result. Once the temperature of the gas is determined, it is a simple matter to find pressure using Equation 2.48.

The entropy of the gas is a quantity useful for flow visualization, and is also required in chemical kinetics calculations. The mixture entropy is

$$s = \sum_{i=1}^{NS} C_i s_i. \tag{2.58}$$

where the entropy contribution from each species can be found by evaluating the integral

$$s_i(T) = \int_0^T \frac{c_{p,i}}{T} dT. \tag{2.59}$$

to yield

$$s_i(T) = s_i^{\mathrm{c}} + R_i \left( A_{i,0} \ln(T) + \sum_{j=1}^{4} \frac{A_{i,j} T^j}{j} \right). \tag{2.60}$$

By definition, the entropy of a species must be zero at a temperature of absolute zero. Even though Equation 2.60 is indeterminate at absolute zero, a constant of integration $s_i^{\mathrm{c}}$ is still introduced to ensure consistency with the zero entropy reference condition at other temperatures.

The mixture sound speed, $a$, can be evaluated by

$$a = \sqrt{\gamma R T}. \tag{2.61}$$

Remembering that

$$R = c_p - c_v, \tag{2.62}$$

the ratio of specific heats for a mixture can be written as

$$\gamma = c_p/c_v = (1 - R/c_p)^{-1}. \tag{2.63}$$

The sound speed is required for evaluation of the CFL stability criterion, as well as for some flux solvers.

### 2.9.2 Calorifically Perfect, Ideal Gas

Calorifically perfect gases are defined as those which possess constant specific heat and constant specific volume. The calorifically perfect assumption simplifies many of the expressions in Section 2.9.1, and allows us to write the mixture enthalpy as

$$h = c_p T + h^{\mathrm{f(0K)}}. \tag{2.64}$$

The gas temperature can now be directly evaluated by

$$T = \frac{e - h^{\mathrm{f(0K)}}}{c_p - R}, \tag{2.65}$$

and pressure by the ideal gas equation of state. Providing that the calorifically perfect assumption is justified, these computationally inexpensive relations are used in preference to those in the previous section. The assumption, however, is usually only valid for limited temperature ranges, and the range of validity is strongly species dependent.

## 2.10 Reaction Models

Most notably at high temperatures, the composition of a gas in some flows will not always remain uniform throughout the solution domain. Reactions within the gas may occur and cause the dissociation of molecular compounds into their component atoms or other molecules, the dissociation of polyatomic species into atomic species, the combination or recombination of molecules and atoms into new molecules, and the ionization of molecules and atoms into ions and electrons. Three classes of reaction models are now described for modelling such processes, and their ranges of applicability are discussed.

### 2.10.1 Frozen Reactions

All chemical reaction rates are zero in a gas with frozen reactions and, in the absence of diffusion, species concentrations remain constant. Hence no specific reaction models are necessary in this case. When coupled with fluid mechanics, it is valid to assume that a gas has frozen chemistry if the characteristic timescales of reactions are much larger than the characteristic flow timescales. Because reaction rates increase with temperature, and production rates increase with density, the frozen gas assumption is most applicable to high speed, low temperature, low density (high altitude) flows containing slowly reacting species and few catalysts.

### 2.10.2 Equilibrium Reactions

The opposite extreme to frozen reactions is equilibrium gas chemistry. Physically, chemical equilibrium is reached after a gas has been allowed to react for an infinite amount of time. The final equilibrium composition of the system is a state variable and, with one other state variable, completely defines the gas state. For a flow containing regions of gas at different states, we take the equilibrium assumption to mean that the gas is everywhere in a state of local equilibrium. Thus, the equilibrium assumption is valid when the timescales of reactions are much smaller than the characteristic flow timescales. This means that for low speed, high temperature, high density (low altitude) flows containing quickly reacting species and catalysts, the equilibrium assumption may be applicable.

The SF3D code includes three equilibrium gas models. The equilibrium carbon dioxide code and model are obtained from Reference 56, and supplies the gas pressure, temperature, sound speed, and species concentrations in the form

$$P = P(\rho, e), \;\; T = T(\rho, e), \;\; a = a(\rho, e), \;\; \text{and} \;\; \mathbf{C} = \mathbf{C}(\rho, e). \tag{2.66}$$

The model is valid for density in the range $1 \times 10^{-7} \leq \rho \leq 1 \times 10^{2}$ kg/m$^3$, and for energies from $4.7 \times 10^4$ J/kg to a floating limit representing a temperature of 15,000 K at the required density. The model assumes a five species, three reaction system:

$$\mathrm{CO_2} = \mathrm{CO} + \tfrac{1}{2}\mathrm{O_2} \tag{2.67}$$

$$\mathrm{O_2} = 2\mathrm{O} \tag{2.68}$$

$$\mathrm{CO} = \mathrm{C} + \mathrm{O}. \tag{2.69}$$

In common with most equilibrium solvers, species concentrations at equilibrium are found by minimizing the Gibbs' free energy of the carbon dioxide system.

Equilibrium nitrogen and equilibrium air codes are obtained from Reference 57, and are based on the model data from References 58 and 59 respectively.

### 2.10.3 Nonequilibrium Reactions

When neither equilibrium nor frozen reaction models provide an adequate thermochemical description, the finite rates of chemical reactions must be considered. The implementation of finite-rate (nonequilibrium) gas models is more complex than for the preceding models, and they are substantially more computationally intensive due, in part, to their stiffness.[60] The nonequilibrium gas code implemented in SF3D is a re-written, tailored version of that appearing in Reference 61.

Consider a system of $NR$ reactions. Reaction $r$ can be written in the form

$$\sum_{i=1}^{NS} \alpha_{i,r} X_i \underset{k_{b,r}}{\overset{k_{f,r}}{\rightleftharpoons}} \sum_{i=1}^{NS} \beta_{i,r} X_i \quad \text{for } r = 1, 2, \ldots, NR, \tag{2.70}$$

where the per volume molal concentration of species $i$ is denoted by $X_i$, and defined as

$$X_i = \frac{C_i \rho}{M_i}. \tag{2.71}$$

The stoichiometric coefficient for reactant $i$ in reaction $r$ is $\alpha_{i,r}$, and is similarly $\beta_{i,r}$ for product $i$. The forward and backward reaction rates are denoted by $k_{f,r}$ and $k_{b,r}$ respectively. With these definitions, the density production rate for a species is found using the general rate equation:

$$\dot{w}_i = M_i \sum_{r=1}^{NR} (\beta_{i,r} - \alpha_{i,r}) \left( \sum_{i=1}^{NS} Z_{i,r} X_i \right) \left( k_{f,r} \prod_{i=1}^{NS} X_i^{\alpha_{i,r}} - k_{b,r} \prod_{i=1}^{NS} X_i^{\beta_{i,r}} \right). \tag{2.72}$$

The derivation of the general rate equation is found in Reference 62. Expressions for forward reaction rates as a function of temperature are available in the literature. Most such expressions are quite approximate at high temperature, since the rates are difficult to calculate or measure, and reaction pathways are not well known. Also, reaction rates may be affected by the catalytic properties of some surfaces. The $Z_{i,r}$ term in Equation 2.72 represents the efficiency of the catalytic effects of interactions between reactants and non-reacting (third) bodies. The forward rate models used in SF3D are of the modified Arrhenius from and are sourced from Gupta *et al.*[63] The backward reaction rates are calculated using the relation

$$k_{b,r} = \frac{k_{f,r}}{K_{eq,r}}, \tag{2.73}$$

where $K_{eq,r}$ is the equilibrium constant for reaction $r$. The equilibrium constant is defined in terms of the Gibbs' free energy of reaction and a pressure correction as

$$K_{eq,r} = \exp\left( -\frac{\Delta G_r}{R\,T} \right) \left( \frac{P_{atm}}{R\,T} \right)^{\sum_i (\beta_{i,r} - \alpha_{i,r})}, \tag{2.74}$$

where $P_{\text{atm}}$ is standard atmospheric pressure. The Gibbs' free energy of reaction, in mole units, is

$$\Delta G_r = \sum_{i=1}^{NS} M_i(\beta_{i,r} - \alpha_{i,r})(h_i - Ts_i). \tag{2.75}$$

Blottner's reaction model,[64] without ionization, is used for all of the nonequilibrium air flows in this thesis. The model comprises five species and seventeen reactions,

$$\begin{aligned}
\mathrm{O_2 + M_1} &\rightleftharpoons \mathrm{2O + M_1}, && (2.76)\\
\mathrm{N_2 + M_2} &\rightleftharpoons \mathrm{2N + M_2}, && (2.77)\\
\mathrm{NO + M_3} &\rightleftharpoons \mathrm{N + O + M_3}, && (2.78)\\
\mathrm{N_2 + N} &\rightleftharpoons \mathrm{2N + N}, && (2.79)\\
\mathrm{NO + O} &\rightleftharpoons \mathrm{O_2 + N}, \text{ and} && (2.80)\\
\mathrm{N_2 + O} &\rightleftharpoons \mathrm{NO + N}, && (2.81)
\end{aligned}$$

where Equations 2.76–2.78 represent fourteen individual third body reactions. The catalytic third bodies for each reaction are

$$\begin{aligned}
\mathrm{M_1} &= \mathrm{O_2, N_2, O, N, NO},\\
\mathrm{M_2} &= \mathrm{O_2, N_2, O, NO}, \text{ and}\\
\mathrm{M_3} &= \mathrm{O_2, N_2, O, N, NO}.
\end{aligned}$$

CHAPTER 3

# Reconstruction, Flux Solvers, and Stability

The numerical scheme described so far has only first-order spatial accuracy. In this chapter, we initially discuss methods to achieve higher orders of accuracy using solution reconstruction. Solution reconstruction involves two processes: interpolation and limiting. Interpolation refers to the calculation of flow variables at particular locations within the flow field, based on the knowledge of those variables at surrounding, discrete points. Limiting is subsequently used to ensure that spurious values are not admitted during the interpolation process. As a preprocessor, the purpose of reconstruction is to provide high-order estimates of flow quantities at cell interface states, which can be used as input data for a flux solver. The solver is then invoked to determine the fluxes of mass, momentum, and energy between cells. In the alternative postprocessing form, reconstruction is performed on the fluxes after they are computed. A taxonomy of flux solvers, and descriptions of individual algorithms, is also included in this chapter. The chapter is concluded with test cases that demonstrate and exacerbate some common failings of flux solvers applied to blunt-body flows. We investigate the failings, explain their causes, and evaluate some possible cures.

## 3.1 Interpolation

Perhaps the simplest approach to solution interpolation is the first order scheme suggested by Godunov.[65] Godunov originally approximated the flow field as consisting of a series of adjoining piecewise constant states. If we make this assumption, and each finite-volume

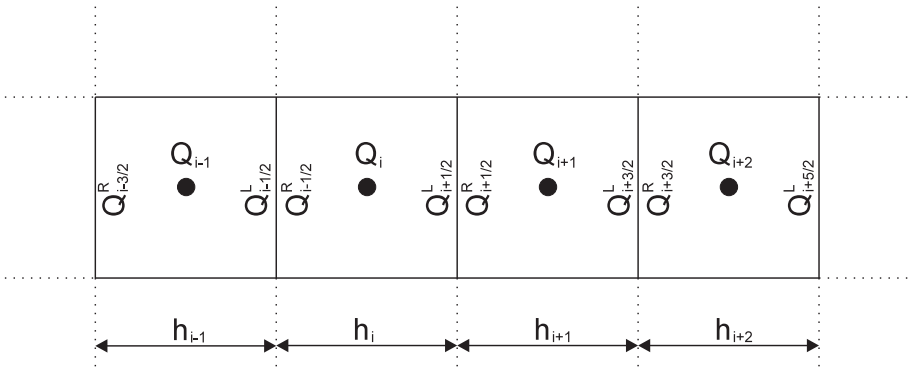



Figure 3.1: Cell nomenclature for one-dimensional reconstruction and limiting.

cell in the solution domain contains constant conditions throughout, then the flow conditions at the immediate left (L) and right (R) of cell interfaces are just

$$Q^{\mathrm{L}}_{i+1/2} = Q_i, \text{ and } Q^{\mathrm{R}}_{i+1/2} = Q_{i+1}, \tag{3.1}$$

where $Q$ is the interpolated quantity, and the nomenclature is defined in Figure 3.1. The scheme is both simple and computationally cheap. A problem with Godunov's approach, however, is its low order of accuracy; the scheme tends to cause smearing of flow features, especially at shocks and discontinuities. A range of higher order multi- and one-dimensional schemes have since been developed to address the accuracy issue.

Multi-dimensional interpolation schemes[66] are popularly used on unstructured grid computational fluid dynamics applications, where cells are not consistently aligned with the flow direction. Hence, to accurately interpolate the solution at a cell interface, the flow states in nearby cells in all directions should be considered. Ideally, multi-dimensional limiters and multi-dimensional flux solvers (Section 3.3) should be used to complement multi-dimensional interpolation.

One-dimensional interpolation requires fewer mathematical operations than multi-dimensional interpolation, and is thus preferred in those cases where the one-dimensional approximation does not cause significant accuracy loss. Well-designed structured grids, where possible, contain cells parallel or close to parallel to the flow gradients and flow direction. In this case, the use of locally one-dimensional interpolation is reasonably accurate and justified. A one-dimensional, fully one sided, second-order interpolation scheme is

$$Q^{\mathrm{L}}_{i+1/2} = Q_i + \tfrac{1}{2}\Delta^-_i, \qquad \text{where } \Delta^-_i = Q_i - Q_{i-1}, \text{ and} \tag{3.2}$$

$$Q^{\mathrm{R}}_{i+1/2} = Q_{i+1} - \tfrac{1}{2}\Delta^+_{i+1}, \quad \text{where } \Delta^+_i = Q_{i+1} - Q_i, \tag{3.3}$$

where $\Delta^+_i = \Delta^-_{i+1}$, and an equispaced grid is assumed such that $h_i = h_{i+1}$.

The one sided nature of Equations 3.2 and 3.3 means that downstream information is ignored, even though it may add to interpolation accuracy. Blending upstream and downstream contributions yields

$$Q^{\mathrm{L}}_{i+1/2} = Q_i + \tfrac{1}{4}[(1-\kappa)\Delta_i^- + (1+\kappa)\Delta_i^+] \tag{3.4}$$

$$Q^{\mathrm{R}}_{i+1/2} = Q_{i+1} - \tfrac{1}{4}[(1-\kappa)\Delta_{i+1}^+ + (1+\kappa)\Delta_{i+1}^-], \tag{3.5}$$

which is in the MUSCL[67] (monotone upwind schemes for conservation laws) form. The blending parameter is bounded by $-1 \leq \kappa \leq 1$, and in the special case of $\kappa = -1$, the MUSCL scheme reverts to the fully one sided scheme. For $\kappa = 0$ the interpolation uses equal contributions from upstream and downstream differences. When $\kappa = 1$, the scheme is no longer upstream biased, and the interpolated quantity at the interface is equal to the average of adjacent cell centre values.

As already stated, the purpose of interpolation is to define the fluxes at each side of each cell interface. Thus we have the option of either interpolating fluxes directly, or interpolating primary flow variables and calculating the fluxes later. While the latter technique is more expensive,[68] it leads to solutions with fewer spurious oscillations,[69] and is the mode in which SF3D operates.

## 3.2 Limiting and Reconstruction

The second-order interpolation techniques in Section 3.1 work well if $Q$ is a flow quantity smoothly distributed in space. However, for flows containing strong gradients or discontinuities, higher-order interpolation techniques can instigate high frequency noise production and instability, and cause the CFD solver to generate incorrect solutions or fail completely. In this section we introduce limiting functions, which are used to restrict the action of the higher-order interpolation schemes depending on the nature of the flow field. Ideally, in regions of severe flow discontinuities the limiter should cause the interpolation scheme to revert to the first-order Godunov approximation. In smooth flow regions the ideal limiter should have no effect, and allow the interpolation scheme to act unhindered.

As well as satisfying the above properties, the ideal limiter needs to be computationally economical, and work over a range of CFL numbers.[70] The search for the ideal limiter has generated considerable interest in the literature, and inspired the formalization of descriptions of limiter types and behaviours. The process of both limiting and interpolation is termed solution reconstruction, and we now give specifications for three popular classes of reconstruction schemes:

**Total Variation Diminishing.** As discussed in Section 2.6.3, total variation diminishing (TVD) schemes are defined as those which prohibit generation of new extrema, so that in one dimension

$$\mathrm{TV}(Q^{n+1}) \leq \mathrm{TV}(Q^n), \quad \text{where } \mathrm{TV}(Q) = \sum_i |Q_{i+1} - Q_i|. \tag{3.6}$$

As a result of this condition it can be shown that all TVD schemes preserve monotonicity,[69] and it is a requirement that a TVD limiter be non-linear to achieve higher than first order accuracy.[71] In general, reconstruction schemes satisfying the TVD condition are restricted to only second order accuracy.[70] They are further restricted to first order accuracy at non-sonic critical points (where $\partial Q/\partial x = 0$ in one dimension).[72] Although TVD schemes do help suppress the appearance of spurious noise in solutions, an unwanted side effect is the restricted development of high frequency components that may be present in the flow. Thus TVD schemes may cause the loss of genuine extrema, and flattening of spatial distributions.[70]

**Total Variation Bounded.** By relaxing the TVD condition to allow a finite bound on variation, we have the total variation bounded (TVB) condition[72]

$$\mathrm{TV}(Q^n) \leq B, \quad B > 0. \tag{3.7}$$

Hence any TVD scheme is also TVB. An advantage of TVB schemes are that the cited TVD order of accuracy restrictions do not apply, and they may be constructed with globally high order accuracy.[72] Additionally, TVB schemes do not necessarily prohibit the creation of genuine extrema.

**Essentially Non-Oscillatory.** By allowing any reduction in variation between timesteps to be regained in the next step, essentially non-oscillatory (ENO) schemes, like TVB schemes, allow local extrema to be accentuated. Unlike TVD schemes, ENO schemes can be constructed to have uniform global accuracy of any order. ENO schemes usually operate on adaptive stencils[73] and thus require additional computational effort. Since some TVD and TVB schemes can achieve a similar accuracy to simple ENO schemes with less computational expense,[70] only TVD and TVB schemes are implemented in SF3D.

The remainder of Section 3.2 is devoted to the definition and description of three limiters used for the TVD and TVB reconstruction schemes available in SF3D.

### 3.2.1 The min-mod Limiter

Consider the one sided interpolation for the left interface state given in Equation 3.2. Premultiplying the difference term by a limiter function $\psi$ gives

$$Q^{\mathrm{L}}_{i+1/2} = Q_i + \tfrac{1}{2}\psi(r_i)\Delta_i^-, \tag{3.8}$$

where $r$ is the ratio of differences, and defined as $r = \Delta^+/\Delta^-$. When $\psi = 1$, Equation 3.8 collapses to the original unlimited scheme. A commonly used limiter function is,

$$\psi(r) = m(1,r) \tag{3.9}$$

and is based on the minimum-modulus (min-mod) function $m$,

$$m(x,y) = m(y,x) = \mathrm{sign}(x)\max\{0,\min[|x|,y\cdot\mathrm{sign}(x)]\}, \tag{3.10}$$

which returns zero for two arguments of different signs, or the value of the argument with the smallest modulus otherwise. An equivalent expression to Equation 3.9 is

$$\psi = \frac{m(\Delta^-,\Delta^+)}{\Delta^-}. \tag{3.11}$$

The arguments to the min-mod function are now biased with a parameter $b$, and the upstream and downstream biased components are blended with parameter $\kappa$ to generate a MUSCL-type formulation:

$$\psi = \frac{(1-\kappa)\,m(\Delta^-,b\Delta^+) + (1+\kappa)\,m(b\Delta^-,\Delta^+)}{2\Delta^-}. \tag{3.12}$$

Note that for $b = 1$ we return to the one-sided scheme for all $\kappa$ in the range $-1 \le \kappa \le 1$. Substituting Equation 3.12 into Equation 3.8 yields

$$Q^{\mathrm{L}}_{i+1/2} = Q_i + \tfrac{1}{4}[(1-\kappa)\,m(\Delta_i^-,b\Delta_i^+) + (1+\kappa)\,m(b\Delta_i^-,\Delta_i^+)], \tag{3.13}$$

and similarly, for the right side of the interface,

$$Q^{\mathrm{R}}_{i+1/2} = Q_{i+1} - \tfrac{1}{4}[(1-\kappa)\,m(\Delta_{i+1}^-,b\Delta_{i+1}^+) + (1+\kappa)\,m(b\Delta_{i+1}^+,\Delta_{i+1}^-)]. \tag{3.14}$$

The reconstruction Equations 3.13 and 3.14 are equivalent to those appearing in Reference 74.

The shaded area in Figure 3.2 highlights the domain in which a limiter must operate to satisfy the TVD constraint for second order schemes,[69] while simultaneously not displaying undesirable overcompressive behaviour.[75] It is observed that the min-mod limiter lies within this domain. The main disadvantage of the min-mod limiter is that it is not differentiable.

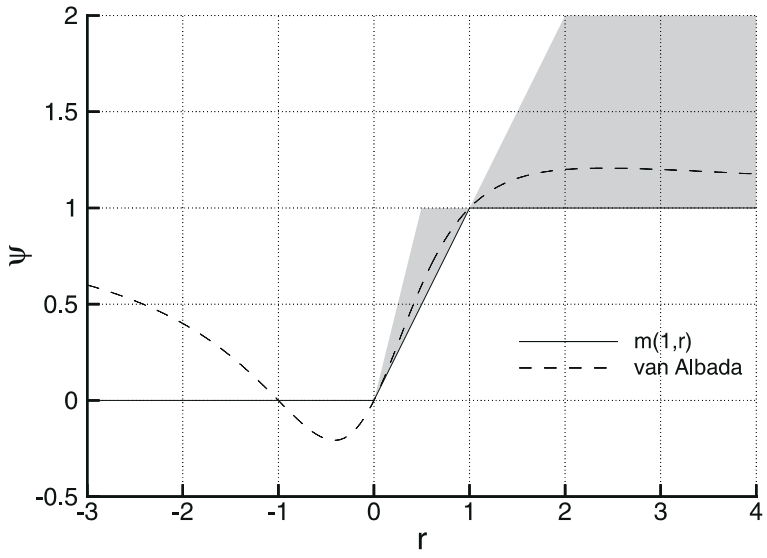



Figure 3.2: Behaviour of the min-mod and van Albada limiters.

### 3.2.2 The van Albada Limiter

The convergence of a solution to steady-state can be hampered by non-differentiable limiters. Such limiters can cause small temporal oscillations in solutions, prohibiting residuals to fall to machine precision levels. A differentiable limiter due to van Albada[76] is now presented.

We again begin with limited fully one sided interpolation:

$$Q^{\mathrm{L}}_{i+1/2} = Q_i + \tfrac{1}{2}\psi(r_i)\Delta_i^- . \tag{3.15}$$

The van Albada limiter is defined as

$$\psi(r) = \frac{r^2 + r}{r^2 + 1}, \tag{3.16}$$

and its behaviour in $\psi$–$r$ space is shown in Figure 3.2. The reconstruction scheme so far described is thus TVD, and is also second order accurate[71] except at critical points. We may equivalently express Equation 3.16 in the form

$$\psi = \frac{s(\Delta^+ + \Delta^-)}{2\Delta^-}, \tag{3.17}$$

where the function $s$ is given by

$$s = \frac{2\Delta^+\Delta^- + \epsilon}{(\Delta^+)^2 + (\Delta^-)^2 + \epsilon}. \tag{3.18}$$

The parameter $\epsilon$ is introduced to avoid division by zero, and is set slightly higher than machine precision (typically $\epsilon = 1 \times 10^{-12}$). By choosing a larger $\epsilon$, a few orders of

magnitude smaller than typical $\Delta^2$ values, the scheme loses the TVD property. In this case, the $\epsilon$ term dominates the difference terms at critical points and causes the limiter to switch off and revert to the underlying second order interpolation. Venkatakrishnan[66] claims that numerical experiments indicate such a scheme is TVB, although it has not been rigorously proven. We now weight the upstream and downstream components in the numerator of Equation 3.17 and substitute the result into Equation 3.15 to achieve a MUSCL formulation:

$$Q^{\mathrm{L}}_{i+1/2} = Q_i + \frac{s_i}{4}[(1-\kappa)\Delta_i^- + (1+\kappa)\Delta_i^+]. \tag{3.19}$$

Here, the original scheme is recovered when $\kappa = 0$. Numerical experiments conducted in the course of this work reveal that Equation 3.19 produces stable reconstruction for $\kappa = 0$, but is not always stable for $\kappa = 1$ and $\kappa = -1$. To achieve stability, we limit the action of the blending parameter so that

$$Q^{\mathrm{L}}_{i+1/2} = Q_i + \frac{s_i}{4}[(1-s_i\kappa)\Delta_i^- + (1+s_i\kappa)\Delta_i^+], \text{ and,} \tag{3.20}$$

$$Q^{\mathrm{R}}_{i+1/2} = Q_{i+1} - \frac{s_{i+1}}{4}[(1-s_{i+1}\kappa)\Delta_{i+1}^+ + (1+s_{i+1}\kappa)\Delta_{i+1}^-], \tag{3.21}$$

which is equivalent to the differentiable reconstruction scheme appearing in Reference 74.

### 3.2.3 A Modified van Albada Limiter

The assumption of equispaced cells is inherent to the reconstruction schemes presented so far. The difference in neighbouring cell widths in most well-designed grids does not often vary more than 10%, and the equispaced assumption is adequate for this level of variation.

The modelling of the HYFLEX in this thesis requires three-dimensional, time accurate, viscous CFD calculations with chemistry. This kind of problem is very computationally and memory intensive, and it is important to make optimum use of all cells in the solution domain. For this reason, the HYFLEX grid (see Section 6.1.1) makes a sudden transition in spacing between the cells in the boundary layer and cells in the core flow. The difference in neighbouring cell widths is sometimes as high as 60%, and the equispaced cell assumption is not valid.

We now present some new modifications to the van Albada limiter, to cope with unequally spaced grids. Accounting for the width of cell $i$, denoted by $h_i$, the one-sided reconstruction equation with limiting becomes

$$Q^{\mathrm{L}}_{i+1/2} = Q_i + \frac{h_i}{2}\psi(r'_i)\Delta_i^{\ominus}, \tag{3.22}$$

where the ratio of differences is now the ratio of gradients, $r' = \Delta^{\oplus}/\Delta^{\ominus}$. The downstream and upstream gradients are defined as

$$\Delta_i^{\ominus} = 2\Delta_i^{-}/(h_i + h_{i-1}), \text{ and} \tag{3.23}$$

$$\Delta_i^{\oplus} = 2\Delta_i^{+}/(h_{i+1} + h_i), \tag{3.24}$$

providing that no cells have zero width. The TVB limiter function given in Equation 3.18 is re-used with arguments $\Delta^{\ominus}$ and $\Delta^{\oplus}$ replacing $\Delta^{-}$ and $\Delta^{+}$ respectively, to obtain the MUSCL reconstruction

$$Q_{i+1/2}^{\mathrm{L}} = Q_i + \frac{h_i s_i}{4}[(1 - s_i\kappa)\Delta_i^{\ominus} + (1 + s_i\kappa)\Delta_i^{\oplus}], \text{ and} \tag{3.25}$$

$$Q_{i+1/2}^{\mathrm{R}} = Q_{i+1} - \frac{h_{i+1} s_{i+1}}{4}[(1 - s_{i+1}\kappa)\Delta_{i+1}^{\oplus} + (1 + s_{i+1}\kappa)\Delta_{i+1}^{\ominus}], \tag{3.26}$$

which is correctly spatially sensitive. This scheme is tested on a severely unequally spaced grid in Section 5.3, and excellent results are observed.

## 3.3 Flux Solvers

In this thesis, flux solvers are used to calculate the magnitude of mass, momentum, energy and species fluxes that pass from a finite-volume cell to its neighbour, via their shared interface surface. To operate, the solvers require knowledge of the flow state on each side of the interface, which is obtained through the solution reconstruction techniques already described. The simplest kind of flux solver is flux averaging, where the interface flux is taken as the mean of the fluxes at the left and right interface states. Although easy to implement, flux averaging is undesirable due to poor solution accuracy. A wide range of more accurate flux calculation algorithms have been developed, each with specific strengths and weaknesses. The perfect solver does not yet exist.

The value of a flux solver can be judged against several criteria. A good solver will exhibit high accuracy, and achieve good resolution in all areas of a flowfield. A high accuracy scheme should intrinsically possess low levels of artificial dissipation[77] to avoid smearing fine flow features. Unfortunately, low dissipation schemes are generally more prone to the development of numerical noise, including spurious oscillations, disturbances and discontinuities. Hence, the dissipation level governs a tradeoff between accuracy and noise. The goal of combining high accuracy and low noise is an elusive one.[78]

Flux solvers can fail altogether when numerical noise levels become excessive. The solver algorithm should ideally be robust, based on considerations of flow physics, and never allow unphysical solutions to be admitted. To allow autonomous operation, the absence of selectable tweaking or tuning parameters is preferred. Because the flux solver

is frequently called during the operation of a CFD code, it is important that the algorithm be computationally efficient.

Categorizing flux solvers is useful, since there is a degree of commonality in properties for solvers from the same class. We next describe a taxonomy of flux solvers, and subsequently examine some specific schemes. The form of the input/output interface of a flux solver is then discussed, and the mathematical transformations required for moving grids are presented.

### 3.3.1 Taxonomy of Flux Solvers

One of the coarsest classifications of flux solvers is dimensionality. One-dimensional schemes, which are by far the most common, rely on the assumption that the flux convection speed is independent of the fluid velocity tangential to an interface. One-dimensional schemes may be applied to multi-dimensional problems by computing the fluxes through each interface of each cell independently. A three dimensional oblique shock wave, for example, is computed by one-dimensional flux solvers as the superposition of three normal shocks emanating from each face of a cell. Although this kind of approach can lead to some smearing and has no physical basis, one-dimensional schemes work remarkably and inexplicably well in practice.[79] The development of genuine multi-dimensional schemes is a current area of vigorous research aimed at reducing the smearing effect.[80] Some multi-dimensional schemes work by considering the infinite directions of propagation of perturbing waves,[81,82] while others operate by solving a complicated multi-dimensional Riemann initial value problem.[83] The main disadvantage of multi-dimensional flux solvers is their complexity and thus greater computational requirements, and for this reason none will be considered in this thesis.

All of the flux solvers implemented in SF3D can be categorized as upwind schemes. Upwind schemes recognize that the local flow speed and direction should and does affect the way information propagates between cells. This is unlike the central difference scheme, which may unphysically draw flow information from outside the domain of dependence of an interface during flux calculation.[15] This detrimental property can cause central differencing to generate noise or fail, and necessitates the introduction of artificial viscosity terms. Upwind schemes have up to double the stability bound of central difference schemes,[84] and do not generally require the addition of extra artificial viscosity to maintain stability. For high Mach number flows, flows with shocks, and advection dominated problems, upwind schemes also give superior resolution.[32]

Many upwind schemes can be classed as either flux difference splitting (FDS) or flux vector splitting (FVS). The first FDS scheme was proposed Godunov,[65] whereby inter-

face fluxes are determined by the exact solution of the Riemann problem, using the left and right interface states as initial values. Approximate and linearized Riemann solvers are also FDS schemes. In FVS schemes the interface flux is calculated as the combination of split forward and backward component flux vectors, depending on the sign of the associated eigenvalues. In general, FVS schemes are simpler and faster than their FDS counterparts,[85] and the FVS formulation is particularly well-suited for use with implicit techniques.[86] A disadvantage associated with most FVS schemes is excessive dissipation, which can diffuse contact surfaces,[85] thicken shock waves,[87] and cause boundary layers to be inaccurately resolved.[87] For these flow features, FDS schemes are less dissipative and substantially more accurate than FVS schemes.[86,87] In fact, no FVS scheme can preserve a stationary contact discontinuity.[85] A number of FDVS hybrid schemes have been proposed to merge the speed and robustness of FVS, with the accuracy of FDS.[88–90] Still other schemes, which can be classed as neither FVS nor FDS, have been developed to achieve this same goal.[91]

There are several conditions imposed by physics that a good flux solver should satisfy, although they are not commonly adhered to. Positivity preserving schemes correctly disallow the production of negative values of the scalar variables density, pressure, temperature, and species concentrations. This condition is particularly important to prevent code failure in flows with colliding shocks, strong rarefactions, and multiple species.[92] As well as maintaining positivity, a flux solver should also not violate the entropy condition. This is usually manifested in the admittance of unphysical solutions such as discontinuities in expansion waves. Another desirable property of flux schemes is differentiable operation in the absence of flow discontinuities. In common with limiters, non-differentiable splittings can cause undesirable glitches or oscillations.[74]

### 3.3.2 Some Specific Flux Solvers

The operation and properties of some specific one-dimensional flux solvers and formulations, most of which are implemented in SF3D, are now described.

**Exact Riemann Solvers.** Also known as Godunov-type schemes, exact Riemann solvers find the unique solution to a Riemann initial value problem defined by the left and right interface states. The Riemann problem is solved by considering the speed, direction and strength of discrete pressure waves, shock waves, and a contact surface emanating from the interface. There are a number of approaches to solving the exact Riemann problem,[93] however all of them are iterative and at least moderately computationally expensive. Further, exact solutions become very expensive for non-polytropic gases and gases with gen-

eral equations of state.[79] In any case, for finite-volume schemes the detail of the exact Riemann solution at cell interfaces is lost during the averaging process that takes place during the update of cell centre properties.[90] Although somewhat wasteful, exact Riemann solvers do produce accurate and (usually) well-behaved shock waves because of their close physical base.[94] All types of Riemann solvers are inherently upwind and FDS.

**Approximate Riemann Solvers.** Because the accuracy of exact Riemann solvers is wasted to a large extent, approximate solutions can be employed without significant degradation of overall simulation results. Roe's approach[95] of solving a linearized Riemann problem is both cheap and popular. Unfortunately Roe's scheme admits entropy violating expansion shocks, and requires the addition of artificial dissipation to cure it (known as an entropy fix).[79] The Roe scheme also suffers from the so-called "carbuncle effect," an unphysical protuberance or indenture sometimes visible near the stagnation region of strong bow shocks. An alternative approximate Riemann solver has been proposed by Jacobs, which makes an estimate to the solution of the exact non-linear Riemann problem using only a few iterations.[96] This scheme, and Jacobs' original code, is implemented in SF3D. Although more reliable than Roe's scheme, this semi-iterative method still suffers from expansion shocks and carbuncles under some conditions. Neither scheme satisfies the positivity condition, and both schemes are susceptible to odd-even decoupling (see Section 3.6). Quirk contends that there is no approximate Riemann solver without at least one failing, and outlines a hybrid strategy of selecting the Riemann solver most suited to the local flow conditions.[94] However, this technique introduces further issues: what are the optimum flow detection and switching functions, and which Riemann solvers should be used?

**The Equilibrium Flux Method.** The equilibrium flux method[97,98] (EFM) is a flux vector splitting scheme based on kinetic theory,[99] where interface fluxes are derived by assuming a Maxwellian velocity distribution at the left and right interface states. The EFM preserves positivity and satisfies the entropy condition,[100] is substantially faster than most Riemann solvers, and is included in SF3D. Results later in this thesis confirm that the EFM is extremely robust and stable, but at the cost of excessively high diffusion levels. Recently, a hybrid EFM/FDS scheme has been proposed in an attempt to improve accuracy,[100] but introducing elements of a Riemann solver also introduces some of its associated disadvantages.

**The Advection Upwind Splitting Method.** Strictly neither FVS nor FDS, the advection upwind splitting method[91,101] (AUSM) was developed by Liou and Steffen in an at-

tempt to combine the low diffusion and accuracy of FDS with the stability and speed of FVS in a single scheme. The AUSM concept is to use different splittings for the convective fluxes and pressure terms, with each splitting being some function of an intuitively defined interface Mach number. The AUSM is almost as cheap as FVS schemes, and only slightly more dissipative than FDS schemes. Because no assumptions are made about the thermodynamics of the fluid, the AUSM is ideal for nonequilibrium gas flows and flows containing species that are not calorifically perfect or polytropic. Contrary to claims in the literature,[101, 102] results in this thesis show that the carbuncle phenomenon can be roused at some flow conditions, although we do find that the AUSM suffers this fault more rarely than FDS schemes. Perhaps the most significant failings of the AUSM are the appearance of pressure oscillations behind strong shocks, and poor damping behaviour at low Mach numbers.[89]

**Advection Upwind Splitting Method Variants.** The success of the AUSM has inspired Liou, Steffen, and other investigators to extend and modify the technique to improve its accuracy and alleviate its shortcomings. The AUSMD scheme was proposed to introduce stronger FDS characteristics, and upwinds the momentum flux according to the sign of an appropriately defined interface mass flux.[102] Contact discontinuities and fine flow features are cleanly resolved by the AUSMD, however it does suffer from postshock pressure oscillations similarly to the AUSM. The AUSMV scheme has more of an FVS flavour, and splits the momentum flux depending on signal strengths from the left and right interface states.[103] Numerical experiments show that the AUSMV scheme accurately captures shocks without oscillations, but introduces oscillatory behaviour near moving contact discontinuities.[102] The AUSMDV[102] uses a blending function to combine the AUSMD and AUSMV momentum fluxes to obtain a positivity-preserving scheme that resolves one-dimensional contact surfaces exactly and shock waves accurately, without excessive dissipation or oscillation. Unfortunately the base AUSMDV scheme is prone to the carbuncle problem, and does not satisfy the entropy condition. An AUSM variant free of carbuncles is the AUSM+.[104] This, however, comes at the cost of the reintroduction of postshock overshoots and pressure oscillations near walls.[105] The AUSM+ is claimed to exactly resolve one-dimensional contact and shock discontinuities, and satisfy the positivity condition.[106] An enhanced version of the AUSM+, AUSM+W, has been proposed to obtain better performance in flows with strong shocks.[107] A pressure-weighted scheme by Kim *et al.*,[105] AUSMPW, removes both carbuncles and pressure oscillations by blending AUSM+ and AUSMDV fluxes. The AUSMPW formulation, though, is reasonably complex and introduces the gas $\gamma$ as an extra term in the splitting. Other AUSM variants include the AUSM/van Leer hybrid of Radespiel and Kroll,[89] and the low diffusion flux

splitting method of Edwards[88,108] aimed at improving performance at low Mach numbers. The original AUSM scheme, as well as AUSMD, AUSMV, AUSMDV and AUSM+ are all implemented in SF3D; the AUSMPW was published too recently to be included in this thesis.

Explicit fixes are available for all of the entropy-violating and carbuncle-afflicted schemes (see Reference 102, for example). Fixes, however, degrade solution accuracy and often introduce tuning parameters which must be adjusted for different flow problems.[90,94] Most fixes have little mathematical or physical basis[104] and work by increasing dissipation levels. When multiple fixes are used, the interaction between them is often unknown or poorly understood. Clearly, robust schemes that do not require fixes are preferable.

### 3.3.3 Fluxes in Primitive Variable Form

A complete set of primitive variables describing the flow conditions at an interface constitutes an interface state. If a flux solver is capable of determining the interface state, the interface flux vector $\mathbf{F}_{\text{if}}$ can be directly evaluated as:

$$\mathbf{F}_{\text{if}} = \begin{bmatrix} \rho(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} \\ \rho\mathbf{u}(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} + P\hat{\mathbf{n}} - \mathbf{T}\hat{\mathbf{n}} \\ \rho E(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} + P\mathbf{u}\cdot\hat{\mathbf{n}} - \mathbf{u}\cdot(\mathbf{T}\hat{\mathbf{n}}) - \mathbf{q}\cdot\hat{\mathbf{n}} \\ \rho\mathbf{C}(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} \end{bmatrix}_{\text{if}} . \tag{3.27}$$

### 3.3.4 Fluxes in Conservative Variable Form

Not all flux solvers can be cast in a form that allows the direct calculation of interface states. Many solvers are only capable of calculating the interface fluxes of conserved variables. These are the per unit area flow rate of mass $G$, linear momentum $\mathbf{L}$ and enthalpy $H$. In terms of primitive variables, the inviscid fluxes of conserved quantities are

$$G = \rho\,\mathbf{u}\cdot\hat{\mathbf{n}}, \tag{3.28}$$

$$\mathbf{L} = \rho\mathbf{u}\,\mathbf{u}\cdot\hat{\mathbf{n}} + P\hat{\mathbf{n}}, \text{ and} \tag{3.29}$$

$$H = \rho(e + \tfrac{1}{2}|\mathbf{u}|^2)\,\mathbf{u}\cdot\hat{\mathbf{n}} + P\mathbf{u}\cdot\hat{\mathbf{n}} \tag{3.30}$$

if, and only if, the interface velocity $\mathbf{w} = \mathbf{0}$. Note that it is always possible to calculate the conserved variables from an interface state, while it is not always possible to decompose the conserved variables into primitive variables. Since it desirable to develop a CFD code

that is compatible with a wide range of flux solvers, we shall choose to write the interface flux vector in terms of conserved variables.

Most flux solvers work under the assumption of a stationary interface. This limitation can be overcome, however, by temporarily changing the stationary solution reference frame to one which moves with the interface. Working in the moving reference frame, a flux solver would not return the mass flux given by Equation 3.28, but would instead return

$$G' = \rho\,(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}}. \tag{3.31}$$

In the moving reference frame, the momentum flux would be evaluated by the flux solver as

$$\mathbf{L}' = \rho(\mathbf{u}-\mathbf{w})\,(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} + P\hat{\mathbf{n}}, \tag{3.32}$$

which, after expansion, becomes

$$\mathbf{L}' = \rho\mathbf{u}\,(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} - \rho\mathbf{w}\,(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} + P\hat{\mathbf{n}}. \tag{3.33}$$

By substituting from Equation 3.31, and rearranging, we have

$$\mathbf{L}' + G'\mathbf{w} = \rho\mathbf{u}\,(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} + P\hat{\mathbf{n}}. \tag{3.34}$$

We can deduce from Equation 3.30 that, in the moving reference frame, the enthalpy flux would be evaluated by the flux solver as

$$H' = \rho(e+\tfrac{1}{2}|\mathbf{u}-\mathbf{w}|^2)\,(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} + P(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}}, \tag{3.35}$$

which is the same as

$$\begin{aligned} H' = {} & \rho(e+\tfrac{1}{2}|\mathbf{u}|^2)\,(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} + P\mathbf{u}\cdot\hat{\mathbf{n}} - \tfrac{1}{2}|\mathbf{w}|^2\,\rho(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} \\ & - [\rho\mathbf{u}\,(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} - \rho\mathbf{w}\,(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} + P\hat{\mathbf{n}}]\cdot\mathbf{w}. \end{aligned} \tag{3.36}$$

Substitution of Equations 3.31 and 3.33 yields

$$H' + \tfrac{1}{2}G'|\mathbf{w}|^2 + \mathbf{L}'\cdot\mathbf{w} = \rho E\,(\mathbf{u}-\mathbf{w})\cdot\hat{\mathbf{n}} + P\mathbf{u}\cdot\hat{\mathbf{n}}. \tag{3.37}$$

Comparison of Equations 3.31, 3.34, and 3.37 with Equation 3.27 shows that

$$\mathbf{F}_{\text{if}} = \begin{bmatrix} G' \\ \mathbf{L}' + G'\mathbf{w} - \mathbf{T}\hat{\mathbf{n}} \\ H' + \tfrac{1}{2}G'|\mathbf{w}|^2 + \mathbf{L}'\cdot\mathbf{w} - \mathbf{u}\cdot(\mathbf{T}\hat{\mathbf{n}}) - \mathbf{q}\cdot\hat{\mathbf{n}} \\ G'\mathbf{C} \end{bmatrix}_{\text{if}}. \tag{3.38}$$

Hence it is possible to calculate the flux vector for a moving interface using a general conserved-variable stationary-interface flux solver . Moreover, the interface flux vector is written in a form which requires no knowledge of the interface primitive variables, with the exception of the diffusive terms. The treatment of the diffusive terms has already been described in Section 2.8.

### 3.3.5 The AUSMDV

The advection upwind splitting method combining flux differencing and vector splitting (AUSMDV) is fully defined in Reference 102. Since the scheme is used for many of the simulations in this thesis, we now repeat it for multiple dimensions in a form consistent with Section 3.3.4.

We first define $u_{\mathrm{L}}$ and $u_{\mathrm{R}}$ as the normal velocity components at the left and right interface states, in the interface frame of reference:

$$u_{\mathrm{L}} = (\mathbf{u}_{\mathrm{L}} - \mathbf{w}) \cdot \hat{\mathbf{n}}, \text{ and} \tag{3.39}$$

$$u_{\mathrm{R}} = (\mathbf{u}_{\mathrm{R}} - \mathbf{w}) \cdot \hat{\mathbf{n}}. \tag{3.40}$$

The remaining tangential velocity vectors are then

$$\mathbf{v}_{\mathrm{L}} = (\mathbf{u}_{\mathrm{L}} - \mathbf{w}) - u_{\mathrm{L}}\hat{\mathbf{n}}, \text{ and} \tag{3.41}$$

$$\mathbf{v}_{\mathrm{R}} = (\mathbf{u}_{\mathrm{R}} - \mathbf{w}) - u_{\mathrm{R}}\hat{\mathbf{n}}. \tag{3.42}$$

The interface mass flux is a scalar quantity given by the vector splitting

$$G' = u_{\mathrm{L}}^{+}\rho_{\mathrm{L}} + u_{\mathrm{R}}^{-}\rho_{\mathrm{R}}, \tag{3.43}$$

with the individual splitting terms defined by

$$u_{\mathrm{L}}^{+} = \begin{cases} \alpha_{\mathrm{L}}\left[\dfrac{(u_{\mathrm{L}}+a_{\mathrm{m}})^2}{4a_{\mathrm{m}}} - \dfrac{u_{\mathrm{L}}+|u_{\mathrm{L}}|}{2}\right] + \dfrac{u_{\mathrm{L}}+|u_{\mathrm{L}}|}{2} & \text{if } \dfrac{|u_{\mathrm{L}}|}{a_{\mathrm{m}}} \leq 1, \\ \dfrac{u_{\mathrm{L}}+|u_{\mathrm{L}}|}{2} & \text{otherwise.} \end{cases} \tag{3.44}$$

$$u_{\mathrm{R}}^{-} = \begin{cases} \alpha_{\mathrm{R}}\left[-\dfrac{(u_{\mathrm{R}}-a_{\mathrm{m}})^2}{4a_{\mathrm{m}}} - \dfrac{u_{\mathrm{R}}-|u_{\mathrm{R}}|}{2}\right] + \dfrac{u_{\mathrm{R}}-|u_{\mathrm{R}}|}{2} & \text{if} \dfrac{|u_{\mathrm{R}}|}{a_{\mathrm{m}}} \leq 1, \\ \dfrac{u_{\mathrm{R}}-|u_{\mathrm{R}}|}{2} & \text{otherwise.} \end{cases} \tag{3.45}$$

The common speed of sound at the interface, $a_{\mathrm{m}}$, is selected as

$$a_{\mathrm{m}} = \max(a_{\mathrm{L}}, a_{\mathrm{R}}) \tag{3.46}$$

in this work. The $\alpha$ are functions designed to avoid dissipation at contact discontinuities, and given by

$$\alpha_{\mathrm{L}} = \frac{2P_{\mathrm{L}}/\rho_{\mathrm{L}}}{P_{\mathrm{L}}/\rho_{\mathrm{L}} + P_{\mathrm{R}}/\rho_{\mathrm{R}}}, \text{ and} \tag{3.47}$$

$$\alpha_{\mathrm{R}} = \frac{2P_{\mathrm{R}}/\rho_{\mathrm{R}}}{P_{\mathrm{L}}/\rho_{\mathrm{L}} + P_{\mathrm{R}}/\rho_{\mathrm{R}}}. \tag{3.48}$$

The enthalpy flux is simply upwinded by the sign of the mass flux as

$$H' = \tfrac{1}{2}[G'(h_{\mathrm{L}} + h_{\mathrm{R}}) - |G'|(h_{\mathrm{R}} - h_{\mathrm{L}})], \tag{3.49}$$

where the left and right enthalpies, in a moving interface reference frame, are

$$h_{\mathrm{L}} = e_{\mathrm{L}} + P_{\mathrm{L}}/\rho_{\mathrm{L}} + \tfrac{1}{2}|\mathbf{u}_{\mathrm{L}} - \mathbf{w}|^2 \tag{3.50}$$

$$h_{\mathrm{R}} = e_{\mathrm{R}} + P_{\mathrm{R}}/\rho_{\mathrm{R}} + \tfrac{1}{2}|\mathbf{u}_{\mathrm{R}} - \mathbf{w}|^2. \tag{3.51}$$

We break the interface momentum flux into normal and tangential components,

$$\mathbf{L}' = L'_{\mathrm{n}}\hat{\mathbf{n}} + \mathbf{L}_{\mathrm{t}}, \tag{3.52}$$

and upwind the tangential components by the sign of the mass flux:

$$\mathbf{L}'_{\mathrm{t}} = \tfrac{1}{2}[G'(\mathbf{v}_{\mathrm{L}} + \mathbf{v}_{\mathrm{R}}) - |G'|(\mathbf{v}_{\mathrm{R}} - \mathbf{v}_{\mathrm{L}})]. \tag{3.53}$$

The normal momentum flux is calculated as a mixture of the AUSMD and AUSMV momentum fluxes through the expression

$$L'_{\mathrm{n}} = (\tfrac{1}{2} + s)L'_{\mathrm{V}} + (\tfrac{1}{2} - s)L'_{\mathrm{D}} + P_{1/2}. \tag{3.54}$$

The AUSMV momentum flux is given by the splitting

$$L'_{\mathrm{V}} = \rho_{\mathrm{L}} u_{\mathrm{L}} u_{\mathrm{L}}^{+} + \rho_{\mathrm{R}} u_{\mathrm{R}} u_{\mathrm{R}}^{-}, \tag{3.55}$$

while the AUSMD momentum flux is fully upwinded:

$$L'_{\mathrm{D}} = \tfrac{1}{2}[G'(u_{\mathrm{L}} + u_{\mathrm{R}}) - |G'|(u_{\mathrm{R}} - u_{\mathrm{L}})]. \tag{3.56}$$

The switching factor $s$ is made sensitive to the pressure gradient through the expression

$$s = \tfrac{1}{2}\min\left[1, \frac{K|P_{\mathrm{R}} - P_{\mathrm{L}}|}{\min(P_{\mathrm{L}}, P_{\mathrm{R}})}\right], \tag{3.57}$$

where the sensitivity constant $K$ is nominally set to 10. Finally, the interface pressure term is

$$P_{1/2} = P_{\mathrm{L}}^{+} + P_{\mathrm{R}}^{-}, \tag{3.58}$$

where the individual pressure splittings are given by the polynomial

$$P^{\pm}_{\mathrm{L/R}} = \begin{cases} \frac{1}{4}P_{\mathrm{L/R}}\left(\frac{u_{\mathrm{L/R}}}{a_{\mathrm{m}}} \pm 1\right)^2 \left(2 \mp \frac{u_{\mathrm{L/R}}}{a_{\mathrm{m}}}\right) & \text{if } \frac{|u_{\mathrm{L/R}}|}{a_{\mathrm{m}}} \leq 1, \\ P_{\mathrm{L/R}} \frac{u_{\mathrm{L/R}} \pm |u_{\mathrm{L/R}}|}{2u_{\mathrm{L/R}}} & \text{otherwise.} \end{cases} \tag{3.59}$$

An alternative to these second-order pressure splittings are the original first-order AUSM pressure splittings:[101]

$$P^{\pm}_{\mathrm{L/R}} = \begin{cases} \frac{1}{2}P_{\mathrm{L/R}}\left(1 \pm \frac{u_{\mathrm{L/R}}}{a_{\mathrm{m}}}\right) & \text{if } \frac{|u_{\mathrm{L/R}}|}{a_{\mathrm{m}}} \leq 1, \\ P_{\mathrm{L/R}} \frac{u_{\mathrm{L/R}} \pm |u_{\mathrm{L/R}}|}{2u_{\mathrm{L/R}}} & \text{otherwise.} \end{cases} \tag{3.60}$$

## 3.4 Boundary Conditions

The constraints applied to the boundaries of the solution domain comprise a major part of the definition of a flow problem. To avoid having to explicitly set the fluxes at the domain boundaries, we use external ghost cells containing appropriate flow conditions to cause the reconstruction scheme and flux solver to calculate the correct boundary fluxes. Because all the second- and third-order reconstruction schemes in SF3D use, at most, a four-cell stencil, only two ghost cells are needed at each boundary. The SF3D code has a switch to enable more or fewer ghost cells, however, should different reconstruction schemes be required. A two-dimensional schematic of the ghost cell layout is shown in Figure 3.3. The two ghost cells are denoted by G, and cells in the flow domain are denoted by F.

For an inflow boundary condition, the ghost cells adjacent to the boundary are simply set to the desired fluid inflow state and velocity. A zero-order extrapolation is used for outflow boundary conditions, whereby the fluid state and velocity in the ghost cells adjacent to the outflow boundary are set equal to that of the internal cell nearest the boundary. This kind of extrapolation is accurate only when the outflow cell face advection Mach number normal to the boundary is greater than unity.

For a wall boundary condition, we need to select ghost cell conditions that generate the correct wall pressure and ensure a zero mass flux through the boundary. Even though there is no flow through the wall, the calculation of the correct wall pressure is important since it affects the Navier-Stokes equations via the momentum flux term. Without reconstruction, the solution to the Riemann problem at the boundary will yield the correct wall fluxes if we specify that the boundary ghost cell contains gas at the same state of the flow next to

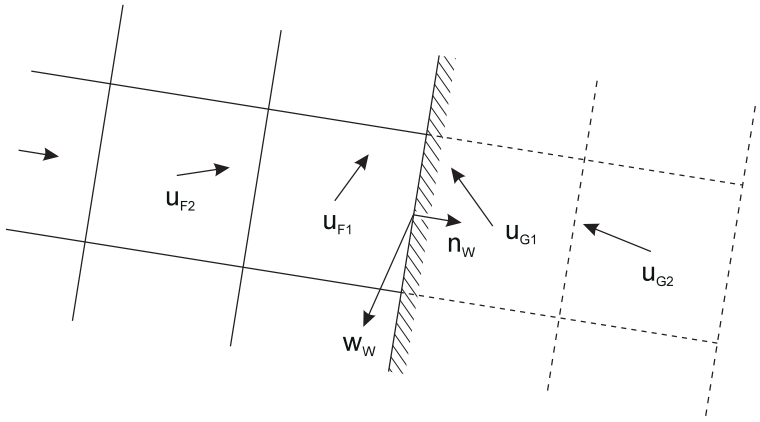



Figure 3.3: Ghost cell configuration for a slip wall condition.

the wall, with a reversed normal velocity component. Thus, the velocity condition

$$(\mathbf{u}_{F1} - \mathbf{w}_W) \cdot \hat{\mathbf{n}} = -(\mathbf{u}_{G1} - \mathbf{w}_W) \cdot \hat{\mathbf{n}} \tag{3.61}$$

is applied, which is applicable to both moving and stationary walls. For a frictionless (or slip) wall, such as the one in Figure 3.3 velocity components tangential to the boundary should be conserved:

$$\mathbf{u}_{F1} - (\mathbf{u}_{F1} \cdot \hat{\mathbf{n}})\,\hat{\mathbf{n}} = \mathbf{u}_{G1} - (\mathbf{u}_{G1} \cdot \hat{\mathbf{n}})\,\hat{\mathbf{n}}. \tag{3.62}$$

Note that this equation is correctly independent of the wall velocity. Combining Equations 3.61 and 3.62 and solving for the ghost cell velocity yields

$$\mathbf{u}_{G1} = \mathbf{u}_{F1} - 2[(\mathbf{u}_{F1} - \mathbf{w}_W) \cdot \hat{\mathbf{n}}]\,\hat{\mathbf{n}}. \tag{3.63}$$

For higher order schemes using a larger stencil, the analysis is extended to the outer ghost cell:

$$\mathbf{u}_{G2} = \mathbf{u}_{F2} - 2[(\mathbf{u}_{F2} - \mathbf{w}_W) \cdot \hat{\mathbf{n}}]\,\hat{\mathbf{n}}. \tag{3.64}$$

For a non-slip wall, viscous stresses are properly calculated when the tangential ghost cell velocity in the wall frame of reference is zero. This condition can be expressed as

$$\mathbf{u}_{G1} - \mathbf{w}_W - [(\mathbf{u}_{G1} - \mathbf{w}_W) \cdot \hat{\mathbf{n}}]\,\hat{\mathbf{n}} = \mathbf{0}. \tag{3.65}$$

Combining Equations 3.61 and 3.65, we have the inner and outer ghost cell velocities

$$\mathbf{u}_{G1} = \mathbf{w}_W - [(\mathbf{u}_{F1} - \mathbf{w}_W) \cdot \hat{\mathbf{n}}]\,\hat{\mathbf{n}}, \text{ and} \tag{3.66}$$

$$\mathbf{u}_{G2} = \mathbf{w}_W - [(\mathbf{u}_{F2} - \mathbf{w}_W) \cdot \hat{\mathbf{n}}]\,\hat{\mathbf{n}}. \tag{3.67}$$

Additional to stationary non-slip walls, this ghost cell formulation is also suitable for sliding walls with friction, such as a piston or moving plate.

With regards to heat transfer at walls, SF3D supports adiabatic and isothermal options. An adiabatic wall is one which is fully insulated and accommodates no heat transfer. Ghost cell temperatures are set equal to the wall flow temperature in this case. Isothermal walls allow heat conduction; they have an infinite heat capacity and thus remain at constant temperature. For an isothermal boundary condition, ghost cell temperatures are set to the isothermal wall temperature. There is no support for wall catalysis or radiative heat transfer at present.

## 3.5 The Apollo Heat Shield

One of the first cases to which SF3D was applied was the simulation of flow around various blunt bodies in carbon dioxide. The simulations were performed in parallel with experiments aimed at simulating drag measurements for heat shields to be used during the atmospheric entry of Mars.[109] The work was motivated by the then future NASA Mars environment survey pathfinder missions.[110–112] Details and comparison of the simulation and experimental results are presented in Section 5.9.

The forebody of the Apollo capsule was one of the blunt body geometries tested. A three-dimensional wireframe model of the body and its corresponding axisymmetric, orthogonal, computational grid are shown in Figure 3.4. The full grid size is 60 cells in the body normal direction and 200 cells in the tangential direction, and is scaled so that the heat shield radius is 10 mm, matching the size of the model tested in the wind tunnel experiments.

The simulation results shown in Figure 3.5 correspond to a supersonic flow speed of 2 km/s with a free-stream density and temperature of 0.032 kg/m$^3$ and 3240 K respectively. Viscous effects were not included, and the equilibrium carbon dioxide model was used. The simulation was run at second-order spatial accuracy using MUSCL reconstruction, and second-order temporal accuracy using Runge-Kutta timestepping with a CFL number of 0.5. Fluxes were calculated with the AUSMDV. Additional to the contour plots of pressure and density, a computational interferogram is also used to visualize the simulation results. Further details on both computational and experimental interferometry techniques are included in Appendix B.

The computed flow field of Figure 3.5 is typical of supersonic blunt-body flows, as discussed in Section 1.1. A relatively clean solution is observed, except for small amounts of noise occurring in the post-shock region, and some oscillations in density near the stagnation line and body. It is suspected that the noise in the stagnation region is due to the grid singularity at the axis of symmetry.

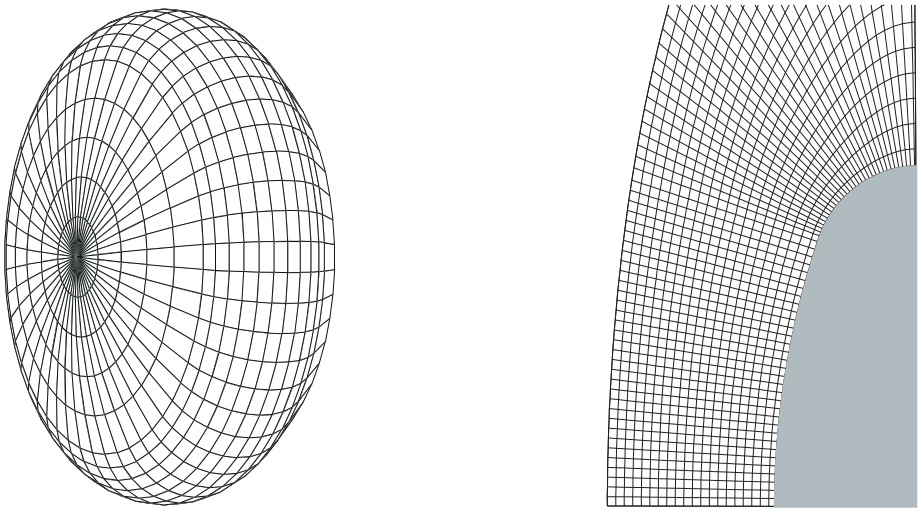


Figure 3.4: A three-dimensional rendering of the Apollo heat shield (left) and solution grid (right). For clarity, only every third grid line is shown.

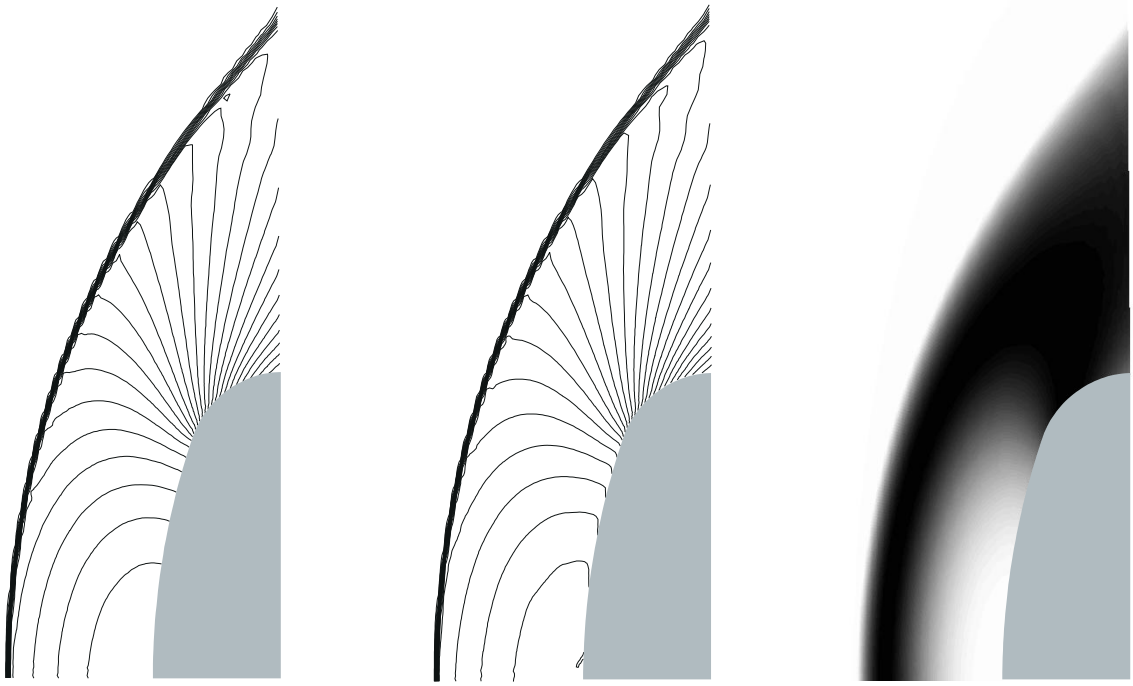


Figure 3.5: Carbon dioxide flow around the Apollo heat shield at supersonic speed. Isopressure contours (left), isopycnic contours (centre), and a computational interferogram (right) are shown.

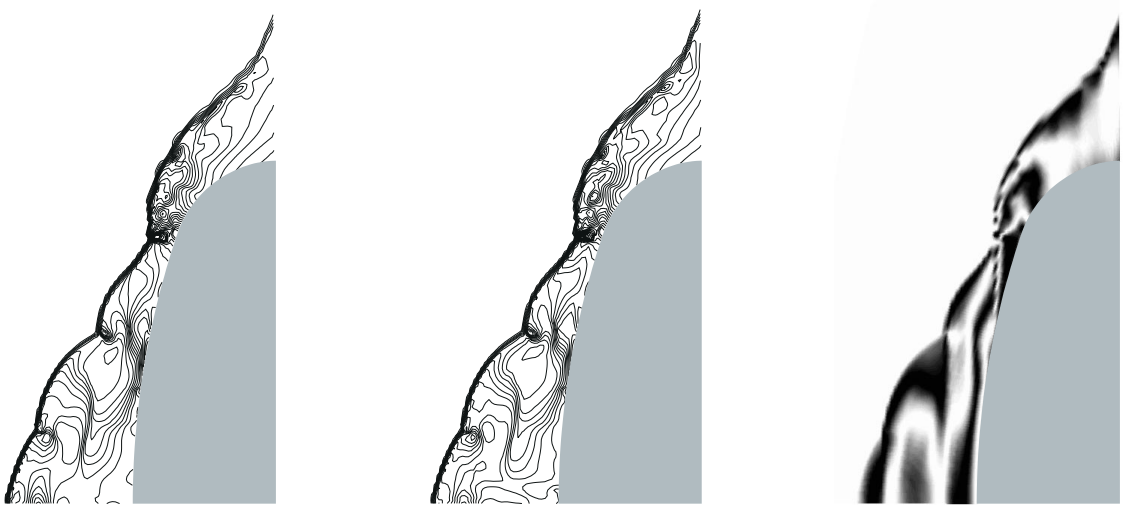


Figure 3.6: Computed unstable flow around the Apollo heat shield at hypersonic speed. Isopressure contours (left), isopycnic contours (centre), and a computational interferogram (right) are shown.

To test the heat shield at a more realistic entry speed, another simulation was performed using a hypersonic free stream of 7.57 km/s. Figure 3.6 shows the computed solution. Alarmingly, the results show a chronically unstable shock wave and a solution that does not converge in time. The bow shock wave oscillates temporally and spatially at different frequencies, and a number of vortices are present in the post-shock flow around the body. It is not immediately clear whether the observed behaviour can be deemed a numerical instability, or whether it is a physical phenomenon. Certainly, the shock wave shape does not match that of the well-known carbuncles discussed in the literature,[94, 104] primarily due to the downstream-propagating cellular structures. The unstable flow was independently reproduced using a second CFD code, which indicates that it is unlikely to be a consequence of programming error.[113] Introducing viscous terms into the simulation does not avert the instability; this is consistent with physical instabilities, according to Bashkirov, who states that (with the exception of a neutral stability region) stability bounds are unaffected by viscous stresses.[114]

An initial attempt to determine the nature of the instability was through consideration of shock wave stability theory. Original work by D'yakov,[115] later clarified by Swan and Fowles,[116] used a linearized analysis to determine criteria for the stability of an infinite planar shock wave subjected to a sinusoidal perturbation. According to D'yakov's theory and experimental results, no perfect gas admits shock wave instability.[117] The analysis also states that instability is possible in a medium of arbitrary equation of state, and is dictated by the slope of the shock Hugoniot, mass flux per unit area, and Mach number.

For a stationary shock, the theoretical requirements for instability are

$$A < -1, \text{ or } A > 1 + 2M, \text{ where } A = (\rho u)^2 \left( \frac{dv}{dP} \right)_H . \tag{3.68}$$

Here, $M$ is the downstream Mach number, and $(dv/dP)_H$ is the gradient of the Hugoniot.

The free-stream flow conditions used in the unstable Apollo heat shield simulation do not satisfy the instability criteria in Equation 3.68. To further investigate the sensitivity of the simulated shock wave stability to flow conditions, an array of simulations were run at flow speeds of up to 8 km/s. Simulation results showed that shock stability was independent of the local Hugoniot gradient, and instability was observed at all flow speeds above 3.2 km/s.[118]

Similarly, shock tunnel tests by Griffiths *et al.*[119] show experimental evidence of plane shock wave instability in carbon dioxide flows at conditions where Equation 3.68 does not indicate unstable flow. The results in Reference 119 do indicate, though, that physical instability is more likely to be observed when $A$ becomes large, and is at its closest to the upper instability bound. It is a possibility that test flow noise within the experimental facility (such as of the type described in Reference 120, for example) contributed to the early occurrence of instability. Another plausible explanation is that the post shock recombination of ionized or dissociated species causes a sudden release of energy or acoustic emission, triggering an instability.[121–124]

The equilibrium carbon dioxide model used for the numerical simulations does not model ionization nor the kinetics involved in post-shock recombination. Further, the physical free-stream noise encountered in the experiments does not exist in the simulations. Hence, since there is no early (physical) instability triggering mechanism, and the simulated instability does not correlate with the gradient of the Hugoniot nor satisfy the theoretical instability requirements, it would seem reasonable to postulate that the instability observed in Figure 3.6 is actually a numerical artifact.

Although it is not likely that the instability is real, we take the occurrence of numerical instability to indicate there may be some propensity towards instability in the physical system. This could be manifested in an increased time for the bow shock to recover from a perturbation, for example. It is also suggested that physical instability modes are naturally emulated when numerical instability occurs. Thus, it can be a difficult task to differentiate between numerical and physical instabilities, and in some cases an accurate experiment is the only way to provide final verification. The holographic interferogram[125] in Figure 3.7 was produced in the University of Queensland X1 expansion tube,[126] for carbon dioxide flow around the Apollo heat shield with free-stream conditions corresponding to those listed earlier in this section. A laser of wavelength 532 nm was used to produce the image. A stable shock is observed, confirming our postulate that the simulated instability is spurious. The image is partly obscured by a damaged optical window.

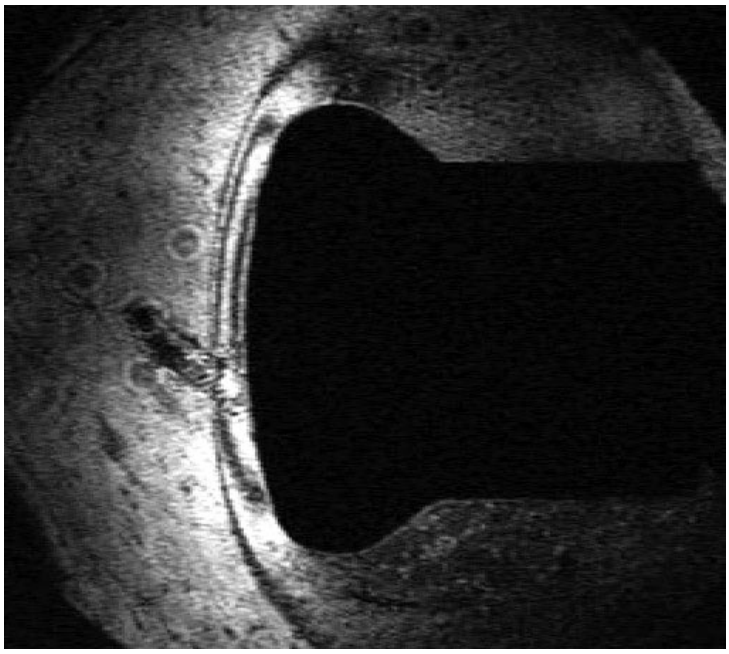


Figure 3.7: Experimental interferogram of flow around the Apollo heat shield. Source: University of Queensland laser physics group.

## 3.6 Investigation of Numerical Instability

Before investigating the bow shock instability, we shall first study a related problem in order to gain more insight into its possible cause. A perturbed grid test case proposed by Quirk[94] exposes an odd-even decoupling problem suffered by some flux solvers. A normal shock wave, initially aligned with a two-dimensional rectangular grid, is examined as it propagates through the mesh and processes a volume of stationary gas. Instead of being perfectly orthogonal, the grid geometry is contrived to induce shock instability; every second cell vertex on the grid centreline in perturbed by a small distance. Specifically, we use a Mach 10 shock processing air at standard temperature and pressure (300 K, 100 kPa), with grid perturbations of magnitude $1 \times 10^{-6}$ mm in the transverse direction. The grid cells are nominally 2.5 mm square, and 40 cells span the 0.1 m shock tube width.

Figure 3.8 shows density contours of the flow solution after 250 $\mu$s, which corresponds to roughly 1000 steps at CFL 0.5. To avoid introducing the effects of limiters and interpolation schemes, only first order spatial and temporal accuracy is used. The results show that EFM, the most dissipative scheme considered in this thesis, resolves the normal shock without noise or instability. The AUSM, which is less dissipative than EFM, also resolves the normal shock well, with only very small density oscillations visible towards the rear of the shock structure. In contrast, the AUSMDV generates a shock wave with a centreline protuberance, as well as significant post shock noise. The approximate Rie-

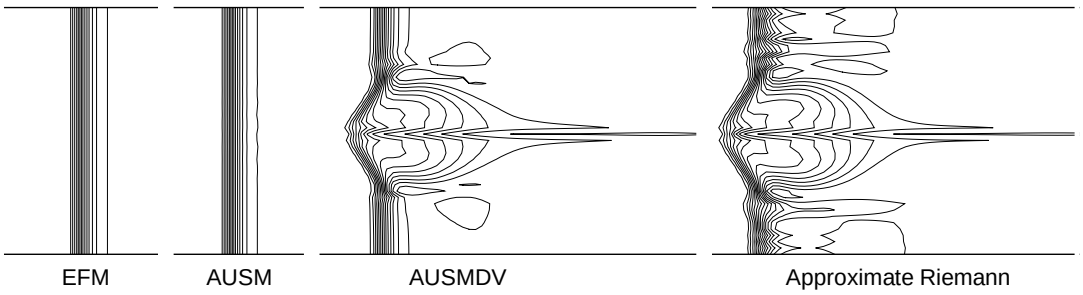



Figure 3.8: The odd-even decoupling test case. Density contours show a shock instability generated by some flux solvers.

mann solver, which is the least dissipative scheme of the four, suffers an instability of slightly greater magnitude than the AUSMDV, with plenty of downstream noise apparent.

To date, no single conclusive explanation for the problem has been put forward. However, an approximate analysis by Quirk[94] and the hypothesis of Wada and Liou[102] provide interpretations that properly predict characteristics of the instability. It is Quirk's contention that the behaviour is a result of two competing processes. The first is an odd-even decoupling of pressure and density that exists in some schemes, where pressure disturbances continually feed density disturbances if the system is repeatedly perturbed and the disturbances are of opposite sign. Presumably, the disturbances are generated by small errors in resolving the fluxes across the interfaces which are almost, but not aligned with the flow. This odd-even decoupling can be observed in Figure 3.9, where the post-shock density and pressure profiles show disturbances with opposite sign and out of phase. A second contributing process is the variation in shock speed along the length of the shock, which is attributable to the variations in post-shock conditions caused by the first process. The different shock speeds amplify the pressure perturbations and reinforce the instability.

Liou and Wada[107] theorize that the instability is due to the internal structure of the shock. When a shock wave is resolved within the computational grid, the schemes discussed thus far will approximate (or capture) it as a blurred region of high gradient, rather than a perfect discontinuity. The numerical shock will thus contain an internal structure, and usually span several cells. For example, the first-order EFM simulation in Figure 3.8 resolves the bulk of the shock within four cells. Unphysical transfer of information between these cells in the transverse direction is promoted by the perturbed grid, and if not sufficiently damped by the numerical scheme, may be blamed for the occurrence of instability. The transverse flow of information is eventually manifested in large values of transverse velocity, as shown in Figure 3.10. The plot shows high $u_y$ oscillations directly behind the perturbed part of the shock, which progressively dampen with increased

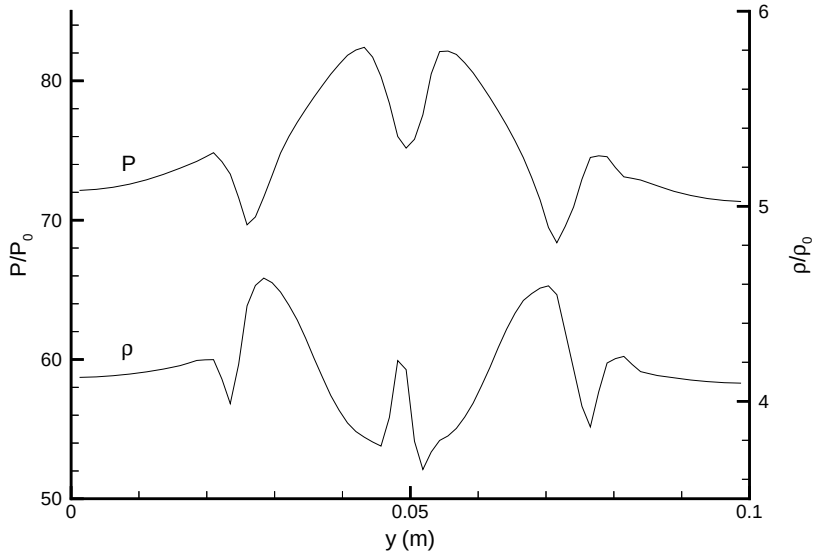



Figure 3.9: A transverse section of the odd-even decoupling test case, showing density and pressure decoupling. The section is taken 3 cells (7.5 mm) after the shock front.

proximity from the shock front.

For schemes suffering the odd-even decoupling problem, Quirk, Wada and Liou all suggest the introduction of a more dissipative flux solver when a shock is detected. Wada and Liou show that this technique also alleviates bow shock carbuncles. Increased dissipation comes at the cost of thicker shocks and degraded accuracy, and the introduction of adaptive flux solvers based on shock-detectors is not an attractive option. We have again arrived at the tradeoff between accuracy, stability, and scheme simplicity.

Armed with insight into the odd-even decoupling problem, we now return to investigating the unstable bow shock. Figure 3.11 shows a typical time history of the development of bow shock carbuncles. The heat shield is initially immersed in gas at rest, and flow is started by an incident shock propagating from left to right in Figure 3.11(a). The plot contains an exponentially distributed set of transverse velocity contours, and shows small amounts of post shock noise. Most of the noise is of magnitude $1 \times 10^{-5}$, and noise is also present in the state variables and in $u_x$. Note that there are additional post-shock perturbations near the simulation centreline, where the grid is most closely aligned with the flow. The centreline noise is also amplified by the grid singularity along the axis of axisymmetry.

It is supposed that the post-shock noise is probably not caused by the same processes that account for the odd-even decoupling. Examination of the noise shows that pressure and density perturbations are in phase. More likely, the noise is due to changes in the

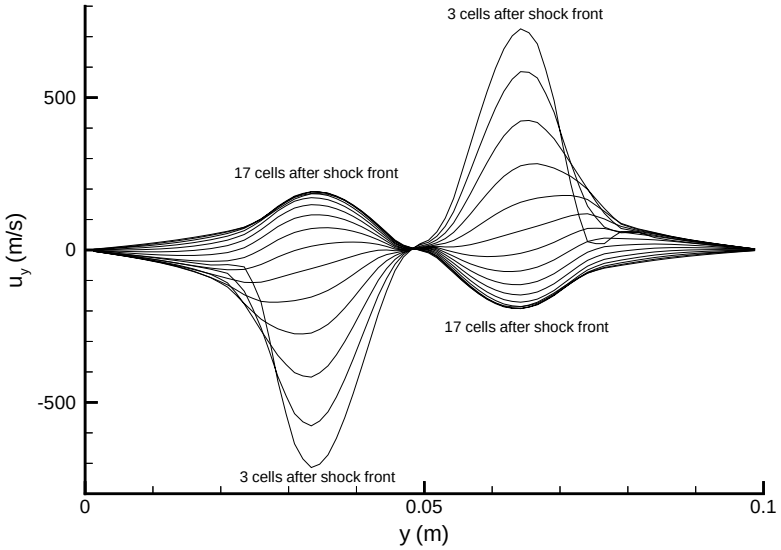



Figure 3.10: Post-shock transverse velocity perturbations.

captured shock width as it passes through the grid.[127] If the shock is positioned directly between cells at a particular timestep, it will be captured at the cell interface relatively cleanly, and be defined by the two neighbouring cells. If aligned with a cell centre, the shock may be blurred over three or more cells. Thus, the shock may strengthen or weaken, depending on where it falls at the end of a timestep. This approximately random behaviour leaves a downstream trail of noise.

After the incident shock hits the body and reflects, the remnant noise is subsequently processed by the newly formed bow shock. If we take this noise as the analog of the grid misalignment perturbations in the odd-even decoupling problem, it is no surprise that a protuberance is soon formed. It is also unsurprising that the protuberance forms in the region of highest inflow noise and closest grid alignment (and thus least numerical diffusion). The development of the first carbuncle is visible in Figure 3.11(b). Note that the carbuncle is also observed to develop in impulsively started simulations, where no incident shock (and associated downstream noise) is present. For impulsively started simulations, the newly formed bow shock wave will initially travel slowly across the mesh towards its steady-state position. The generation of spurious numerical oscillations behind slowly moving shocks is a well-known and much researched phenomenon,[90, 128, 129] and is a problem experienced by all standard shock capturing schemes to some degree.[70] It is postulated that these oscillations, combined with the axisymmetry singularity, are enough to trigger the carbuncle instability in the case of an impulsively started flow. More artificial dissipation is the only presently available cure for noise behind slowly moving

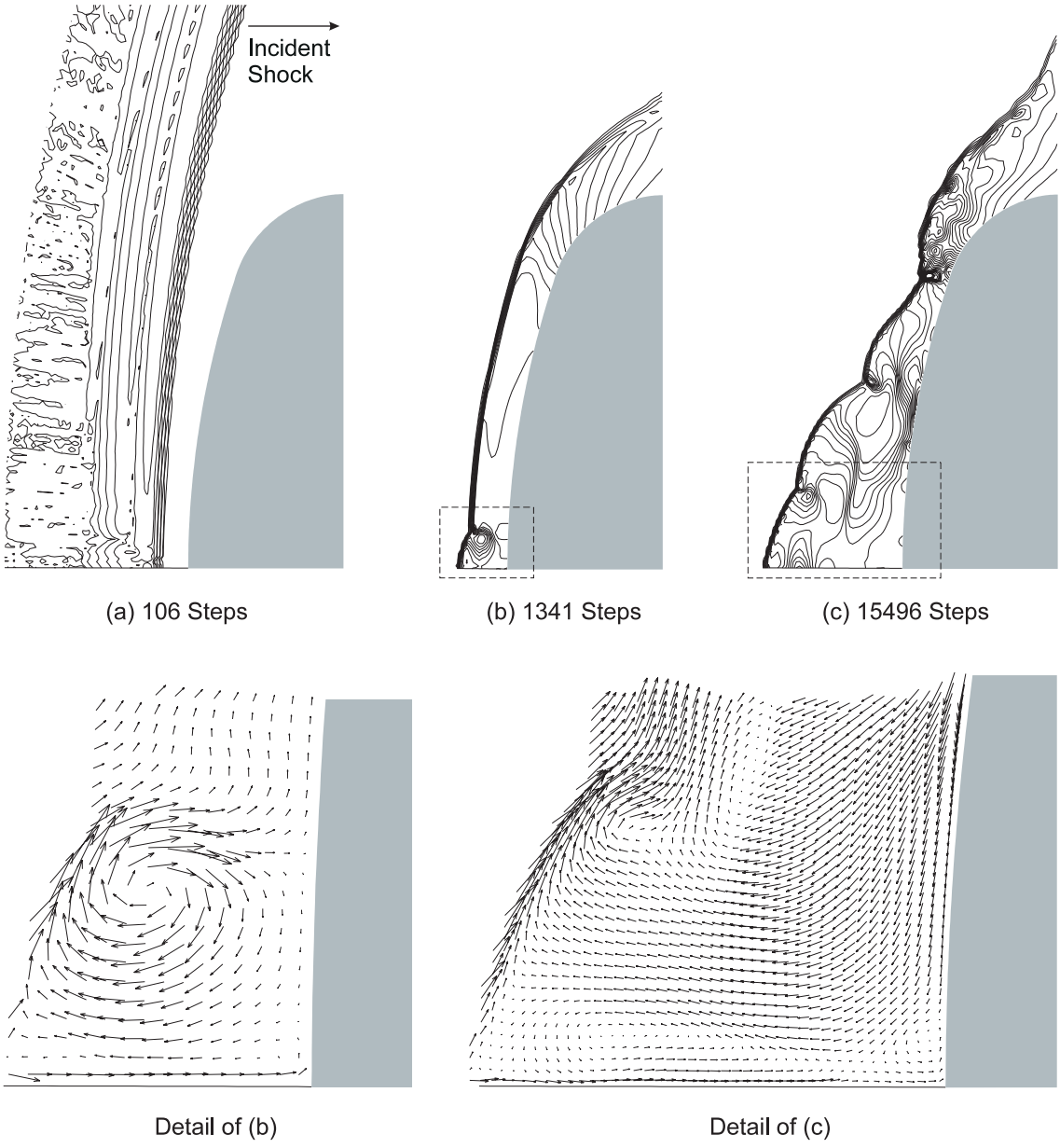



Figure 3.11: Time development of carbuncles. (a) Exponentially distributed contour plot of absolute $u_y$ during incident shock impingement on the Apollo heat shield. (b),(c) Isopressure contours showing carbuncle development on the Apollo heat shield, with magnified views of the corresponding velocity vectors shown below.

captured shocks.[70]

The post-shock perturbations in $u_y$ observed near the protuberance in the odd-even decoupling problem (Figure 3.10) are also apparent in the carbuncle instability. The magnified view of the stagnation region of Figure 3.11(b) shows velocity vectors with substantial components of $y$ velocity. A very small spurious vortex, the first to appear in the simulation, is visible in the four cells closest to the stagnation point. This initial vortex is thought to have caused the generation of the larger, counter-rotating vortex structure, closer to the shock. Since the magnitude of vorticity produced by a shock depends largely on the magnitude of the tangential velocity component,[130,131] we would expect vortex severity to increase with increasing levels of upstream $u_y$ noise, and with increasing $u_y$ noise generated inside the structure of captured shock waves. Conversely, a larger downstream vortex will cause a greater shock protuberance, and thus more $u_y$ perturbation. Thus, via a feedback mechanism involving the shock-perturbing vortex and tangential velocity, the instability is self-sustaining.

The shock-vortex structure in Figure 3.11(b), while not physical, still bears a distinct resemblance to shock-vortex interactions described in the literature.[132] Rusak *et al.*[133] cite experiments which show shock-vortex systems generating downstream regions of compressions, rarefactions and acoustic waves. In numerical and physical experiments of mildly disturbed hypersonic blunt-body flows, Hornung[134] has observed a shear layer instability that causes the production of shock-perturbing and self-reinforcing vortices. This activity is perhaps analogous to the downstream noise occurring in the numerical system here.

After a large number of steps, the bow shock wave becomes fully unstable. Figure 3.11(c), and the corresponding magnified view of velocity vectors, show the presence of many vortices and a shock wave which is unstable along its complete length. The vortices, shock-vortex interactions, and recirculation continue unabated, and steady-state convergence is not achieved.

Numerical experiments revealed factors that induced or contributed to the carbuncle problem. Stronger shocks and higher grid resolution were found to increase the chance of instability. Grids with distorted, high aspect ratio cells[135] (elongated in the body-tangential direction) were less prone to carbuncles, and although crude, this is a possible cure. Both non-reacting and equilibrium flows were tested, with the equilibrium flows being far more likely to generate carbuncles. The problem could be induced in non-reacting flows, however, at very high grid resolutions. Similarly to the odd-even decoupling results, less dissipative solvers were more prone to the instability. While the AUSM, AUSMDV, and approximate Riemann solver all produced carbuncles on at least one test case, only the approximate Riemann solver suffered the problem at relatively low grid resolutions.

To summarize, we have so far reported the following options to avoid computing carbuncles on blunt body flows with the finite-volume method: (i) Use a coarse or distorted grid, (ii) Use a highly dissipative flux solver, or (iii) Use an accurate flux solver for the bulk of the flow field, but switch to highly dissipative fluxes at the shock. Clearly, none of these options are desirable since they all involve loss of solution accuracy to some extent. Options (ii) and (iii) involve the introduction of artificial viscosity purely to ease computation, an idea that was first proposed by von Neumann and Richtmyer[136] in 1949 and is even now the subject of vigorous research and complex treatment.[77,137,138] A stable and carbuncle-free technique that avoids artificial dissipation at the shock altogether would certainly be preferred. Such a technique is presented in the next chapter.

CHAPTER 4

# Shock Fitting

Shock capturing techniques resolve shocks and other discontinuities across several cells. A natural consequence of shock capturing, then, is the formation of an internal shock structure. For the inviscid Navier-Stokes equations, internal shock structure has no physical basis and is a simply a side-effect encountered in the numerical production of a shock wave. Highly dissipative flux solvers generally produce thicker, more diffuse shocks than numerical schemes exhibiting low dissipation. Although resolving shocks more crisply, low dissipation solvers generate shocks with increased numerical noise. Spurious, shock-induced noise can range from an annoyance, to full instabilities which cause code failure. Relationships between numerical shock structure and instability were established in Chapter 3, and we would like to avoid them both.

Physical shocks in continuum fluids are generally of the order of three to ten molecules wide. In contrast, the width of numerically captured shocks is incorrectly grid dependent and varies directly with grid spacing. For most practical fluid dynamics problems, such behaviour yields computed shocks of macroscopic width. In contrast, the only mathematically correct solution of the inviscid Navier-Stokes equations for a shock wave in a continuum fluid is a perfect discontinuity, and for viscous fluids the correct solution closely approaches this limit.

Hence, we would like to avoid the calculation of shocks of finite thickness for a number of reasons: numerical stability, mathematical correctness, and physical similitude. Adaptive clustering techniques provide one way to significantly reduce shock thickness and maintain stability. By clustering extra cells around a shock wave and applying a dissipative flux solver, the shock will appear to be more crisp. The dissipative flux solver

preserves stability, and its inaccuracy is compensated by increased grid resolution. Note that the shock wave structure is still present and continues to scale with grid size, it is effectively just hidden from view. The clustering technique also taxes computer resources, since an increased number of cells and cumbersome grid management scheme are required.[139]

The numerically clean and mathematically impeccable[70] method of shock fitting eliminates shock thickness altogether, and usually requires less computational effort while achieving higher accuracy than shock capturing. The method operates by aligning grid cells to the shock wave, and solving the exact Rankine-Hugoniot relations[140] describing the variation in flow conditions across the jump. Grid alignment essentially transforms the multi-dimensional shock wave into a one-dimensional problem with a transverse velocity component. Thus we have the added advantage of avoiding the need for multidimensional reconstruction and flux solvers at the shock: in three-dimensional simulations the shock wave collapses to a surface, and in two-dimensional simulations the shock collapses to a curve.

For the blunt-body problem, the outermost boundary of the solution grid can be aligned with the bow shock wave. In this case, no cells are required in the free stream where the inflow conditions are supersonic and constant. The extra cells that would ordinarily have been required to resolve the shock, and those upstream of the shock, can be better exploited in the region between shock and body. Such savings in computational cost are particularly important for the computationally intensive three-dimensional, time-accurate, viscous simulations with chemistry considered later in this thesis. Figure 4.1 shows a comparison of shock fitted and captured solutions for the inviscid flow of calorifically perfect, chemically frozen air around a cylinder. The free-stream temperature and density are 300 K and 0.01 kg/m$^3$ respectively, and the flow speed is 3 km/s. Second-order spatial reconstruction is used on a relatively coarse grid, with a resolution of 15 cells in the body-normal direction, and 25 cells in the tangential direction. Wasted free-stream cells and a thick shock are clearly visible in the shock-captured result, as is the aliasing of the shock when it becomes misaligned with the grid and jumps between cells. Noise is apparent directly behind the shock, and in the stagnation region. Also, an indenture has been generated where the captured shock hits the axis of symmetry. The shock-fitting solution exhibits smooth contours in the shock layer, properly resolves the shock as a discontinuity, and encounters no such defects. Additionally, fitted shocks are not susceptible to the odd-even decoupling discussed in Section 3.6 and, later in this chapter, it will be shown that shock fitting can cure the carbuncle phenomenon. Hence, with shock fitting, it is possible to efficiently and accurately simulate difficult blunt-body flows and their

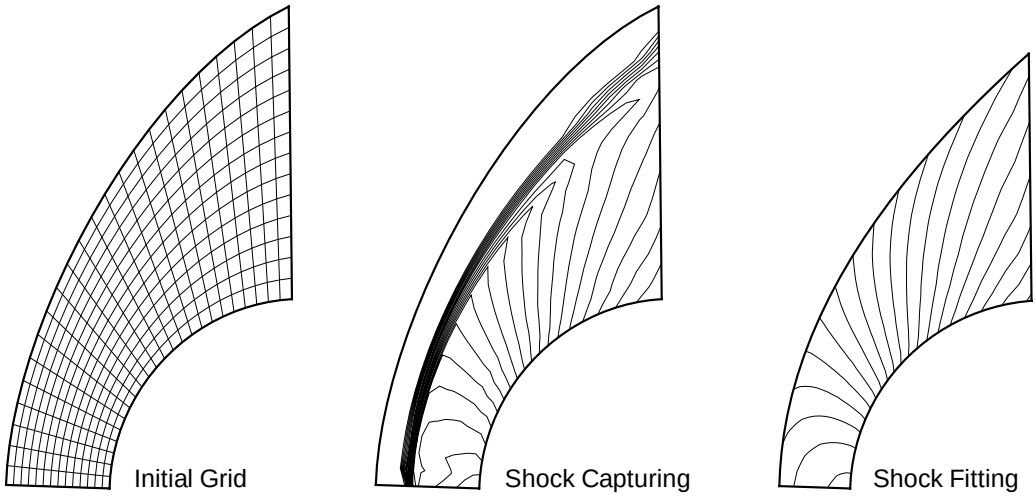



Figure 4.1: Comparison of shock capturing and shock fitting solutions on a relatively coarse grid. Isopressure contours are shown.

bow shocks in a robust manner, without instability or the need for excessively dissipative schemes or fine grids.

With its clear advantages, it is difficult to see why the shock fitting method enjoys limited popularity and few (though staunch) advocates.[141] Perhaps one reason is that shock fitting, although well-suited to external shocks on blunt bodies, is more difficult to apply to internal, embedded, and weak shocks. The relocation of internal cell boundaries to suit the shock geometry or the use of shock-cut cells can become complex, albeit these techniques have been applied with success.[139, 142] Two examples are floating shock fitting,[143, 144] and the shock aligned grid technique.[145] Shock fitting is not restricted to structured grids, and has recently been used to compute backward and forward facing wedge flows on adaptive unstructured grids.[146]

Fitting bow shocks is not new; Moretti has been applying the method to blunt-body flow computations in three-dimensions since 1966.[147] The fact that reasonable inviscid solutions were obtained using between two and seven points between shock and body is testament to the efficiency of the technique. Although further developed by Moretti and others, and coupled to the $\lambda$-scheme,[148–150] the method of characteristics,[151] and finite-difference[152] and finite-volume formulations,[18] the basic philosophy of shock fitting remains unchanged. In the remainder of this chapter, the shock fitting formulation used in SF3D is described. Although the fundamentals of this implementation borrow from existing principles, most of the presented work, including the shock speed upwinding and robust shock interpolation methods, is believed to be original.

## 4.1 Shock Fitting Formulation

For the blunt-body flows considered in this thesis, only external bow shocks are present and need to be fit. The SF3D code resolves shocks in a hybrid manner so that, should an embedded shock should be present behind the fitted bow shock, it can be resolved by shock capturing. Also, shock fitting can be switched on or off during simulations at will, so that a captured shock can be transformed into a fitted shock, and conversely.

In Section 4.1.1, we first describe how the shock speed is determined. Section 4.1.2 discusses the one-dimensional interpolation and limiting technique employed at the shock to gain increased shock speed accuracy, while Section 4.1.3 covers issues related to shock wave stability. Then, we detail the requirements of flux solvers for shock fitting compatibility, and examine the shock capturing to shock fitting transition process.

### 4.1.1 Treatment of the Shock Interface

There are several different approaches to determine the speed of the shock interface, though all use the Rankine-Hugoniot shock wave equations in some way. The Rankine-Hugoniot equations specify that mass, momentum and energy fluxes across a shock wave must be conserved, by relating upstream and downstream flow variables in a shock-stationary frame of reference. The equations may be expressed as

$$\mathbf{u}_\mathrm{L} - \mathbf{u}_\mathrm{L}\cdot\hat{\mathbf{n}}_\mathrm{S} = \mathbf{u}_\mathrm{R} - \mathbf{u}_\mathrm{R}\cdot\hat{\mathbf{n}}_\mathrm{S} \tag{4.1}$$

$$\rho_\mathrm{L}(\mathbf{u}_\mathrm{L}\cdot\hat{\mathbf{n}}_\mathrm{S} - w_\mathrm{S}) = \rho_\mathrm{R}(\mathbf{u}_\mathrm{R}\cdot\hat{\mathbf{n}}_\mathrm{S} - w_\mathrm{S}) \tag{4.2}$$

$$P_\mathrm{L} + \rho_\mathrm{L}(\mathbf{u}_\mathrm{L}\cdot\hat{\mathbf{n}}_\mathrm{S} - w_\mathrm{S})^2 = P_\mathrm{R} + \rho_\mathrm{R}(\mathbf{u}_\mathrm{R}\cdot\hat{\mathbf{n}}_\mathrm{S} - w_\mathrm{S})^2 \tag{4.3}$$

$$e_\mathrm{L} + \frac{P_\mathrm{L}}{\rho_\mathrm{L}} + \frac{|\mathbf{u}_\mathrm{L} - w_\mathrm{S}\hat{\mathbf{n}}_\mathrm{S}|^2}{2} = e_\mathrm{R} + \frac{P_\mathrm{R}}{\rho_\mathrm{R}} + \frac{|\mathbf{u}_\mathrm{R} - w_\mathrm{S}\hat{\mathbf{n}}_\mathrm{S}|^2}{2} \tag{4.4}$$

where we have additionally specified a tangential velocity constancy condition. In Equations 4.1–4.4, L and R represent the left and right sides of the shock respectively, $w_\mathrm{S}$ is the normal shock speed, and $\hat{\mathbf{n}}_\mathrm{S}$ is a unit vector normal to the shock. These equations are valid for shock waves in any dimension.

A method commonly employed to determine the shock Mach number, is to use a complete set of upstream flow variables on the low pressure side of the shock and a compatibility equation along the downstream characteristic on the high pressure side. With such information it is possible to solve Equations 4.1–4.4 for the downstream flow variables and shock speed.[139, 150] This method is well-suited to flow solvers based on the method of characteristics and the $\lambda$-scheme. Unfortunately, though, thermodynamic properties of the gas are directly introduced into the fitting technique.

For finite-volume schemes, a complete set of upstream and downstream flow variables is already known, and thus Equations 4.1–4.4 are overconstrained. Solving Equation 4.2 for shock speed yields

$$w_{S1} = \frac{\rho_L \mathbf{u}_L \cdot \hat{\mathbf{n}}_S - \rho_R \mathbf{u}_R \cdot \hat{\mathbf{n}}_S}{\rho_L - \rho_R}, \qquad \rho_L \neq \rho_R. \tag{4.5}$$

Likewise, manipulation of Equation 4.3 gives[18]

$$w_{S2} = \mathbf{u}_L \cdot \hat{\mathbf{n}}_S - \frac{\text{sign}(P_R - P_L)}{\rho_L} \sqrt{\left| \frac{P_R - P_L}{1/\rho_L - 1/\rho_R} \right|}, \qquad \rho_L \neq \rho_R. \tag{4.6}$$

The sign term has been introduced for compatibility with the entropy condition, and consequently the positive square root should always be selected.

If the left and right conditions exactly constitute those required for a shock wave, the computed speeds $w_{S1}$ and $w_{S2}$ will be equal. If, during shock formation, the left and right states do not exactly satisfy the complete Rankine-Hugoniot conditions, a blending of the two speeds can be used to obtain

$$w_S = \alpha\, w_{S1} + (1-\alpha) w_{S2}, \qquad 0 \leq \alpha \leq 1. \tag{4.7}$$

A blending parameter of $\alpha = 0.5$ is normally used. The introduction of a third wave speed into this equation, based on the conservation of energy condition, was found unnecessary in practice. If the left and right states possess significantly different pressures but have zero velocity, the second term in Equation 4.7 will initially perturb the shock and set it moving, whereupon both terms will contribute to the calculation of the shock speed in the next timestep. An example of this scenario is the starting process in the classic shock tube problem of Sod.[153] Similarly, the first term provides impetus when a velocity jump exists, but there is yet no pressure difference across the newly forming shock.

An advantage of this scheme is that it is conservative. A new downstream flow state is not calculated during shock fitting; only the shock speed is calculated, and the fluxes across the shock and computation of cell properties are handled by the finite-volume scheme in the usual manner. Additionally, the scheme is time-accurate when the left and right states correspond to a shock discontinuity. Otherwise, small errors in temporal accuracy may result for a short period. Numerical experiments in Section 5.2 show that these errors are negligible.

### 4.1.2 Interpolation and Limiting at the Shock

For bow shocks residing on an edge of the solution domain, shock fitting can be implemented as a special boundary condition. In this case, the low pressure side of the shock is

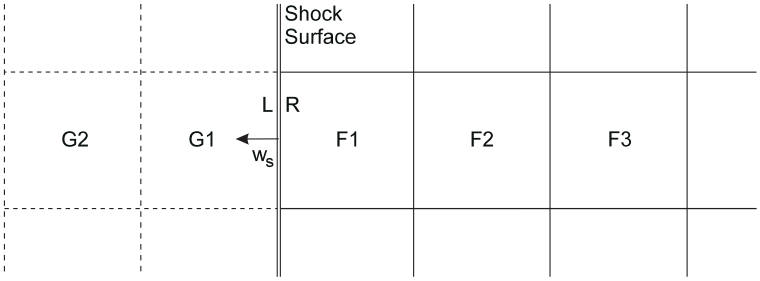



Figure 4.2: Nomenclature for shock fitting at a boundary.

contained within upstream ghost cells, and the high-pressure side is located downstream and lies inside the flow domain. Figure 4.2 shows the nomenclature used for shock fitting at a boundary in this thesis.

Attention is now directed at the calculation of left and right states for use with Equations 4.5 and 4.6 in the case of a bow shock. After setting both ghost cells to the constant inflow conditions, a simple approach is to designate

$$\mathbf{u}_{\mathrm{L}} = \mathbf{u}_{\mathrm{G1}}, \;\; P_{\mathrm{L}} = P_{\mathrm{G1}}, \;\; \rho_{\mathrm{L}} = \rho_{\mathrm{G1}}, \text{ and} \tag{4.8}$$

$$\mathbf{u}_{\mathrm{R}} = \mathbf{u}_{\mathrm{F1}}, \;\; P_{\mathrm{R}} = P_{\mathrm{F1}}, \;\; \rho_{\mathrm{R}} = \rho_{\mathrm{F1}}. \tag{4.9}$$

While convenient, this approach induces spatially first-order errors in a solution which may elsewhere be of higher order. Effectively, a half-cell width of flow gradient is ignored on the downstream side.

A one-sided reconstruction technique that boosts accuracy of the downstream shock state is now presented. First, the modified van Albada limiter and MUSCL reconstruction of Section 3.2.3 is applied to find the necessary flow variables on the right side of the interface. However, instead of interpolating across the shock, we use a purely one-sided reconstruction with an extended stencil. Thus the extreme and numerically troublesome gradients at the discontinuity are avoided. The reconstructed dimensional gradient at the downstream side of the shock is

$$\Delta_1 = \frac{s_{\mathrm{F2}}}{2}[(1 - s_{\mathrm{F2}}\kappa)\Delta^{\oplus}_{\mathrm{F2}} + (1 + s_{\mathrm{F2}}\kappa)\Delta^{\ominus}_{\mathrm{F2}}], \tag{4.10}$$

where the difference terms have their usual meaning. Explicitly, for pressure, they are defined as

$$\Delta^{\ominus}_{\mathrm{F2}} = 2(P_{\mathrm{F2}} - P_{\mathrm{F1}})/(h_{\mathrm{F1}} + h_{\mathrm{F2}}), \text{ and} \tag{4.11}$$

$$\Delta^{\oplus}_{\mathrm{F2}} = 2(P_{\mathrm{F3}} - P_{\mathrm{F2}})/(h_{\mathrm{F2}} + h_{\mathrm{F3}}), \tag{4.12}$$

and similarly for density and and shock-normal velocity. The limiting function $s$ and upwinding parameter $\kappa$ are as described in Section 3.2.3. In most cases, the gradient of

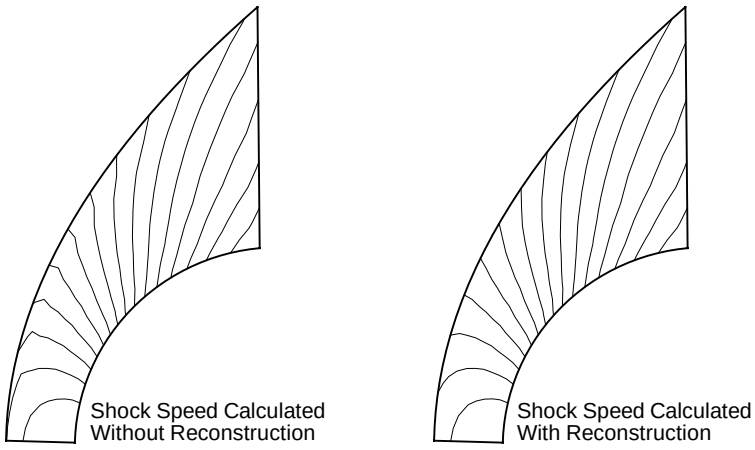



Figure 4.3: Comparison of shock fitting simulations performed using shock speeds calculated with and without reconstruction.

Equation 4.10 can be extrapolated to accurately determine the value of the corresponding flow variable at state R. However, if the states G1 and F1 do not even approximately satisfy the Rankine-Hugoniot equations (in other words, the supposed shock does not exist), difficulties arise: the one-sided nature of the extrapolation and downstream-biased gradient will quickly cause an instability at the shock fitting boundary. Although a condition imposed in Section 4.2 is designed to prevent this event from occurring, it would still be beneficial to make the reconstruction independently robust.

The proposed remedy is to introduce a further modified reconstruction, using the cross boundary gradient to detect the situation of a non-existent interface shock. The pressure gradient at the boundary is given by

$$\Delta_2 = \Delta_{\mathrm{F1}}^{\ominus} = 2(P_{\mathrm{F1}} - P_{\mathrm{G1}})/(h_{\mathrm{F1}} + h_{\mathrm{G1}}). \tag{4.13}$$

When a shock exists at the boundary, we would expect $|\Delta_2| \gg |\Delta_1|$. In this case, it is safe to extrapolate state R using $\Delta_1$. Alternatively, if $|\Delta_2| \leq |\Delta_1|$ there is probably no significant discontinuity at the boundary and it would be wise to discard any downwind-biased reconstruction. Instead, upwinding the right interface state would be a better option. To summarize, we may express the robust interface reconstruction scheme with accuracy for shocks as

$$P_{\mathrm{R}} = P_{\mathrm{F1}} - h_{\mathrm{F1}}[\mathrm{sign}(|\Delta_1| - |\Delta_2|)\,(\Delta_2 - \Delta_1) + \Delta_1 + \Delta_2], \tag{4.14}$$

for the pressure variable. Reconstructions for density and shock-normal velocity are similarly defined.

This formulation has the additional advantage of allowing abrupt cell clustering close to the shock, since the reconstruction scheme senses cell spacing. Post-shock chemical kinetics, for example, could be observed at high resolution with this technique.

Figure 4.3 shows simulations of cylinder flow using shock speeds computed with and without reconstruction. The flow conditions are the same as for Figure 4.1, but the grid is at a slightly different resolution (10 by 30 cells). Results for the unreconstructed scheme show that smaller pressures are predicted behind the shock, and an oscillation soon follows. No such oscillation is observed for the solution with shock reconstruction. Comparison of the two solutions reveals that differences in flow variables do not generally exceed 4% behind the shock. By the time the flow reaches the body, these differences become much smaller. The 4% accuracy gain is obtained at negligible computational cost, and alone justifies the use of Equation 4.14. However, this is not seen as the main benefit of the reconstruction scheme. The primary advantage of the reconstruction is to avoid introducing post shock noise which has the potential to trigger instability.

### 4.1.3 Shock Interface Stability

So far, a method for determining shock speed at boundary interfaces has been proposed and described. Because this speed corresponds to an averaged value over the interface, we assign the value to the interface centre and specify that it is directed in the interface normal direction. Shock speeds at interface centres **A**, **B**, **C** and **D** are diagrammatically represented by the arrows in Figure 4.4.

If the interfaces were allowed to translate with the velocity specified at their centres, the shock surface would quickly break into a series of disconnected or overlapping tiles. The shock surface may be kept contiguous, though, by selecting an equal and appropriate velocity for the coincident vertices from abutting interfaces. For interfaces **A**, **B**, **C** and **D**, the corresponding shared vertex is **V** in Figure 4.4. The vertex velocities should in conjunction cause the boundary interfaces to translate, rotate, and deform to maintain the correct shock displacement and alignment.

An obvious choice for the shock velocity at vertex **V** is the average of the shock velocities computed at the neighbouring interfaces:

$$\mathbf{w}_{\mathrm{S,V}} = (\mathbf{w}_{\mathrm{S,A}} + \mathbf{w}_{\mathrm{S,B}} + \mathbf{w}_{\mathrm{S,C}} + \mathbf{w}_{\mathrm{S,D}})/4. \tag{4.15}$$

Applied to blunt-body bow shocks, Equation 4.15 yields marginally stable shocks with corrugations visible on the shock surface. Figure 4.5 shows the effect of velocity averaging on an oblique part of the bow shock formed on a cylinder in hypersonic flow. The test simulation has conditions corresponding to those in Figure 4.1, and in this case the complete grid size is 30 by 50 cells. The solution shown is that computed after about 6000 timesteps, or roughly 60 body lengths of flow. While this should be ample time to reach convergence, the speeds of some of the boundary vertices remain at values far from

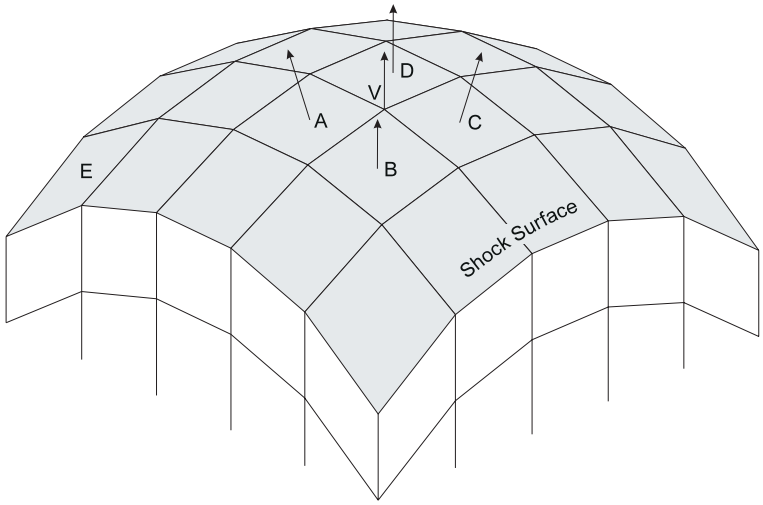



Figure 4.4: Diagram showing shock speeds on a boundary fitted shock surface.

zero. Over time, oscillations in the distribution of shock speeds generate self sustaining corrugations on the shock surface. The corrugations are more likely to occur and have greater amplitude for oblique shocks approaching the Mach angle.

The averaging process of Equation 4.15 does not discriminate between wave speeds obtained from interfaces upstream or downstream of the vertex. It is suggested that the contribution of downstream speeds causes an unphysical upstream propagation of information at each timestep. Thus, for the example simulation being considered, it is possible that information travels from the shock at the outflow boundary to the stagnation stream line within 50 steps. To correct the situation, weighting terms are introduced to penalize wavespeeds gleaned from interfaces downstream of a vertex.

Let $\hat{\mathbf{t}}_{\mathbf{AV}}$ be a unit vector tangent to the shock surface, directed from interface centre **A** to vertex **V**. The post-shock Mach number of the flow at **A**, in the direction of the tangent vector, is thus

$$M_{\mathbf{A}} = \frac{\mathbf{u}_{\mathbf{A}} \cdot \hat{\mathbf{t}}_{\mathbf{AV}}}{a_{\mathbf{A}}}, \tag{4.16}$$

and is similarly defined for the other boundary interface centres. We specify weightings for shock velocity contributions from each interface using a non-negative function based on the associated post-shock Mach number, such that

$$\omega_{\mathbf{A}} = \omega(M_{\mathbf{A}}). \tag{4.17}$$

Once the weightings have been determined, the shock velocity at the vertex is found using the expression

$$\mathbf{w}_{\mathrm{S,V}} = \frac{\omega_{\mathbf{A}}\mathbf{w}_{\mathrm{S,A}} + \omega_{\mathbf{B}}\mathbf{w}_{\mathrm{S,B}} + \omega_{\mathbf{C}}\mathbf{w}_{\mathrm{S,C}} + \omega_{\mathbf{D}}\mathbf{w}_{\mathrm{S,D}}}{\omega_{\mathbf{A}} + \omega_{\mathbf{B}} + \omega_{\mathbf{C}} + \omega_{\mathbf{D}}}, \tag{4.18}$$

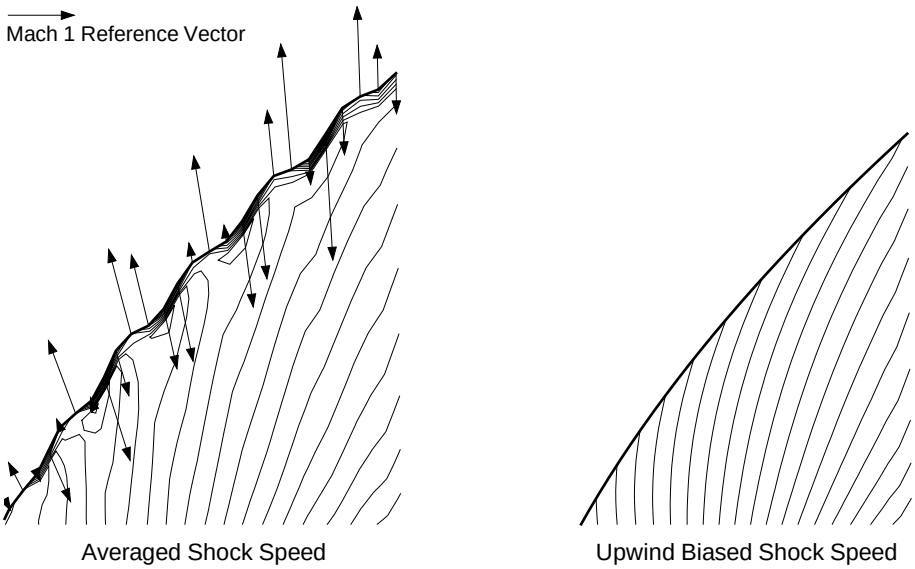



Figure 4.5: The effect of upwinding shock speed at the boundary interface, for an oblique shock. Isopressure contours and shock velocity (not flow velocity) vectors are shown.

providing that not all $\omega$ are zero. In such a case, though rarely encountered, we revert to Equation 4.15. This is equivalent to setting all $\omega$ to unity.

Two different weighting functions are included in SF3D. The first is fully upwind, and very robust:

$$\omega(M) = \frac{M + |M|}{2}. \tag{4.19}$$

Recognizing that wave speed contributions from subsonic downstream flow should also be included for accuracy, it seems reasonable to introduce a smooth weighting function like

$$\omega(M) = \begin{cases} [(M+1)^2 + (M+1)|M+1|]/8 & \text{if } M \leq 1, \\ M & \text{otherwise.} \end{cases} \tag{4.20}$$

Plots of the fully upwind and upwind-biased weighting functions are displayed in Figure 4.6. While somewhat *ad hoc*, both functions work well in numerical experiments; Figure 4.5 shows that upwind biasing produces a smooth bow shock wave for the cylinder flow case considered earlier in this section.

If the vertex velocity is set equal to the vertex shock speed,

$$\mathbf{w}_\mathbf{V} = \mathbf{w}_{\mathrm{S},\mathbf{V}}, \tag{4.21}$$

difficulties with grid management may occur. To explain, consider the general case where cell interfaces at the shock are not orthogonal to the radiating grid lines between shock

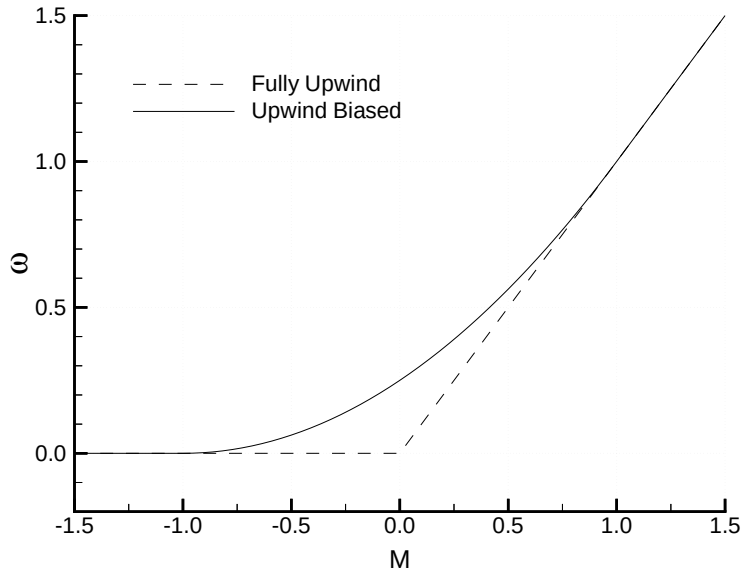



Figure 4.6: Weighting functions for the calculation of shock speed at vertices.

and body. An example where this is clearly the case is the cell marked **E** in Figure 4.4. Since the shock velocities at the surrounding vertices are roughly aligned in the interface normal direction, the shock surface would locally begin to float away. Unchecked, the code may end up simulating the shock in a region of flow that was not initially desired for inclusion in the solution domain. To avoid relocating the shock, we constrain boundary interface elements to move along the radial grid lines, as if they were rails. Thus, rather than Equation 4.21, the following is used to assign the vertex velocities at the shock:

$$\mathbf{w_V} = (\mathbf{w}_{S,\mathbf{V}} \cdot \hat{\mathbf{t}}_{\mathbf{RV}})\, \hat{\mathbf{t}}_{\mathbf{RV}}. \tag{4.22}$$

The unit vector $\hat{\mathbf{t}}_{\mathbf{RV}}$ is defined by alignment with the radial grid line terminating at **V**. After the calculation of velocities at all cell vertices contained on the shock surface, the velocities of the internal mesh vertices are determined. A linear velocity distribution between shock and body is used, based on distance from the body. In this manner, the relative spacing between cells is preserved during shock movement.

Finally, the velocities of all cell interfaces are found by averaging the newly assigned velocities of their vertices. Note that these particular calculations are based purely on geometry considerations; they do not affect flow stability, and clearly should not be upwind biased. Also, the GCL interface velocities may now be calculated, as described in Section 2.7.

The stability measures introduced in this section implicitly impart dissipation on the shock movement, and therefore have the potential to affect time accuracy to a degree. It

could be argued that we have used the guise of shock fitting to introduce numerical dissipation, contradicting our goal of developing a stable low-dissipation scheme. This is not the case, however, for two fundamental reasons: (i) All numerical dissipation associated with the shock fitting vanishes when convergence is reached, and (ii) The dissipation inherently linked to the shock movement does not affect the accuracy of individual flux calculations.

### 4.1.4 Shock Fitting and Flux Solvers

The boundary shock fitting technique can be used in conjunction with virtually any flux solver, providing care is taken at the fitted shock surface. In the case of a blunt body with supersonic inflow, the free-stream characteristics dictate that the correct inviscid fluxes across the shock are those calculated from the inflow conditions alone. Thus, it is a simple matter to set the boundary fluxes explicitly at the shock. Alternatively, any solver which gives the correct fluxes for a shock captured between two cells could also be used to calculate the boundary fluxes. Both approximate and exact Riemann solvers are suitable. The AUSM and most of its derivatives, though, encounter a small amount of dissipation in this situation and slightly undervalue the fluxes. During shock capturing the dissipation is not ordinarily a problem, since the undervalued fluxes create an upstream perturbation that is recovered as it convects back into the shock in the next timestep. Because there are no cells upstream of the shock in boundary shock fitting, this information is immediately lost. Hence if schemes such as AUSM are to be used in a shock fitting simulation, fluxes at the shock boundary should be calculated directly from the inflow state. For the same reason, viscous stresses and heat transfer across the shock are not included while shock fitting, regardless of the inviscid flux solver used. These considerations ensure that the CFD code remains conservative and maintains accuracy.

## 4.2 Hybrid Shock Capturing and Fitting

The transition between shock capturing and shock fitting is now described, using an algorithm that requires little intervention by the code operator. This is the usual way in which fitting is applied to blunt-body flows in this thesis.

Initially, the boundary that is to eventually to comprise the shock is set to track the shock wave, by moving towards it at either the left or right flow wave speed. Alternatively, the boundary can be set to remain stationary at an initial position, where it will wait until a captured shock reaches and coalesces with it. In either case, once a shock is detected at a boundary interface the shock fitting process is locally enabled. With reference to

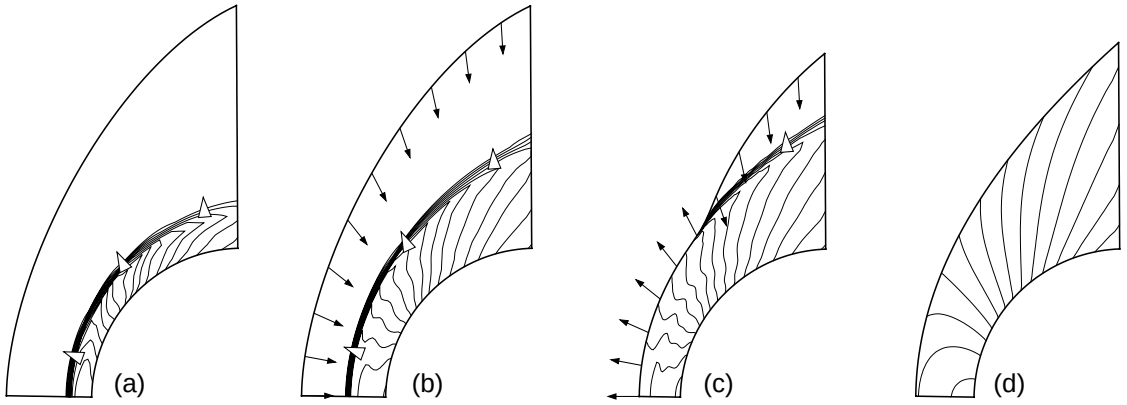



Figure 4.7: Timelapse images showing the transition from shock capturing to shock fitting on an impulsively started blunt-body problem. Isopressure contours and boundary velocity vectors are shown.

Figure 4.2, a simple shock detector is

$$\frac{|\rho_{F1} - \rho_{G1}|}{\max(\rho_{F1}, \rho_{G1})} \geq K, \tag{4.23}$$

where the value of the threshold parameter $K$ dictates sensitivity to density discontinuities. A value of $K = 0.2$ works well in most circumstances. More elaborate shock detectors are available in the literature,[144] however Equation 4.23 suffices for blunt-body bow shocks. Should the detector falsely trigger, the shock-fitting process is robust enough to remain stable and allow the disturbance to pass. Once the disturbance no longer exists at the boundary, the detector will locally turn off the fitting.

The capturing to fitting transition process is now illustrated, again using the example of a cylinder in a supersonic stream. Figure 4.7(a) shows a captured shock emanating from a cylinder towards the outer domain boundary. The flow was impulsively started, and these results correspond to a time of approximately one body length of inflow after the simulation began. Hollow arrowheads indicate the direction in which the captured shock is moving. At a specified time, when the captured shock is well developed, shock tracking is initiated and the outer boundary moves towards and searches for the shock wave. Shock tracking is pictured in Figure 4.7(b), and shows the velocity vectors of boundary vertices. The lower part of the boundary is the first to reach the shock, and shock fitting is locally enabled. As shown in Figure 4.7(c), the fitted shock then continues to move away from the body at the correct shock speed. The upper part of the boundary, though, remains in shock-tracking mode. For the shock-fitted region of Figure 4.7(c), noise is visible from the body to a line roughly midway between body and shock. This noise is contained within flow that was originally processed by the captured shock. Eventually, this noisy flow is discharged through the outflow boundary, being displaced by gas cleanly processed by the

fitted shock. In the meantime, the interface between clean and noisy flow should not be mistaken for a physical flow feature. After the entire shock is fitted, and full convergence is reached, the result in Figure 4.7(d) is finally obtained.

## 4.3 The Apollo Heat Shield Revisited

Armed with a new shock-fitting technique, attention is returned to the carbuncle instability observed on the Apollo heat shield in Sections 3.5 and 3.6.

Flow around the Apollo geometry was resimulated using shock fitting and the AUSMDV flux solver, with conditions, grid size, and spatial and temporal accuracy equivalent to the shock capturing simulation. As previously described, shock fitting compresses the grid, causing effectively unused free-stream cells to become relocated in the shock layer. Therefore, although the captured and fitted Apollo simulations were performed on grids with equal cell number, the shock-fitting results are of higher resolution. In Section 3.6, it was noted that higher grid resolution increases the probability of the carbuncle instability occurring.

Figure 4.8(a) shows a converged shock-fitting result. And, although at higher resolution than the shock-capturing simulation, no carbuncle is observed. Thus, the postulate that the carbuncle is related to numerical shock structure is further evidenced. Another shock-fitting simulation was performed at lower resolution, but with a solution domain covering the expansion region around the shoulder of the forebody. Results produced on this second grid are shown in Figures 4.8(b) and 4.8(c). The computed interferogram compares reasonably well with the experimental interferogram of Figure 3.7; fringe shape and fringe number are quite similar. The computed shock standoff distance, however, is smaller than that measured in experiment. The reduced standoff is probably due to the assumption of chemical equilibrium, since a frozen gas simulation produced a standoff distance greater than that measured in experiment. The relationship between standoff distance and gas chemistry is investigated more thoroughly in Chapter 5.

## 4.4 Implementation Issues

This chapter concludes discussion of the flow models and numerical algorithms implemented in SF3D. The remainder of the thesis is chiefly concerned with the results and application and analysis of results generated by the code. Details concerning the practical operation of SF3D, including information for new users and researchers wishing to modify the program, are presented in Appendix C.

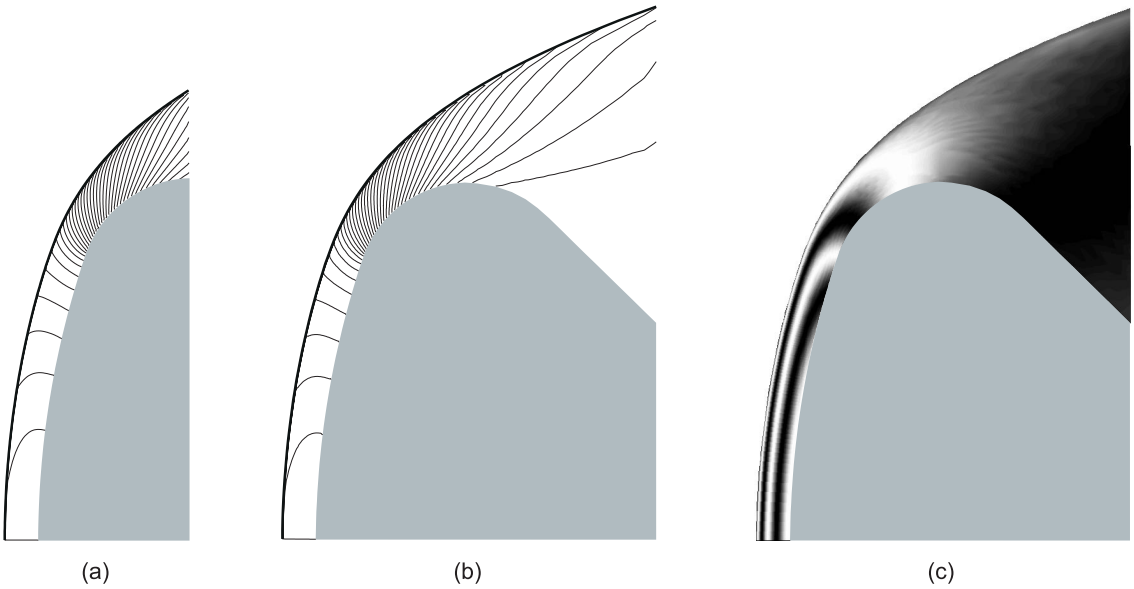



Figure 4.8: Isopressure contours and computed interferogram for shock-fitting simulations of the Apollo heat shield.

CHAPTER 5

# Test Cases

Determining the credibility of computational fluid dynamics simulations and codes is, inherently, subjective. Some authors go as far as stating that it is impossible to universally verify and validate numerical models of natural systems.[154] While semantics may prove this to be the case, we take the approach of Roache[155] and use the concepts of accuracy tolerancing, range of applicability, and physical intuition as evidence of credible simulations.

The range of applicability of SF3D has already been bounded by the limitations of the physical models and numerical techniques described in the preceding chapters. However, we have not yet verified and validated the complete CFD code for operation inside these bounds. Verification is the process of establishing the accuracy and reliability of the numerical technique used to solve the mathematical model of the physical system.[156] In line with this definition, we verify SF3D in Sections 5.1–5.8 without the use of experimental data.[155] In contrast, validation involves determining the accuracy of the mathematical model as it applies to the physical system.[157] SF3D is validated in Sections 5.9–5.11 and Chapter 6. The type and thoroughness of verification and validation needs to be commensurate with the purpose for which the code is to be used. When simulation results are to be used as the basis for the design of hardware (as envisaged with air data systems in Chapter 7) or in a safety critical application, it is especially important to be able to quantify simulation error precisely and accurately.[158]

The credibility of complex simulations is generally difficult to assess and thus rarely established.[159] A common practice is to instead examine code performance on well-known or simple test cases, with each case designed to test a specific model, technique, or likely failing. It is important, though, that verification test cases remain relevant to the intended

code purpose. For this reason blunt-body flows are primarily used to validate SF3D in this chapter, and a verification grid convergence study is performed under HYFLEX simulation conditions in the next chapter. In this chapter, the test cases used for verification are not always closely linked to blunt-body flows, but are instead designed to interrogate the core Navier-Stokes solver, and allow evaluation of errors due to dissipation, noise, and discretization.

Validating and verifying a code does not guarantee the absence of programming errors. It is expected, however, that critical mistakes pertaining to blunt-body simulations would become apparent in the test cases presented. In any case, the existence of programming errors within CFD software does not necessarily justify immediate invalidation; programming errors of some type probably reside within most CFD codes of size.[141]

## 5.1 Geometric Conservation Law Test

The formulation presented in Section 2.7 was introduced to ensure that the CFD code observes the geometric conservation law, which is a necessary condition for numerically maintaining the integrity of the physical conservation laws. We now verify that the implementation of that formulation results in a GCL compliant scheme.

The initial setup of the test case is shown in Figure 5.1. The solution domain is a cube with nondimensional side lengths of unity, and 10 cells in each index direction. The ratio of specific heats of the ideal, polytropic test gas is constant and 1.4, and the initial internal energy and density of the gas are set to nondimensional values of unity. To check that mass fractions in heterogeneous mixtures are conserved, we use a gas composed of two species. The mass fraction of the first species is set to 0.1, and each species is assumed to possess the same thermodynamic properties. The test gas is initially at rest throughout the solution domain.

At the start of the simulation, the speed and direction of all internal cell vertices are randomly assigned, such that $|\mathbf{w}| \leq a/50$. Vertices on the domain boundaries are fixed in space, and each of the six boundaries is an adiabatic, frictionless wall. Maintaining the cell vertex velocities, the solution is marched forward in time through 100 iterations, at CFL 0.5. Nominally second-order Runge-Kutta time integration, and second-order MUSCL reconstruction with the modified van Albada limiter, are used. Fluxes are calculated with the AUSMDV.

The final, distorted, grid geometry is shown in the cutaway view at the right of Figure 5.1. Because the boundary conditions are unchanged, and the test gas was initially at rest with constant conditions throughout, we should physically expect the final solution to be identical to the initial conditions. A code not satisfying the GCL, however, would

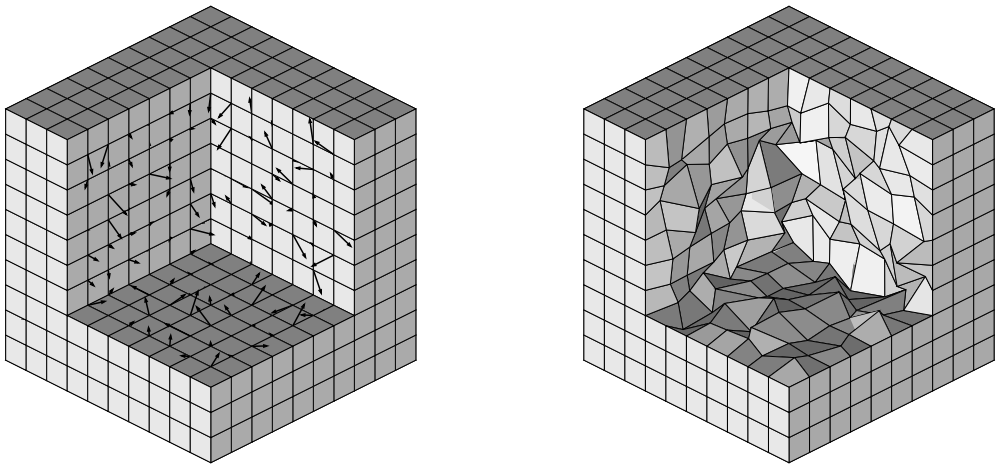


Figure 5.1: Cutaway views showing the initial grid and vertex velocity vectors used for the GCL test (left), and the grid after 100 timesteps (right).

incorrectly introduce perturbations into the solution through the movement of internal control volume interfaces. Such a code would also be expected to lose or gain total mass (or any other physically conserved quantity) during interface movement.

The final solution showed mass leakage of less than $1 \times 10^{-14}$, which is acceptable considering that machine precision is approximately $1 \times 10^{-16}$ and 100 steps were taken. Perturbations in the individual velocity components at cell centres were less than $1 \times 10^{-15}$. No variation in density, internal energy, species mass fractions, pressure and temperature was observed at all. After switching off the GCL routine and rerunning the simulation, perturbations in density and energy of up to 5% were produced, and the gas had attained speeds in the order of $1 \times 10^{-4}$. Although the variation in state properties for the non-GCL simulation is unacceptably large, note that this test case is contrived to exacerbate such problems. When shock fitting and a moving grid is used in practical blunt-body simulations, the adverse effects resulting from GCL violation are smaller and less noticeable, especially when convergence is approached. However, for unsteady problems where time accuracy is vital, the results of this test show that the GCL formulation is a necessity.

## 5.2 Sod's Shock Tube Problem

The code's ability to resolve shocks, contact discontinuities, and expansions is now assessed using a one-dimensional shock tube problem. Additionally, both shock capturing and shock fitting techniques will be tested, so that their comparative performance can be examined.

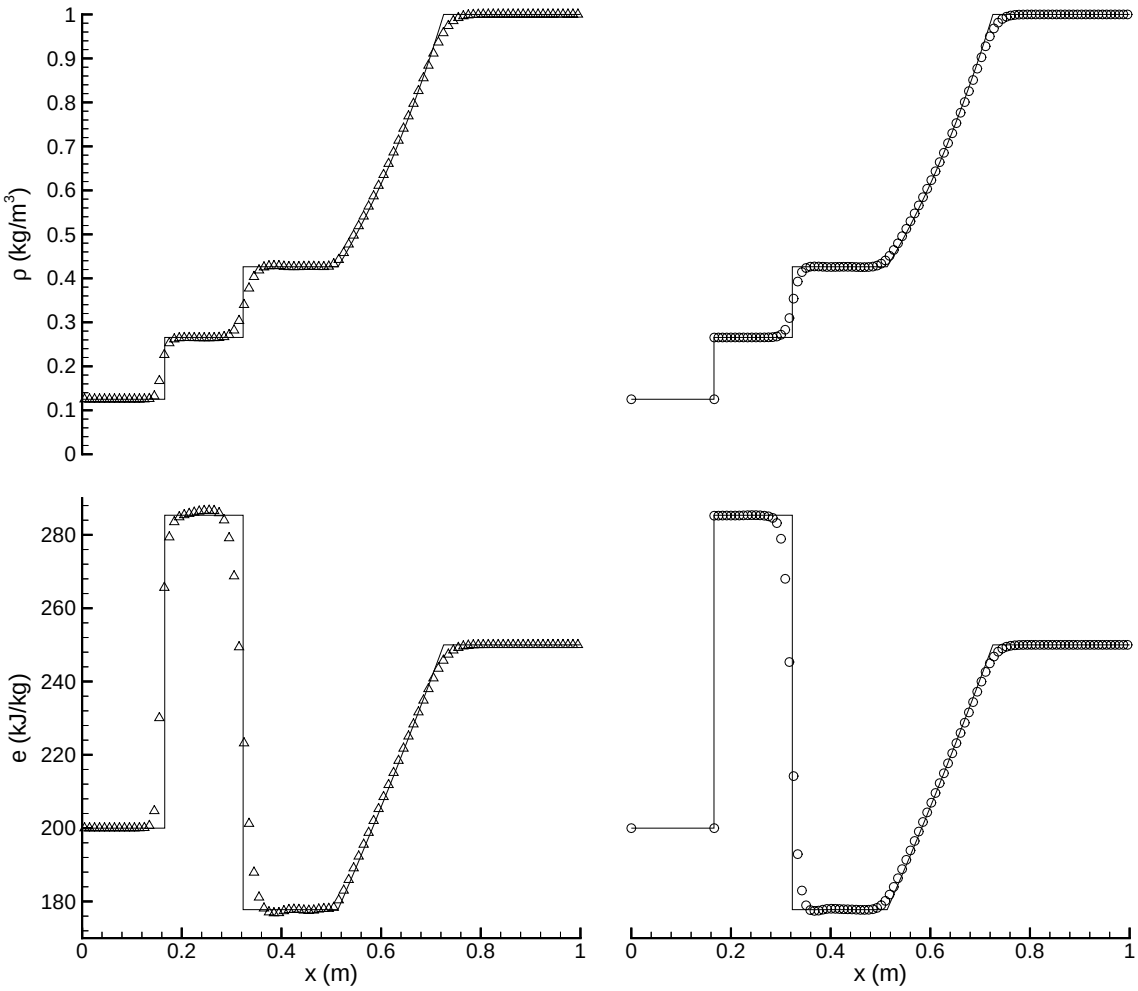



Figure 5.2: Simulation results for Sod's shock tube problem, after approximately $6.0 \times 10^{-4}$ s. —— Exact solution, △ shock capturing, ○ shock fitting.

The conditions and mesh resolution originally suggested by Sod[153] are used in this test case. For the shock-capturing simulation, the shock tube is a one-dimensional grid with 100 cells spaced evenly along its length from $x = 0$ to $x = 1$ m. Initially we suppose that an imaginary diaphragm exists at $x = 0.5$ m, where the conditions to the left of the diaphragm are

$$\mathbf{u} = \mathbf{0},\ \rho = 0.125\ \text{kg/m}^3,\ e = 2.0 \times 10^5\ \text{J/kg K}, \quad x \leq 0.5\ \text{m}.$$

At the right of the diaphragm, the tube is initially filled with gas at higher pressure and temperature,

$$\mathbf{u} = \mathbf{0},\ \rho = 1.0\ \text{kg/m}^3,\ e = 2.5 \times 10^5\ \text{J/kg K}, \quad x > 0.5\ \text{m}.$$

Calorifically perfect, ideal air is chosen as the test gas. At the commencement of the simulation, the imaginary diaphragm is removed, and the ensuing interaction between the high pressure and low pressure gas is observed. Shock-capturing results after approximately $6 \times 10^{-4}$ s are shown in Figure 5.2. The AUSMDV flux solver, second-order temporal and spatial accuracy, and a CFL number of 0.5 were used.

Conditions for the shock-fitting simulation are identical to the shock-capturing simulation, however a different grid is employed. Although both grids contain 100 cells, the shock-fitting grid is initially half as long, running from $x = 0.5$ to $x = 1$ m. The low pressure gas state is implemented as an upstream inflow boundary condition, rather than as an initial condition. As the simulation progresses, the grid expands in synchronization with the shock wave motion. Two computed shock-fitting results are shown at the right of Figure 5.2.

As should be expected, the shock-fitting solution exhibits a perfect discontinuity at the shock, while the captured shock wave is slightly diffuse and required about four cells to adequately define the shock wave. The plateau of internal energy, contained between shock and contact surface, is mostly flat in the shock-fitting solution and agrees reasonably well with the exact solution. In contrast, the plateau has been diffused and excessively rounded during shock capturing. In both the density and internal energy plots, both techniques are observed require roughly six cells to capture the contact discontinuity. However, the fitting solution has a smaller cell spacing since it does not waste cells upstream of the shock, and the discontinuity is thus condensed in space and appears slightly more crisp. For the same reason, the expansion fan is more accurately resolved in the case of shock-fitting.

At $6 \times 10^{-4}$ s after the start of the simulation, the distance traversed by the fitted shock wave is slightly in error, being too small by 1%. This error is attributable to a delay in the starting processes occurring directly after diaphragm removal, while the shock, contact

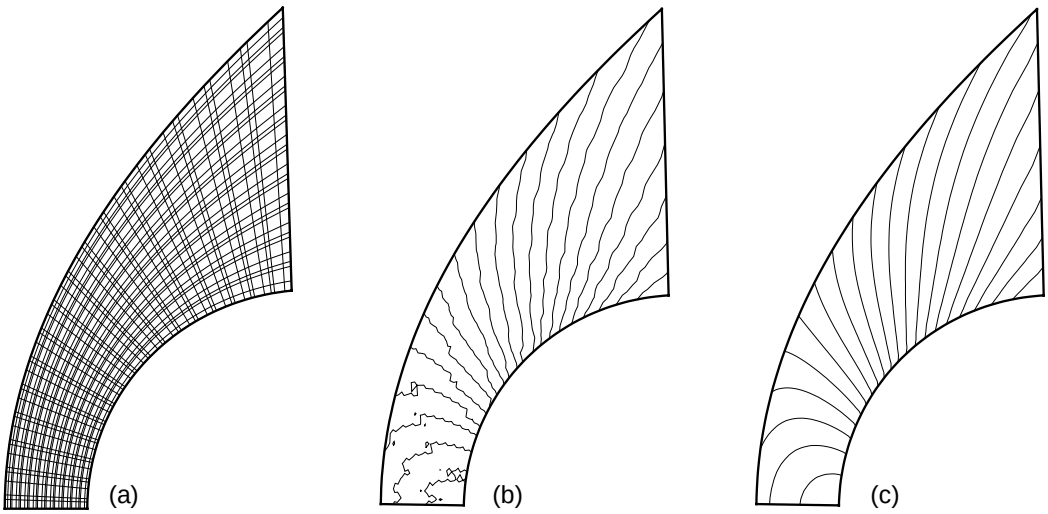



Figure 5.3: Results showing the operation of different reconstruction schemes on a nonuniformly spaced grid.

surface, and expansion fan structures are forming. Section 4.1.1 discusses the source of the problem in more detail. In terms of time accuracy, the 1% error is considered to be slight, especially after noting that the width of the captured shock corresponds to a 15% fraction of the traversed distance.

## 5.3 Time Convergence on a Nonuniform Grid

In Section 3.2.3, we discussed why severe cell clustering is sometimes required to make optimum use of cells in three-dimensional blunt body simulations. In turn, this motivated the development of a one-dimensional reconstruction scheme that accounts for nonuniform cell spacing. The spatially sensitive scheme is now tested against a spatially insensitive reconstruction scheme, through the simulation of blunt-body flow on a nonuniform grid.

Again, calorifically perfect, ideal air is the test gas, and a cylinder is used for the blunt body. A temperature of 300 K and density of 0.01 kg/m$^3$ define the free-stream gas state, and the inflow speed is 3 km/s. The simulations are performed on a 30 by 50 cell grid with periodic clustering, as shown in Figure 5.3(a). Second-order Runge-Kutta time marching, the AUSMDV flux solver, and hybrid shock capturing and fitting are used.

The reconstruction scheme described in Section 3.2.1, combining MUSCL interpolation and the min-mod limiter, were used to produce the results in Figure 5.3(b). The pressure contours display significant noise and oscillation, presumably due to the spatial

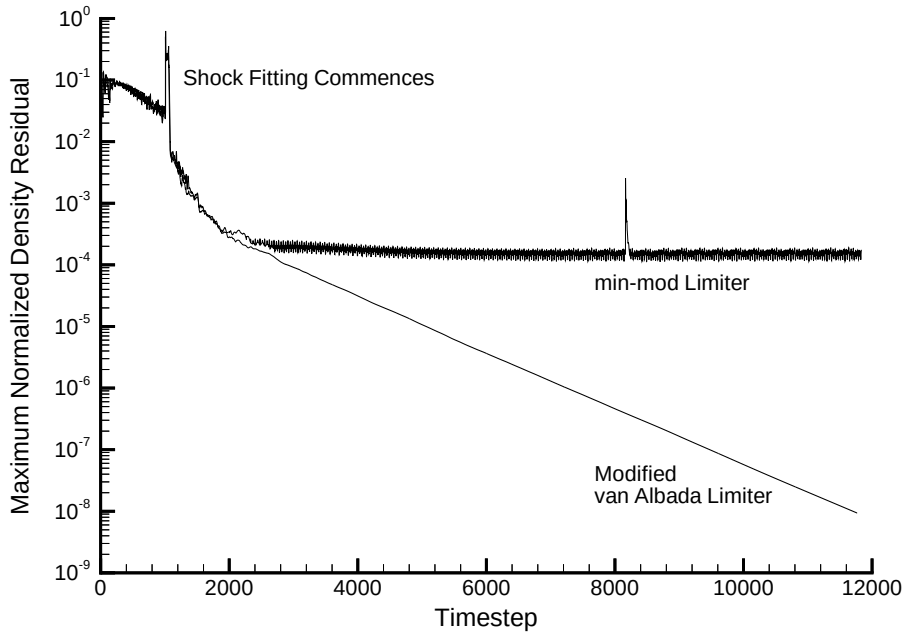



Figure 5.4: Convergence history for the nonuniform grid problem of Figure 5.3.

insensitivity of the reconstruction scheme. By using the spatially sensitive reconstruction described in Section 3.2.3, combining MUSCL interpolation with a modified van Albada limiter, the results in Figure 5.3(c) are achieved. The noise has vanished, and the detrimental effects of the poor quality discretization have been alleviated. For comparison, an attempt was made to also perform a spatially first order simulation on the nonuniform mesh. A first-order solution could not be made to converge on the grid of Figure 5.3(a), however.

A convergence time history for the two spatially second-order simulations is shown in Figure 5.4. The largest normalized density residual encountered within the solution domain is plotted against timestep. For both reconstruction schemes, a spike in the residual is observed during the transition from shock capturing to shock fitting. After shock fitting begins, both the density residuals start to decrease in a like manner, until about 2000 timesteps and a residual of $10^{-4}$ are reached. At this point, the reconstruction using the min-mod limiter reaches and oscillates about a residual limit, and does not reach convergence. We attribute this behaviour to the nondifferentiable nature of the min-mod limiter. In contrast, the reconstruction scheme incorporating the differentiable van Albada limiter continues to converge at a constant rate. For practical simulations, convergence criteria vary and depend on the problem and required solution accuracy. In many cases, though, an acceptable convergence indicator is when residuals of less than $10^{-6}$ are achieved. Hence, the use of a differentiable limiter is certainly advantageous.

## 5.4 Grid Convergence for a Supersonic Vortex

Nominally, the reconstruction schemes presented in Chapter 3 are of second-order spatial accuracy in one-dimension. However, in the extension to multiple dimensions the order of accuracy is not necessarily retained. A grid convergence test is now performed to determine the convergence rate of the schemes in two-dimensional flow.

The inviscid supersonic vortex test proposed by Aftosmis *et al.*[160] is a convenient case on which to evaluate the order of spatial convergence. Because there exists an exact, analytical solution to the flow problem, it is easy to determine the level of simulation error. Additionally, the absence of shock waves allows us to observe convergence for a smooth flow field, somewhat analogous to the relatively smooth flow that exists behind a fitted bow shock. It is stressed, however, that order of convergence is dependent on the specific grid geometry and flow case, and thus a separate grid convergence study is still performed for flow around the HYFLEX blunt body in Chapter 6.

The supersonic vortex is established using a duct bounded by two, circular, ninety-degree arcs, as shown in Figure 5.5(a). For the simulations presented here, we set the outer arc radius $r_o$ equal to 1.384 times the inner radius $r_i$, similarly to Reference 160. Conditions at the inner radius of the inflow plane, in nondimensional form, are

$$\mathbf{u}_i = 2.25\hat{\mathbf{i}},\ M_i = 2.25,\ \rho_i = 1,\ P_i = 1/\gamma,\ \gamma = 1.4.$$

The inflow density distribution is a function of radius, and for a calorifically perfect gas given by the expression[160]

$$\rho(r) = \rho_i \left[ 1 + \frac{\gamma - 1}{2} M_i^2 \left\{ 1 - \left( \frac{r_i}{r} \right)^2 \right\} \right]^{\frac{1}{\gamma - 1}}. \tag{5.1}$$

Flow speed is distributed inversely proportional to radius. For an isentropic vortex, we additionally have the relation $P = P_i \rho^\gamma$.

Ideally, the distribution of conditions across the inflow plane should be preserved, though rotated, at all downstream radial grid lines. Deviation of the simulation results from the correct solution can be measured in a number of ways. We use the $L_1$ norm to indicate average error, and the $L_2$ norm to give root mean square error. For density, the norms may be written

$$L_1 = \frac{1}{n} \sum \left| \frac{\rho - \rho_e}{\rho_e} \right|, \qquad \text{and } L_2 = \sqrt{\frac{1}{n} \sum \left( \frac{\rho - \rho_e}{\rho_e} \right)^2}, \tag{5.2}$$

where $n$ is the number of cells in the grid, and $\rho_e$ represents the exact solution to the problem.

Table 5.1: Grid convergence order $p$ evaluated with the $L_1$ and $L_2$ norms.

| Flux Solver | Norm | Unreconstructed | MUSCL Interpolation | |
|---|---|---|---|---|
| | | Unlimited | min-mod Limiter | van Albada Limiter |
| AUSM | $L_1$ | 1.11 | 1.21 | 1.28 |
| | $L_2$ | 1.10 | 1.19 | 1.25 |
| AUSMDV | $L_1$ | 1.10 | 1.43 | 1.41 |
| | $L_2$ | 1.12 | 1.43 | 1.43 |
| EFM | $L_1$ | 1.10 | 1.63 | 1.75 |
| | $L_2$ | 1.11 | 1.60 | 1.72 |
| Riemann | $L_1$ | 1.05 | 1.47 | 1.50 |
| | $L_2$ | 1.05 | 1.51 | 1.49 |

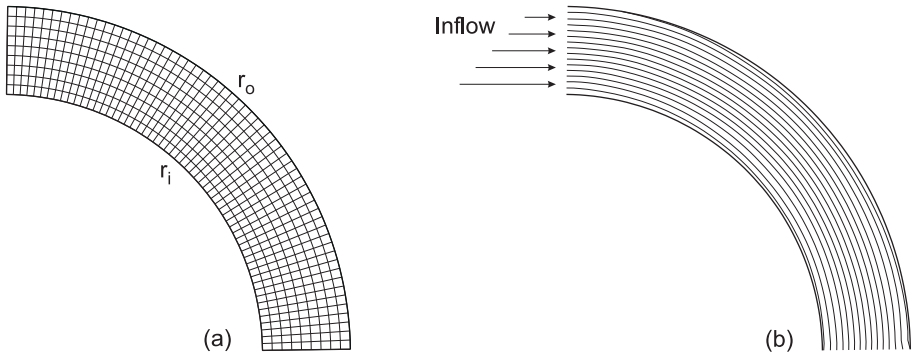



Figure 5.5: A sample grid used for the supersonic vortex problem, and isopressure contours of a solution generated using AUSMDV with the modified van Albada limiter and MUSCL interpolation.

Simulations performed on extremely high resolution grids should give close to exact results. By monitoring the error norms on simulations of different resolutions, we can evaluate how quickly the numerical scheme is converging to the exact solution. Using the $L_1$ norm, the order of grid convergence $p$ can be defined as[161]

$$p = \frac{\ln[(L_{1,b} - L_{1,a})/L_{1,a} + 1]}{\ln(h_b/h_a)}, \tag{5.3}$$

where $a$ and $b$ denote simulations performed at different grid resolutions, and $h$ is a characteristic cell width.

Table 5.1 presents the results obtained from a series of grid convergence tests using the supersonic vortex. Grid resolutions of 5 by 30, 10 by 60, and 20 by 120 cells were used, and each reported value of $p$ represents an average for all grids. Convergence orders

calculated using the $L_1$ and the $L_2$ norms agree well in each case.

As would be expected, the simulations run without reconstruction displayed close to first order convergence for all flux solvers. Reconstructions using the van Albada limiter exhibited convergence better, or at least roughly equal, to that achieved with the min-mod limiter. Note that there is effectively no difference between the van Albada and modified van Albada limiters in this problem, due to the equispaced grids. Overall, orders of convergence were significantly less than two. Particularly slow to converge was the AUSM scheme, especially when using the min-mod limiter. The fastest converging scheme was the combination of EFM with the modified van Albada limiter. Note, however, that a faster convergence rate does not necessarily imply better accuracy, as will be seen in some of the examples later in this chapter.

## 5.5 A Rearward Facing Step

The flow past a rearward facing step is now simulated, to examine the behaviour of flux solvers in resolving an expansion fan caused by the diffraction of a shock wave. This test case is relevant to blunt-body flows, since rapid expansions can occur at points where the body curvature suddenly changes, or at a shoulder.

A Mach 10 shock, propagating through calorifically perfect, inviscid air at atmospheric conditions, is diffracted around a ninety degree corner. A 1 m square, two-dimensional solution domain is used, with 100 cells in each direction. Isopressure contours in Figure 5.6 show solutions generated using different flux solvers, at a time 250 $\mu$s after the initially planar shock passes the corner. No reconstruction has been used, so that the operation of the flux solvers can be assessed in isolation.

The AUSM solution, shown in Figure 5.6(a), exhibits a strong shock wave developing curvature and weakening as it diffracts. The captured shock is quite broad, on account of the first order spatial accuracy. Also visible is a reasonably well defined contact surface, separating gas that was processed by the normal and oblique parts of the shock wave.[162] Extending horizontally from the corner of the step is a sudden discontinuity in pressure, appearing partway through the newly formed expansion fan. The stationary discontinuity processes upstream air, decreasing its pressure, and is thus termed an expansion shock.[94] Although expansion shocks satisfy conservation laws for mass, momentum, and energy, they are are entropy decreasing (violating the second law of thermodynamics) and do not physically occur in gases. Thus the flux solver has chosen a mathematically possible, but physically incorrect solution to the governing equations.

The results in Figure 5.6(b) were produced using the AUSMDV flux solver, with the entropy-fix suggested in Reference 102. When an expansion shock at a sonic point is de-

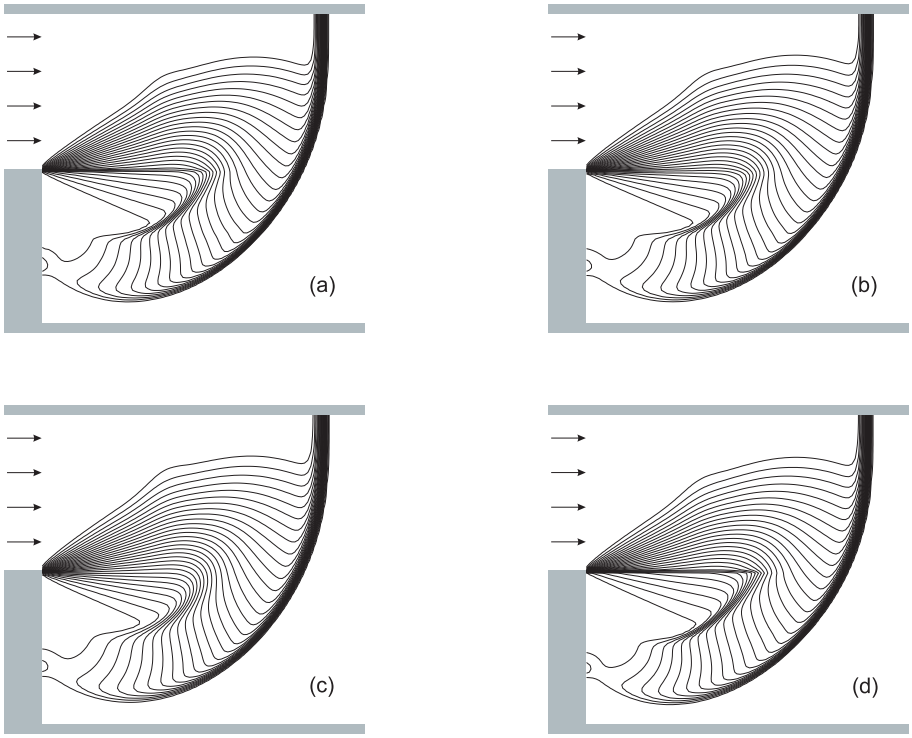



Figure 5.6: Flow over a rearward facing step computed using different flux solvers, showing spurious expansion shocks in some cases. (a) AUSM, (b) AUSMDV, (c) EFM, (d) Approximate Riemann solver.

tected, the entropy-fix works by adding dissipation terms to the fluxes. Although absolute flow speed in the expansion shock region is significantly greater than sonic, the component of velocity normal to the expansion shock is exactly sonic. Since the grid cells are aligned to the expansion shock, the entropy-fix will only add dissipation to fluxes in the normal direction. Even with the fix, however, an expansion shock is found embedded in the AUSMDV solution. We observe that the severity of the shock is not as bad as that produced by the AUSM, though.

Simulation results using the EFM are shown in Figure 5.6(c). The EFM is dissipative enough so as to not encounter the spurious expansion shock. Unfortunately, this dissipation acts throughout the entire flowfield and blurs the contact surface somewhat. Figure 5.6(d) was produced using the approximate Riemann solver, and shows an expansion shock which is clearly the strongest of all the schemes. The Riemann solver, though, does resolve the shock and contact surfaces acceptably, especially considering that no reconstruction is used.

When reconstruction is switched on, expansion shocks are not produced by any of the flux solvers. As expansion shocks are monotone and weak, the limiters are not fully activated and allow some high order interpolation across the wave. Because the shock is

completely bounded by two cell centres, and a four cell interpolation stencil is used, it is suggested that gradients drawn from outside the zone of sudden expansion contribute to the removal of the shock. In practical CFD calculations, reconstruction is almost always used. Thus, regardless of the mechanism by which reconstruction averts the spurious shock, explicit entropy fixes are seemingly not required in SF3D. It remains a concern, however, that most of the underlying flux algorithms are so readily capable of violating the second law.

## 5.6 Influence of Chemical Reaction on Flow Noise

In Section 3.6, it was stated that the introduction of chemical reactions into a system can cause increased noise and aggravate flow instability. To check the susceptibility of the different flux solvers to such noise, they are now tested in a flow of a reacting gas around a blunt-body. Carbon dioxide in chemical equilibrium will be used, since we have already seen that this gas promotes instability, and in addition the equilibrium model is not too computationally expensive.

For the test, the inflow gas is set to a temperature of 300 K and density of 0.01 kg/m$^3$, and has a free-stream speed of 1.5 km/s. A cylinder is used for the blunt body, and is meshed with a 30 by 50 cell grid. The cylinder has a 1 m radius, and solutions are marched forward in time to 20 ms.

The solution in Figure 5.7(a) was generated using the AUSM, second-order time integration, and MUSCL reconstruction with the van Albada limiter. A generally clean flow field is observed, however pressure oscillations are visible near the wall, particularly apparent towards the downstream boundary. Such oscillations are a frequently cited defect of the AUSM scheme.[105] It is unfortunate that the noise is located at the wall, since wall measurements usually form the basis for comparison with experiment, and are probably the most sought-after simulation results. The maximum magnitude of the oscillations is about 3% in pressure, measured peak to trough, with amplitude decreasing away from the wall. It will be shown in Chapters 6 and 7 that this level of error is unacceptably large for the HYFLEX simulations, since it is of the order of pressure variations that need to be resolved.

Figure 5.7(b) shows results from an AUSMDV simulation. Smooth pressure contours are observed throughout the solution domain, with no visible noise or spurious flow features. The absence of noise is an advantage since instabilities and their associated error are unlikely to triggered, however the smooth contours in no way validate the solution to be accurate. Validation tests are performed later in this chapter, since, unfortunately, there is no independent experimental data available to check these particular computed results.

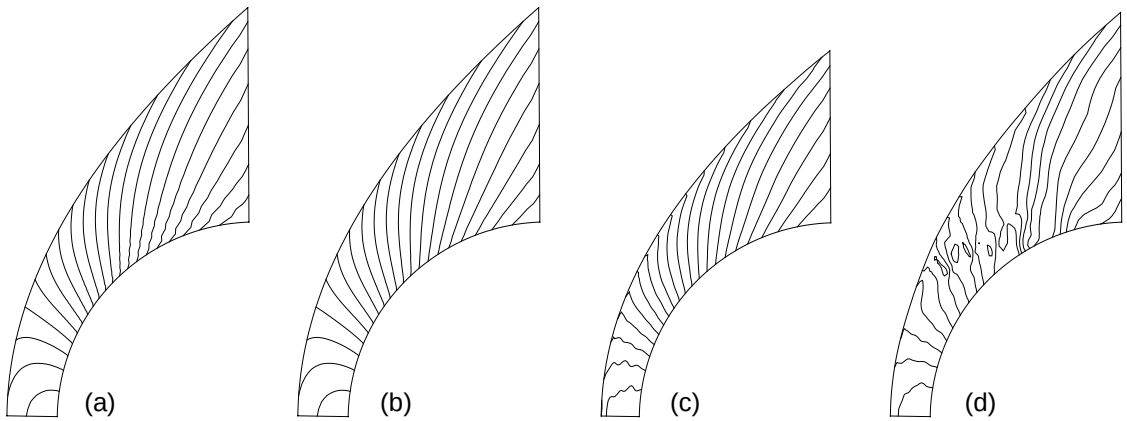



Figure 5.7: Simulation of carbon dioxide flow around a cylinder, in chemical equilibrium. Isopressure contours are shown for (a) AUSM, (b) AUSMDV, (c) EFM, and (d) approximate Riemann solver.

The solution produced using the EFM is presented in Figure 5.7(c). The EFM produces significant noise in the stagnation region, as well as flow disturbances in cells just downstream of the shock. The noisy flow has caused a reduced shock standoff distance and increased shock curvature to be computed.

Finally, results from the approximate Riemann solver are shown in Figure 5.7(d). Although the shock curvature roughly matches with the AUSM and AUSMDV solutions, the standoff distance has reduced to that calculated by EFM. A large amount of noise is present throughout the entire domain, from stagnation point to outflow boundary. The worst noise, however, is observed running along the sonic line from shock to body. With a 3 km/s inflow speed, the Riemann solver breaks the positivity condition and fails to produce a solution at all.

Unlike the EFM and approximate Riemann solver, the splittings used in AUSM and AUSMDV do not require any detailed assumptions about the thermodynamic behaviour of the test gas. This is the key to good flux solver performance for reacting gases with arbitrary equations of state. The approximate Riemann solver, for example, requires knowledge of the ratio of specific heats at the interface, which is calculated as a density averaged value from the left and right interface states.[96] It is suggested that this approximation contributes to the noise production, and likewise for a similar approximation contained in the EFM scheme.[98] Colella and Glaz[163] suggest alternative approximate techniques for solving the Riemann problem for any gas with a convex equation of state, and Grossman and Cinella[164] give algorithms which incorporate nonequilibrium chemistry coupling into a range of FDS and FVS schemes. These improvements, though, do attract increased computational effort.

Table 5.2: Code speed with different flux solvers and reconstructions.

| Flux Solver | Reconstruction (%) | Flux Calculation (%) | Total Time (s) |
|---|---|---|---|
| AUSM[a] | 24.6 | 10.7 | 635 |
| AUSMDV[a] | 24.2 | 16.1 | 632 |
| AUSMDV[b] | 11.1 | 19.3 | 548 |
| EFM[a] | 21.3 | 24.5 | 736 |
| Riemann[a] | 17.6 | 39.6 | 915 |

[a] Spatially sensitive MUSCL reconstruction and the van Albada limiter.

[b] MUSCL reconstruction and the min-mod limiter.

## 5.7 Code Speed

While not strictly part of verification, we now investigate the practical issue of code speed. Albeit not particularly relevant to the simple tests presented in this chapter, computational expense does rank as a significant consideration for the HYFLEX simulations, which can take several days to complete. Again, flow around a cylinder on a 30 by 50 grid is used as the test problem, with calorifically perfect air at a temperature of 300 K and density of 0.01 kg/m$^3$ approaching the body at 3 km/s. Simulations are marched forward in time to 10 ms, and bow shocks resolved using shock capturing.

A Silicon Graphics Origin 2000 supercomputer, installed with 64 MIPS R10000 processors running at 195 MHz, was used to perform the speed tests (see Figure C.1). Although the machine is capable of executing code in parallel, each test was conducted on a single processor. A summary of results is presented in Table 5.2. The relative proportion of the total computer time used by the flux solver and reconstruction technique is shown for each test. Note that total time refers to the time used by the computer to run the job in isolation, including system and input-output overheads, but excluding the effects of other processes which may also be running. Most of the time unaccounted for by reconstruction and flux calculation, though, is due to the equation of state, time integration, boundary conditions, memory allocation, flux transformations, and other housekeeping tasks.

The coded implementation of the AUSM and AUSMDV algorithms were the fastest flux solvers, with the entire code running at 104 and 103 $\mu$s per finite volume cell per complete second-order Runge-Kutta timestep, respectively. Using the min-mod limiter and equispaced MUSCL reconstruction significantly reduced the total simulation time, being over two and a half times faster than the spatially sensitive reconstruction with modified van Albada limiter. The EFM was slower than the AUSM and its derivative, causing

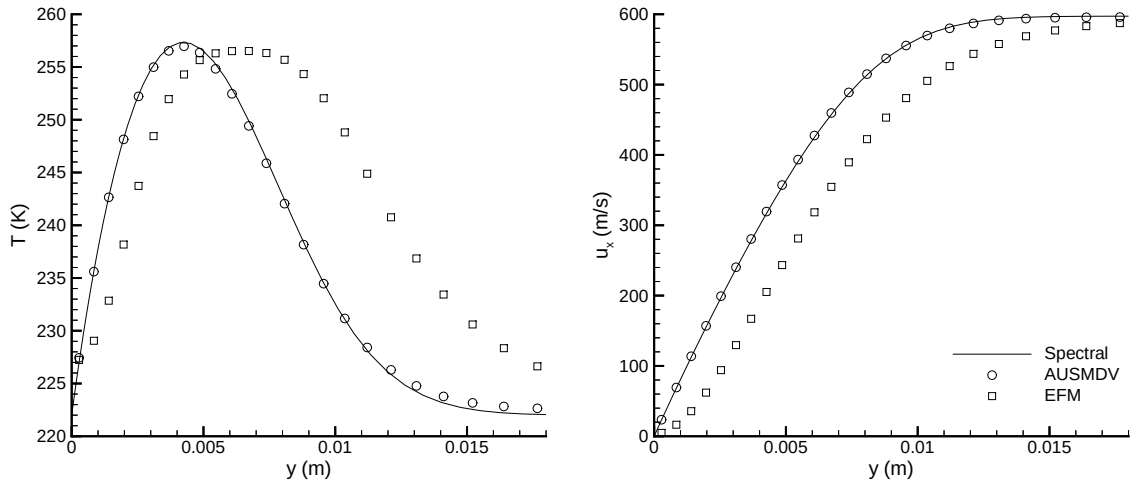



Figure 5.8: Boundary layer profiles for a flat plate, calculated using different flux solvers.

the entire code to clock 119 $\mu$s per cell per step. Slower again was the approximate Riemann solver, which consumes computational effort far outweighing that expended during reconstruction. Using the Riemann solver, the complete code runs at 151 $\mu$s per cell per step for this test case.

## 5.8 Viscous Flow along a Flat Plate

Verification of the correct implementation of diffusive terms in the Navier-Stokes equations is now conducted, by examining boundary layer development in a viscous, laminar stream over a flat plate. The simulation is performed with flow conditions corresponding to those previously used by other investigators for this test case.[57,165]

A two-dimensional plate, aligned with the $x$ and perpendicular to the $y$ coordinate directions, is meshed with 50 cells from $x = 0$ to $x = 1$ m. Normal to the plate, 80 cells are clustered towards the surface to capture the development of the boundary layer and the shock formed off the leading edge. An isothermal, nonslip wall condition is imposed at the plate surface, keeping it at a constant and uniform temperature of 222 K. The flow conditions upstream of the plate are

$$\mathbf{u} = 597.3\hat{\mathbf{i}}\ \text{m/s},\ \ \rho = 0.00404\ \text{kg/m}^3,\ \ T = 222\ \text{K}.$$

The test gas is calorifically perfect air, and Sutherland's law is used to describe its viscosity as a function of temperature. Simulations are stopped after 8 ms of flow time.

Results in Figure 5.8 show boundary layer profiles computed using SF3D with the AUSMDV and EFM, at a position $x = 0.91$ m from the leading edge of the plate. Also

shown are results produced by the authors of an accurate spectral collocation method.[166] The AUSMDV and spectral profiles are in very good agreement, both predicting similar boundary layer thicknesses and gradients. Towards the outer edge of the boundary layer a slight divergence is noticed, presumably because the spectral method does not incorporate the leading edge shock, and is thus modelling the core flow at a slightly different state. In contrast to the AUSMDV, the EFM computes a smeared boundary layer of exaggerated thickness and does not match the spectral solution. The dissipation that was so advantageous in the odd-even decoupling test case (Figure 3.8) is now working against the accuracy of the EFM.

When performed using the approximate Riemann solver or the AUSM, simulation results were of a similar accuracy to that enjoyed by the AUSMDV. Additionally, the flat plate problem was run at different orientations in order to verify that viscous stresses are correctly calculated in each coordinate direction.

## 5.9 Comparison with Drag Experiments

The accurate prediction of aerodynamic drag on blunt bodies is an important part of planning the trajectory for entry and re-entry vehicles. In this, the first validation test case, we compare numerically simulated drag of three bodies, with that measured in a hypervelocity expansion tube experiment by Smith *et al.*[109] Figure 5.9 shows the bodies to be tested, which are namely a 30 degree cone, the Apollo heat shield that has been examined in earlier chapters, and a blunted cone of similar geometry to the heat shields used in the Viking and Pathfinder missions. Carbon dioxide is used as the test gas, to emulate the chemistry encountered during Martian atmospheric entry.

For all experiments and simulations, the free-stream flow conditions were

$$\mathbf{u} = 7.5\hat{\mathbf{i}}\ \text{km/s},\ \ T = 2990\ \text{K},\ \ P = 15.6\ \text{kPa},\ \ \rho = 0.0194\ \text{kg/m}^3.$$

The uncertainty in obtaining the correct pressure and temperature in the experiments is estimated at $\pm 2\%$, and the uncertainty in density estimated at $\pm 10\%$.[109] As a result, the experimentally measured drag also has error bounds of $\pm 10\%$. The operation of the expansion tube causes a somewhat dissociated free-stream flow, with the composition

$$C_{\text{CO}_2} = 0.310,\ C_{\text{O}_2} = 0.189,\ C_{\text{CO}} = 0.439,\ C_{\text{C}} = 0.000,\ C_{\text{O}} = 0.062,$$

in terms of mass fraction.

In the short test time available, experimental drag was measured by using an impulse response function to deconvolve signals obtained from a stress wave force balance.[167] For

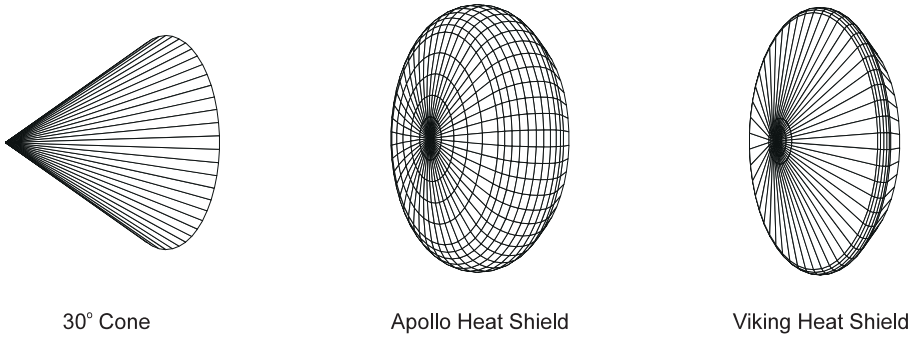



Figure 5.9: Models used in the drag experiments and simulations.

the 30 degree cone at zero angle of attack, a nondimensional drag coefficient was 0.57 was recorded. The coefficient is defined as

$$C_{\mathrm{D}} = \frac{2F_{\mathrm{D}}}{\rho|\mathbf{u}|^2 A}, \tag{5.4}$$

with the density and velocity corresponding to that in the free stream, and $F_{\mathrm{D}}$ being the measured drag force. The form area of the body is denoted by $A$.

The approximate Riemann solver, MUSCL reconstruction, min-mod limiter, and equilibrium carbon dioxide model were used to simulate the viscous cone flow. The solution was marched in time until the incident flow had passed the body by a length of at least 25 times the cone base diameter. The total force exerted on the body by the flow may be expressed as the sum of pressure and friction components:

$$\mathbf{F} = \mathbf{F}_{\mathrm{P}} + \mathbf{F}_{\mathrm{T}}, \quad \text{where } \mathbf{F}_{\mathrm{P}} = \int P\, d\mathbf{A}, \text{ and } \mathbf{F}_{\mathrm{T}} = \int \mathbf{T}\, d\mathbf{A}. \tag{5.5}$$

Because the cone was aligned with the flow, the magnitude of the calculated total force $|\mathbf{F}|$ is equivalent to the drag force. The computed coefficient of cone drag is 0.57, in excellent agreement with the experimentally measured value. Of the total drag coefficient, 0.54 is attributable to form drag and the remaining 0.03 is due to skin friction. Taylor and Maccoll[168] give an analytical treatment for pressure over a cone, which in this case yields a form drag prediction of 0.54 and verifies the simulation results.[109] For the cone flow, computed drag results were found to be relatively independent of the reaction model; a simulation run with frozen reactions also resulted in a total drag coefficient of 0.57. This is not unexpected since pressure, which is responsible for the bulk of the drag force, is foremost a mechanical quantity and is primarily determined by the free-stream momentum.

For the blunt bodies, simulations were performed using the AUSM with shock-fitting to avoid the occurrence of carbuncles. Both models were tested at zero angle of attack. In case of the Apollo heat shield, the computed total drag coefficient was 1.41 for carbon

dioxide in chemical equilibrium, and 1.42 when the frozen reaction model was used. The coefficient of drag measured in the expansion tube was 1.54. Computed and measured results agree to within the 10% tolerance imposed by experimental uncertainty. Likewise, the numerical and experimental drag over the Viking heat shield agreed to within experimental uncertainty. The measured drag coefficient was 1.78, and the computed values were 1.68 and 1.71 with the equilibrium and frozen reaction models respectively. In Reference 169, Gnoffo *et al.* present results from nonequilibrium carbon dioxide simulations of the same heat shield geometry, over a range of free-stream conditions. Although none of these conditions fully correspond to those used in the present work, an approximate comparison can be made by interpolating the reported data for an equivalent velocity of 7.5 km/s. A drag coefficient of 1.67 is subsequently obtained, which is in close agreement with the value calculated by SF3D assuming chemical equilibrium.

It is reiterated that the primary purpose of SF3D is to accurately simulate the pressure distribution around blunt bodies. Although the effect of chemistry on the drag of the heat shields may seem small, at 1% and 2%, it will be shown in Chapter 7 why this kind of variation in pressure is significant for the calibration of air data systems. Further, the good agreement observed in measured and computed drag is a necessary, but not sufficient, condition for validation. Because drag is an integrated quantity, local inaccuracies in the flowfield may not immediately be apparent in such results. More validation test cases are now needed, which incorporate greater sensitivity to the distribution of flow state throughout the entire flowfield.

## 5.10 Shock Shape and Standoff

Shock shape and standoff distance are flow features that are reasonably easy to measure in experiment, using optical techniques such as luminosity photography, Schlieren imaging, interferometry, and shadowgraphs.[170] The position and shape of the shock is strongly dependent on flow physics, and will not be correctly predicted by a CFD solver with an improper implementation of the Navier-Stokes or Euler equations. Hence, comparison of these features constitutes a useful element in the complete validation of a CFD code. In this section, we consider shock curvature and standoff distance for the flow of frozen, calorifically perfect air. The effects of nonequilibrium reactions and real gas thermodynamics on shock standoff will be examined in the next section.

For low temperature air, Ambrosio and Wortman used experimental results to develop correlations for shock standoff distances as a function of Mach number.[171] Specifically,

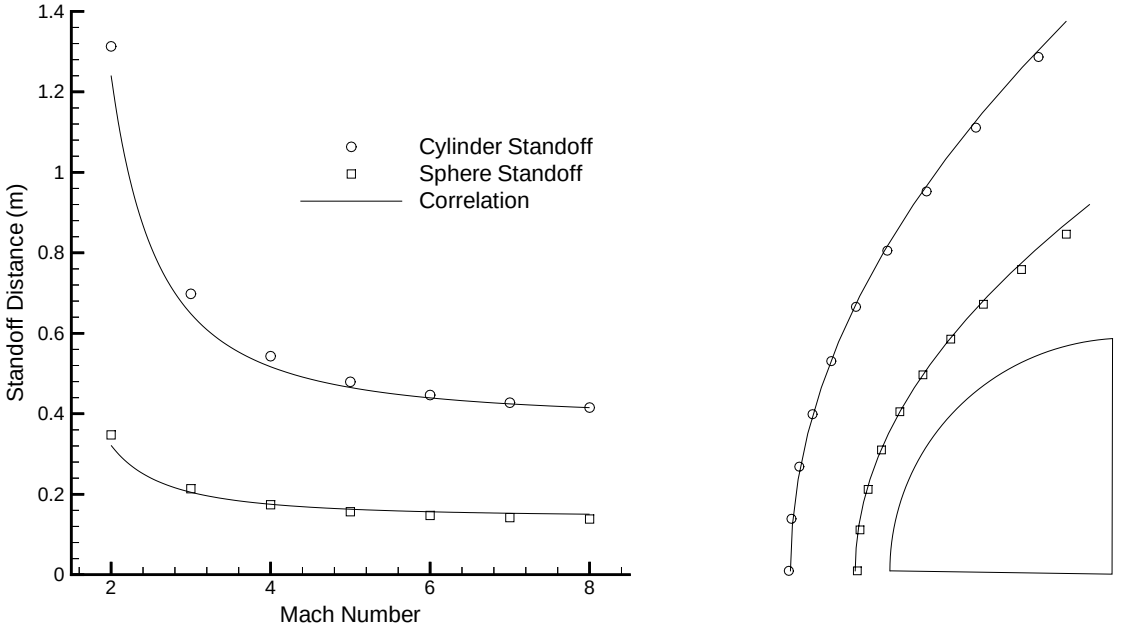



Figure 5.10: Standoff distances obtained by simulation compared with predictions from the correlation of Ambrosio and Worman (left), and comparison of shock curvature at Mach 6 with Billig's correlation (right).

the correlations are

$$\Delta/r_\mathrm{b} = 0.143\,\exp(3.24/M^2), \quad \text{and} \tag{5.6}$$

$$\Delta/r_\mathrm{b} = 0.386\,\exp(4.67/M^2), \tag{5.7}$$

for spheres and cylinders respectively. Standoff is denoted by $\Delta$, and $r_\mathrm{b}$ is the body radius. Simulated standoff distances for Mach numbers ranging from two to eight are presented in Figure 5.10, together with the curves of Equations 5.6 and 5.7. The simulations were performed using AUSMDV, shock fitting, and MUSCL reconstruction with the van Albada limiter. For the purposes of comparison, $r_\mathrm{b}$ is set to 1 m, and standard atmospheric conditions are used as the free-stream gas state. Simulations were performed on a 30 by 50 cell grid. Good agreement is observed for both the sphere and cylinder geometries at high Mach numbers, with results diverging slightly at lower speeds.

Some empirical correlations for shock shape based on experimental results are listed by Billig in Reference 172. Shock shape is constrained to a hyperbolic curve fit which asymptotes to the freestream Mach angle,[173] defined by $\mu = \arcsin(1/M)$. The equation for the shock in Cartesian coordinates is

$$x = r_\mathrm{b} + \Delta - r_\mathrm{c}\cot^2\mu\left[\sqrt{1+\left(\frac{y\tan\mu}{r_\mathrm{c}}\right)^2}-1\right], \tag{5.8}$$

where $r_c$ is the radius of curvature. The radii of curvature for spheres and cylinders are respectively defined by the empirical formulas

$$r_c/r_b = 1.143 \exp[0.54/(M-1)^{6/5}], \text{ and} \tag{5.9}$$

$$r_c/r_b = 1.386 \exp[1.8/(M-1)^{3/4}]. \tag{5.10}$$

A comparison of Billig's correlations with simulation, for air at Mach 6, is shown in Figure 5.10. Good agreement is obtained in both sphere and cylinder test cases. Since the correlations are based on a number of experimental results containing scatter, perfect agreement is not expected.

## 5.11 Dissociating Flow over a Cylinder

The final validation test case in this chapter is the flow of nitrogen over cylinders, at speeds high enough to induce nonequilibrium dissociation, but low enough to preclude ionization. Under such conditions, just one reaction occurs in the nitrogen system

$$N_2 \rightleftharpoons N + N, \tag{5.11}$$

thus making it a useful, simple problem on which to test both the flow and nonequilibrium kinetics models in SF3D.

In References 174 and 175, Hornung presents experimental interferograms and shock standoff distances that were recorded in the T3 shock tunnel[176] for nitrogen flows over cylinders of various diameters. The stated free-stream test conditions are

$$\mathbf{u} = 5.5\hat{\mathbf{i}} \text{ km/s}, \ \rho = 0.0055 \text{ kg/m}^3, \ T = 1400 \text{ K}, \ C_N = 0.07, \text{ and } C_{N_2} = 0.93.$$

The free-stream gas pressure is 2446 Pa, assuming that the nitrogen mixture is thermally perfect and obeys an equation of state based on the thermodynamic data in Appendix A.

To examine the effect of gas chemistry on results, simulations were performed using frozen, nonequilibrium, and equilibrium kinetic models. The frozen and nonequilibrium simulations are identical in practice, except for setting the reaction rate of Equation 5.11 to zero in the former case. When a system is in chemical equilibrium, however, species concentrations become a state variable, and it becomes impossible to satisfy the free-stream test conditions and mixture composition at once, due to overconstraint. With the already stated temperature and density, a nitrogen mixture at chemical equilibrium contains essentially no monatomic species. Also, the mixture pressure reduces to 2286 Pa at chemical equilibrium, because the system has relaxed and the molar density has decreased. These factors, while unavoidable, will slightly influence the results.

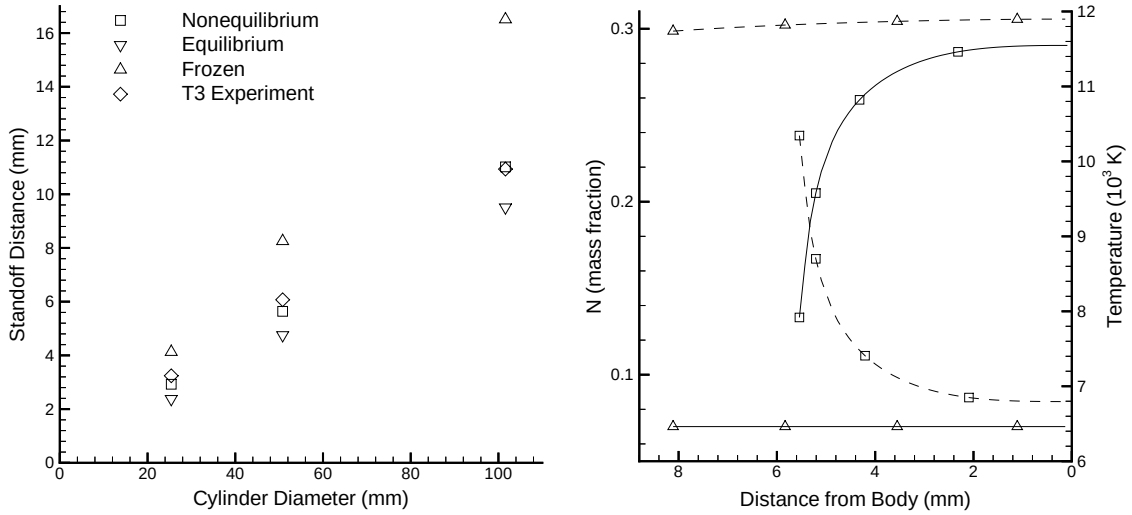



Figure 5.11: Standoff distances for dissociating nitrogen flow over a cylinder (left), and gas composition and temperature along the stagnation line for the 50.8 mm diameter cylinder (right). —— Mass fraction of monatomic nitrogen, - - - temperature.

Again, MUSCL reconstruction with the van Albada limiter, shock-fitting, and the AUSMDV are used to generate the simulations. Standoff distance results for cylinders with 25.4, 50.8 and 101.6 mm diameters (equivalently, 1, 2 and 4 in) are presented in Figure 5.11. Simulations performed assuming a frozen reaction yield standoff distances significantly larger than those observed in experiment, and for simulations assuming chemical equilibrium the standoff distances are observed to be too small. Simulations incorporating nonequilibrium reaction rates give results that match reasonably well with experiment. In particular, results for the largest cylinder (101.6 mm diameter) are in excellent agreement. The standoff distances calculated for the smaller cylinders are marginally shy of the experimental values, perhaps because the reaction model is too conservative, or possibly the stated experimental test conditions were not exactly achieved in the shock tunnel.

The large variation in standoff between frozen and equilibrium simulations highlights the sensitivity of the flow field to chemistry. The mechanism by which chemical reaction affects standoff distance is evidenced by the graph in Figure 5.11, showing gas composition and temperature along the stagnation line of the 50.8 mm cylinder. In the case of the nonequilibrium model, we observe that dissociation of the diatomic nitrogen soaks energy from the flow and causes temperature to decrease towards the stagnation point. In contrast, a slight temperature increase is observed for the frozen reaction model, since post-shock kinetic energy is being converted to internal energy, and there is no reaction pathway to relax the system. As would be expected, equilibrium calculations give a stag-

nation temperature lower than that of the nonequilibrium simulation (6415 K, not shown). Because post-shock pressure in the stagnation region is primarily due to the impact of momentum carried in the free-stream gas, it is only weakly dependent on the gas model used; there is a 3% difference in stagnation pressure between frozen and equilibrium simulations. Hence, in this case, the equation of state for a thermally perfect gas dictates that any large temperature variation must cause a roughly proportional variation in specific volume, assuming that the gas constant $R$ indeed remains constant. For a reacting gas, then, the decreased post-shock temperature causes decreased specific volume, and allows the shock to encroach upon the body.

It is noted that there is a reasonably large discrepancy between the stagnation temperature calculated in the nonequilibrium simulation, and stagnation temperature produced using an equilibrium gas model. This is thought to be a result of quenching,[177] where the temperature loss caused by dissociation slows the reaction rate and retards any further dissociation so that, in the available flow time, chemical equilibrium is never reached.

An experimental interferogram,[174] and an interferogram computed using the nonequilibrium model, are shown in Figure 5.12 for the 50.8 mm cylinder. Light with 533 nm wavelength and a cylinder of length 152.4 mm (6 in) were used to produce the experimental infinite fringe interferogram. Computed shock shape is superimposed onto the experimental result to allow easier comparison. As indicated by Figure 5.11, the computed shock standoff distance is slightly smaller than the standoff in experiment, but after accounting for this difference the shock shapes match quite well. The shapes of the computed interferogram fringes are similar to those observed in experiment, however the two results differ by approximately one fringe by the time flow reaches the downstream boundary. The difference is most likely due to a combination of factors: inaccuracies in the nitrogen thermodynamic and reaction models, three-dimensional effects caused by flow leakage around the sides of the cylinder in experiment,[174] and uncertainty in the stated free-stream shock tunnel conditions. Interferograms produced using frozen and equilibrium chemistry models exhibited much worse agreement with experiment, in terms of both fringes and shock shape.

The pressure distribution around the cylinder depends on the manner in which the bow shock processes upstream gas. In particular, the local shock angle has a significant bearing on the post-shock pressure produced. Since gas chemistry affects shock shape, this is one mechanism by which the chemistry model influences the distribution of body pressure. Thus, to obtain a highly accurate simulation of blunt-body surface pressures, the correct shock shape and a reasonably accurate chemistry model are requirements.

For reference and comparison with the interferogram, computed isopycnics are presented in Figure 5.12. Additionally, the figure shows concentrations of monatomic nitro-

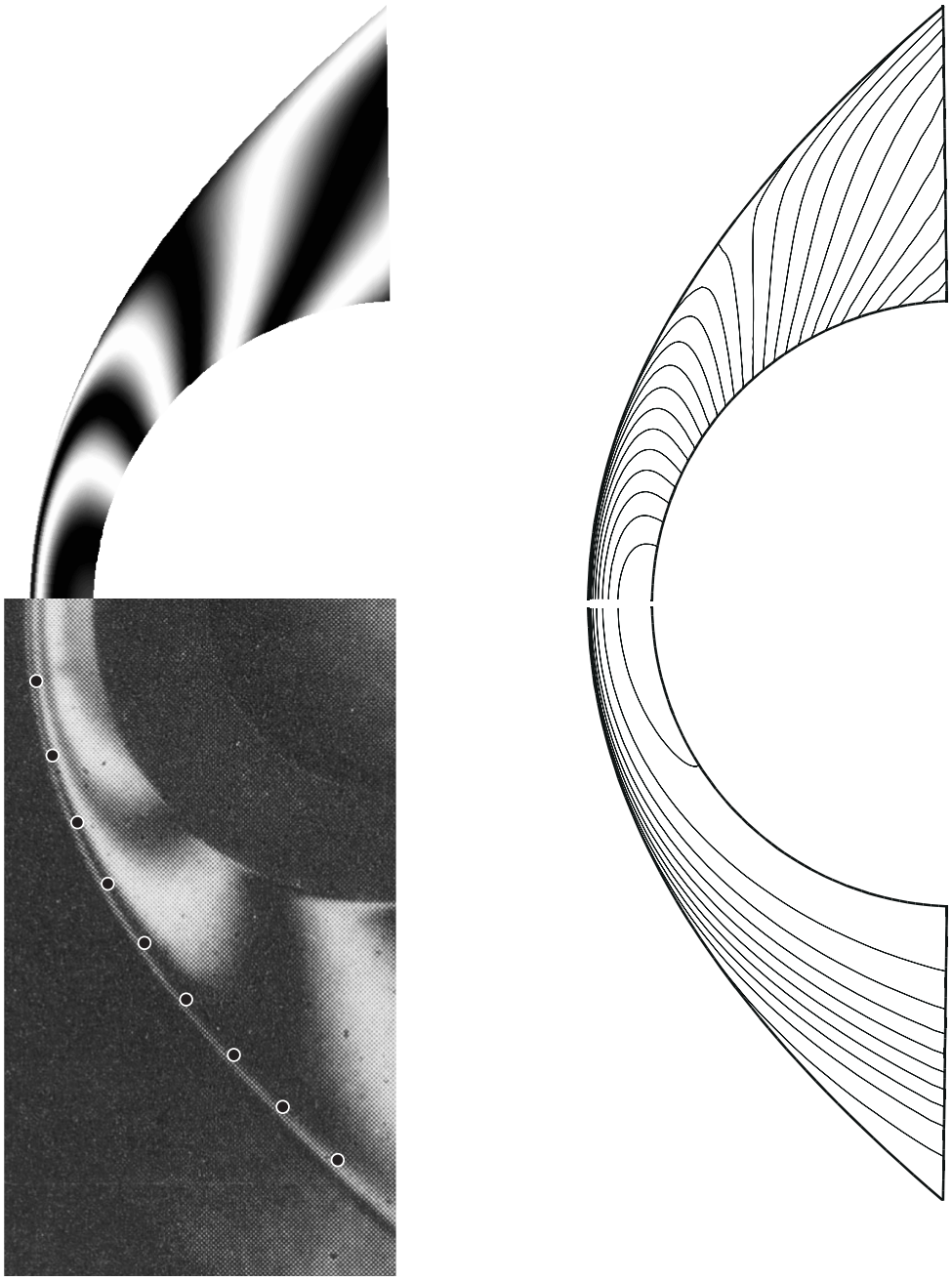


Figure 5.12: Computed (top left) and measured[174] (bottom left) interferograms for dissociating nitrogen flow over a cylinder. For comparison, filled circles are used to mark the computed shock shape on the experimental interferogram. Computed isopycnic contours ranging from 0.0106 to 0.0596 kg/m$^3$ (top right), and contours of mass fraction for monatomic nitrogen ranging from 0.0786 to 0.282 (bottom right).

gen throughout the flow field, indicating regions of rapid dissociation. Little change in mixture composition is observed close to the body, probably due to the downstream flow of species that have been quenched in the stagnation region.

## 5.12 Selection of Code Elements

The following chapter deals with simulation of flow around the blunt-nosed HYFLEX vehicle. Based on the test cases in this chapter, as well as evidence from the literature, numerical experiments, and arguments in Chapters 2, 3, and 4, the optimal code elements for blunt-body simulation are now selected.

The AUSMDV flux solver is chosen to perform the simulations because, of the tested flux algorithms, it was the only one displaying robustness, accuracy, and efficiency at once. The AUSMDV is selected over the AUSM, since it did not suffer post-shock and wall pressure oscillations in the reacting gas blunt-body tests. Also, the AUSMDV exhibits better grid convergence than the AUSM. The approximate Riemann solver, while being accurate for simple equations of state and displaying a marginally better grid convergence than the AUSMDV, also fails for reacting gases with complicated equations of state. In addition, while both the approximate Riemann solver and AUSMDV suffer the carbuncle effect, the Riemann solver is plagued by the problem at lower grid resolutions with less instigation. Another disadvantage of the approximate Riemann solver is its computational expense, slowing the code down by as much as 45%. While robust, the EFM was shown to be excessively dissipative in boundary layer calculations, and thus can not alone be used to simulate the HYFLEX flow. While displaying a good spatial convergence rate in the supersonic vortex test, the thermodynamic assumptions inherent to the EFM solver caused it to produce excessive noise on a reacting gas blunt-body test case.

The modified van Albada limiter and MUSCL interpolation constitute the selected reconstruction technique. As described in Section 3.2.3, the spatial awareness of the reconstruction copes with the relatively abrupt grid clustering that will be used on the HYFLEX grid. Additionally, the differentiable property of the reconstruction facilitates convergence to steady-state for simulations of the HYFLEX in flight. Although it was shown in Section 5.7 that this reconstruction is more than twice as slow as a min-mod MUSCL implementation, overall code speed is only degraded by 15%. This is more than accounted for by the increased accuracy for unequally spaced cells, and the ability to use a coarser grid.

Because the AUSMDV flux solver is prone to the carbuncle effect, shock fitting — rather than shock capturing — will be used on all the HYFLEX simulations. In addition to maintaining stability for low dissipation solvers, shock fitting also increases efficiency and produces shock waves that are both accurate and sharp, as shown in Chapter 4.

CHAPTER 6

# Simulation of the HYFLEX

The hypersonic flight experiment (HYFLEX) undertaken by the Japanese National Aerospace Laboratory (NAL) and National Space Development Agency (NASDA), was an initiative aimed at obtaining fundamental data on entry vehicle aerodynamics and aerothermodynamics.[178] The program culminated in February 1996 when the HYFLEX vehicle successfully performed an aerobraking manoeuvre to decelerate and descend to sea level, from an initial height of 110 km and Mach number of 13. During the flight, an array of pressure sensors was used to measure the distribution of surface pressure around the nose of the vehicle.

This pressure data provides an ideal base to evaluate the accuracy of the SF3D code, and the relative usefulness and accuracy of some other techniques that are available to simulate hypervelocity flow. These other techniques include modified Newtonian theory, and scale-model shock tunnel experiments. An accuracy assessment for each simulation method is presented in this chapter so that, in the following chapter, their suitability for use in air data system calibration can be determined.

As well as examining the usefulness of each simulation technique alone, in this chapter they are also used in conjunction to help explain and alleviate the particular shortcomings of each method. Shock tunnel experiments are capable of simulating the high temperature gas phenomena occurring in flight, but are subject to the effects of scale. Particularly affected by scale is the chemical composition of the flow. Adverse scale effects can be reduced by applying binary scaling,[179] and CFD simulation is employed to examine how successful this approach is. In addition, the effects of nonuniformity in the shock tunnel test flow are examined using numerical methods. For the experiments examined here, the nose of the model was placed near the edge of the core flow produced by the

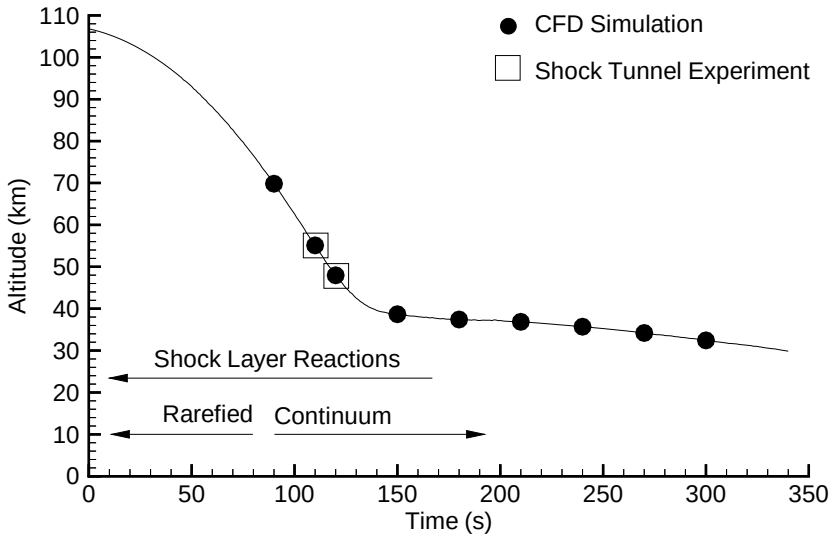



Figure 6.1: The HYFLEX time-altitude map.

shock tunnel nozzle. At the high angle of attack used, this placed the lower part of the vehicle surface in the core flow, and some of the upper surface in a region of nonuniform flow.

CFD simulations of the full-scale vehicle allow the investigation of the relative importance of real gas phenomena encountered in flight. Such phenomena include chemical nonequilibrium, and boundary layer growth. The flight data and shock tunnel results are used to check that the numerical gas models are working correctly, and producing reasonable results. Quick and simple comparison between flight and scale-model data is made via modified Newtonian theory.

A time-altitude map of the HYFLEX flight is presented in Figure 6.1. Indicated on the map are the flight times where numerical and experimental simulations of the vehicle have been performed. In this chapter, effort is concentrated around the 120 s flight time, where the flow regime may be considered continuum, vehicle speed is close to maximum, and nonequilibrium chemistry effects are strong. By about 170 s after separation, the high temperature effects are weak and the vehicle has decelerated to 2.28 km/s (Mach 7.1). At 300 s the vehicle speed is 0.90 km/s (Mach 2.93).

A diagram of the HYFLEX geometry is shown in Figure 6.2. The flight vehicle body is 4 m long, and has a nose radius of 0.4 m. Also shown is the cruciform array of nine pressure sensors around the vehicle nose (labelled ps1 – ps9). More detail of the HYFLEX vehicle and nose layout is presented in Appendix D. The numerical simulations discussed in this chapter consider flow over the nose region, and along the undersurface for a dis-

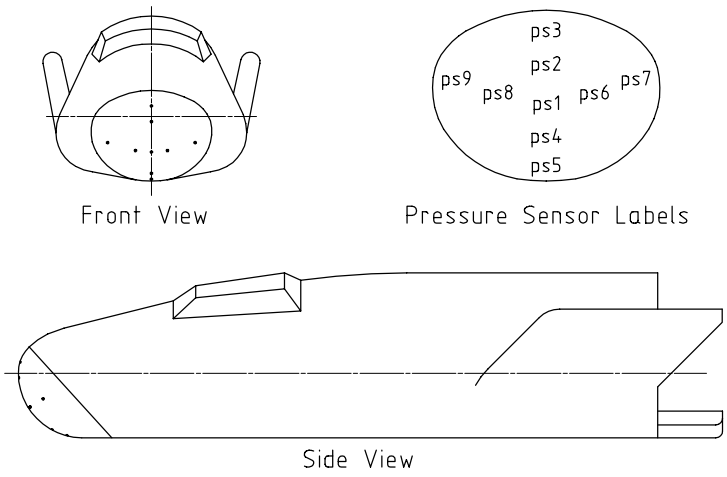



Figure 6.2: The HYFLEX vehicle and nose pressure tapping layout.

tance of 1.4 m. This region is sufficient to include all of the pressure tappings and the subsonic flow region extending from the stagnation point. The shock tunnel model was a one-tenth scale model of the first 1.4 m of the flight vehicle, and includes pressure tappings at points corresponding to the flight vehicle pressure transducers. Thus, the primary data that will be discussed are the pressure measurements recorded during flight and experiment, and the computed pressures at corresponding locations in the numerical simulations.

## 6.1 Computational Flow Modelling

In Chapter 2, it was stated that the Navier-Stokes equations are only truly applicable in the continuum flow regime. Hence, to ensure that the SF3D code is valid for the flight conditions simulated in this chapter, the Knudsen number for the highest altitude simulation is now evaluated. The Knudsen number is conventionally defined as the ratio of mean free molecular path to a characteristic body dimension,[180]

$$\mathrm{Kn} = \frac{\lambda}{L} \tag{6.1}$$

and gives an indication of the importance of rarefied gas effects. By this definition, then, the continuum flow approximation is more accurate at smaller Knudsen numbers. The mean free path for a single species gas is given by[180]

$$\lambda = \frac{1}{\sqrt{2} n \pi \Omega}, \tag{6.2}$$

where $n$ is the gas number density in molecules per unit volume, and $\Omega$ is the collision cross section, which is roughly equal to the square of molecular diameter. Table D.1

in Appendix D lists the atmospheric conditions at 69.8 km altitude, corresponding to the most rarefied simulated condition. For simplicity, and without significant loss of accuracy, the Knudsen number at this altitude is calculated assuming that the free-stream gas is composed solely of diatomic nitrogen. After obtaining the collision cross section relevant to the atmospheric temperature at this altitude from Reference 63, and choosing the nose radius as characteristic body dimension, the Knudsen number evaluates to $7.2 \times 10^{-4}$. Transition to rarefied flow begins at a Knudsen number of about 0.03, and thus the 69.8 km condition is an order of magnitude within the continuum regime.[13]

Air is assumed to be thermally, but not calorifically perfect for the simulations. According to Hansen and Heims,[181] a 0.3 m radius sphere entering the atmosphere along the HYFLEX velocity-altitude trajectory will experience both chemical equilibrium and nonequilibrium regimes. This result is applicable to the forebody of the HYFLEX vehicle, since its nosecap is a approximately a 0.4 m radius sphere. However, the five species, seventeen reaction nonequilibrium model of Section 2.10.3 is always used to ensure continuity in accuracy amongst simulations, regardless of whether frozen or equilibrium kinetic limits are being approached.

Since the post-shock gas temperature of the HYFLEX rarely exceeds 5500 K, and ionization in air typically occurs at temperatures above 8000 K,[59] it is assumed that no ionized species or electrons are present in the flow. Additionally, results in Reference 181 indicate that the HYFLEX flow field is always in thermal equilibrium, and justify use of the single-temperature thermodynamic model in SF3D. For completeness, however, this is now verified. The collision frequency of molecules in a single species gas is

$$\nu = n\Omega\sqrt{\frac{8\pi kT}{m}}, \tag{6.3}$$

where $k$ is Boltzmann's constant, and $m$ is molecular mass. Making the assumption that air is completely composed of diatomic nitrogen for simplicity, and using the post-shock flight conditions at 120 s after experiment commencement (see Table D.1), a collision frequency of $2.6 \times 10^8$ s$^{-1}$ is obtained. One hundred intermolecular collisions are typically required for thermal equilibrium, and are thus completed within $3.7 \times 10^{-7}$ s. At the convective flow speed in the shock layer, the gas travels roughly 0.8% of the shock standoff distance in this time. Therefore, even accounting for the simplifications made in this analysis, the assumption of thermal equilibrium is clearly valid.

No turbulence model is used for the HYFLEX simulations, since boundary layer transition is expected to occur well aft of the pressure port locations. The maximum Reynolds number encountered in the HYFLEX computational domain at flight conditions corresponds to one-quarter of the smallest Reynolds number at which transition is observed on the space shuttle orbiter.[182]

All flight simulations are performed with the assumption of an isothermal, noncatalytic vehicle surface, set to a typical temperature experienced by entry bodies (1200 K). The surface temperature was found to have only a small effect on the gas pressure around the vehicle. For example, increasing the surface temperature to an unreasonably high 2200 K changed pressure sensor results (on average) by 0.24%. Radiation, which primarily affects the vehicle surface temperature, is thus ignored. The simulations of shock tunnel experiments assume the model to remain at the ambient laboratory temperature (300 K). In both flight and shock tunnel simulations, a nonslip wall boundary condition is used.

Simulations of the HYFLEX in flight are marched until steady-state flow is reached, independent of the time required. Time accuracy, however, is used for the shock tunnel simulations to model flow starting processes. In this capacity, the code was applied to ensure that the test time achieved in the shock tunnel was large enough to produce steady state flow around the model. For some of the experiments discussed in this chapter, simulation of the shock tunnel flow is further improved by using an approximation of the actual nonuniform test flow, rather than assuming that uniform flow impinges on the model.

### 6.1.1 Finite Volume Grids

It was desired to create a computational grid that would allow both efficient and accurate CFD simulations of the HYFLEX. Such a grid needs to have high resolution around the nine nose pressure sensors, and at the same time not require unreasonable computer resources. To this end, two grids were produced.

The first grid design allowed modelling of flow around the entire HYFLEX forebody, and is of similar style to that used in Reference 183. The mesh was created in a manual fashion, by appropriately sectioning the HYFLEX geometry and extending rays orthogonal to the surface to the outer domain boundary. Cutaway views of the complete mesh are shown in Figure 6.3. Unfortunately, this style of grid introduces a group of distorted cells concentrated in rings near the nose. At the nose tip, one face of each hexahedroidal cell collapses, forming the series of wedges visible on the vehicle surface in Figure 6.3(a). Separate rings of the wedge-shaped cells are contained in every layer from the body to the outer mesh boundary, as shown in Figure 6.3(b).

Theoretically these cells are no problem, since Equation 2.17 dictates that cell faces with zero area can only admit zero fluxes. Although this is the case, results from early simulations indicated that the mesh singularity along the vehicle axis was introducing small perturbations into the solution. Pressure, in particular, was adversely affected. Because the singularity does necessarily coincide with the stagnation line, transverse flow

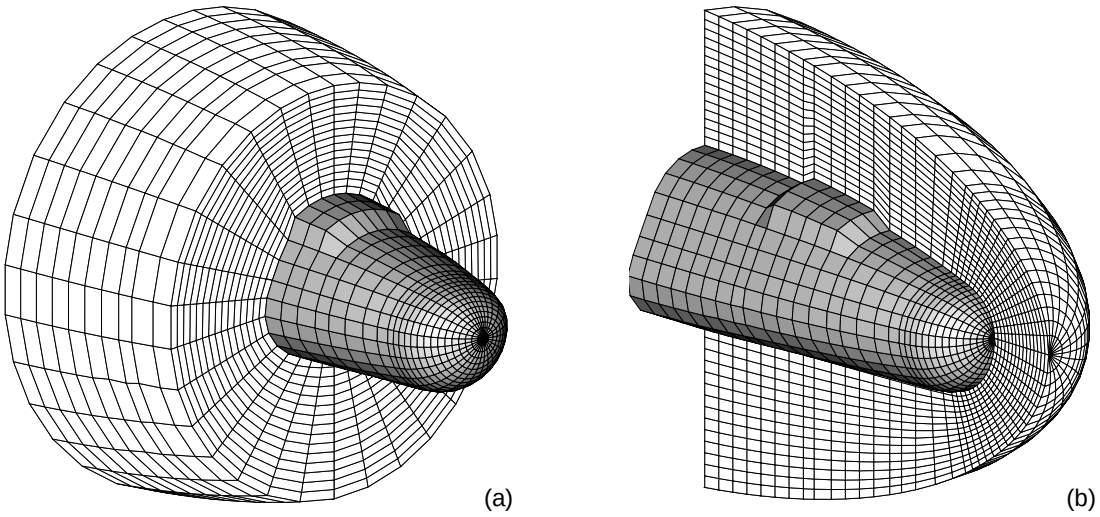



Figure 6.3: Sectional views of the original HYFLEX computational mesh.

can exist through the wedge-shaped cells. It is supposed that the one-dimensional reconstruction, and continual application of flux solvers over such a short flow length, both contribute to noise in this region.

In an attempt to alleviate the noise, each ring of wedge-shaped cells was merged into a multi-sided conglomerate cell. Properties within each conglomerate cell were equalized in a conservative manner at each timestep, with fluxes calculated at each external boundary. A similar idea was developed independently by Ma *et al.*[184] Although this technique reduced noise at the nose tip to an extent, unacceptably large perturbations were still observed to propagate downstream, originating from the singularity.

A second grid, shown in Figure 6.4, is used to overcome this difficulty. The mesh contains no singularity, and covers only the region of the forebody geometry necessary to properly reproduce flow around the pressure sensors. It is ensured that the mesh contains all subsonic parts of the forebody flow, and that the advection Mach numbers on all outflow planes are large enough to prevent the outflow boundary characteristics affecting the flow around the pressure sensors. Also, nine of the grid cells around the bow surface are arranged to lie with centroids positioned at each of the pressure port locations. Cell clustering in the body-normal direction is used to help resolve boundary layer effects in viscous simulations, and in the longitudinal direction to concentrate computational effort around the pressure sensor area. Abrupt changes in cell spacing are accounted for by the reconstruction scheme, as already described. A typical mesh contains 46 cells in the longitudinal direction, 45 cells in the transverse direction, and 15 to 25 cells in the body-normal direction. Because shock fitting is used, accurate solutions can be generated using a small number of cells in the body-normal direction in the shock layer core flow.

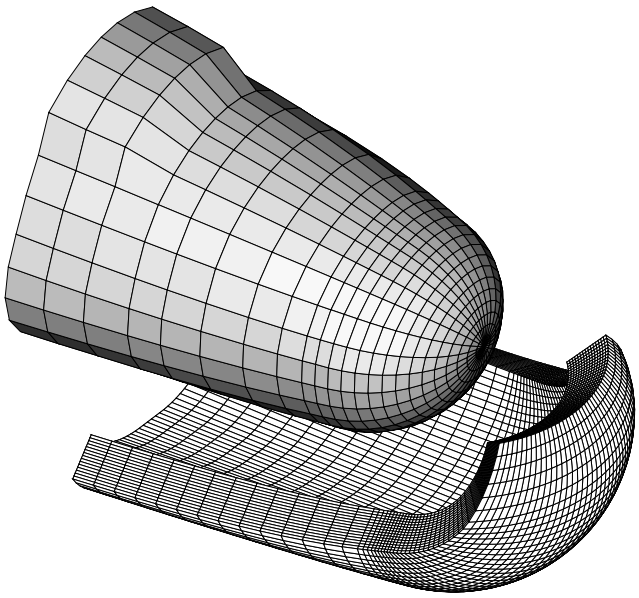


Figure 6.4: The improved computational mesh (for clarity, shown without cell clustering).

Thus, enough cells are usually available to simulate the boundary layer to the accuracy required for surface pressure estimation. The minimum cell width in the boundary layer is of the order of 0.1% of the nose radius. Additionally, the use of shock fitting conveniently allows simulations to be produced at different vehicle orientations and free-stream conditions without having to readjust the outer grid boundary; new shock wave positions are automatically handled by the fitting process. Simulations on the grid in Figure 6.4 require two to three days of processing time to reach steady state flow, on a single MIPS R10000 processor running at 195 MHz.

A grid convergence test was performed to establish the accuracy to which the governing fluid equations are solved by SF3D. Flow through a coarse three-dimensional grid partially wrapped around a sphere, of diameter equal to that of the HYFLEX nose, was used to generate a base solution of vehicle surface pressures. Pressure results were also obtained on grids of double and triple resolution. The cell sizes of the double resolution grid correspond to those on the actual HYFLEX grid. Quadratic Richardson extrapolation was used to find approximations to the exact pressure values over the vehicle surface.[161, 185] Inherent to this kind of extrapolation is the assumption that the CFD code has spatial accuracy of, or less than, the second order. For the double resolution grid, results showed that the average error in computed pressure for the internal finite-volume cells was 0.08% of the post-shock stagnation pressure. The average error for cells located at the flow domain boundary was higher, at 0.65% of the post-shock stagnation pressure. Since all the pressure sensor positions are well inside the edges of the grid, the 0.08% accuracy is representative of the computational results presented in this and the next chapter.

## 6.2 Modified Newtonian Model

Newtonian theory provides a simple, approximate method to predict pressures on the HYFLEX vehicle surface. The theory assumes that the free-stream flow consists of elements of non-interacting fluid, which initially travel directly towards the vehicle. Upon impact with the vehicle surface, it is supposed that the fluid transfers all momentum in the surface-normal direction to the body, while the tangential momentum is retained. With this remaining momentum, the fluid proceeds to flow rectilinearly away from impact site at the tangent angle. The normal momentum flux is thus representative of the surface pressure, according to the theory. The Newtonian argument does not incorporate interactions within the fluid, nor effects such as boundary layer development, shock waves, or chemical nonequilibrium. Also, the theory generally has poorer accuracy at lower flow speeds.[13] For hypervelocity flows the best results can be expected when the shock shape closely follows the body shape, and the standoff distance is small; otherwise excessive flow interactions within the shock layer may invalidate the Newtonian assumptions. Although limited in accuracy, the Newtonian method is often used as a simple design tool, and its predictions form a convenient reference base to compare flight, shock tunnel, and CFD surface pressure results.

By scaling the pressure distribution so that it has the correct post-shock stagnation pressure, the accuracy of the method can be somewhat improved. This is referred to as modified Newtonian theory,[13] and is given by

$$P = (P_t - P_\infty)\cos^2\theta + P_\infty \tag{6.4}$$

where $P$ is the surface pressure, $P_t$ and $P_\infty$ are the post-shock stagnation and free-stream static pressures, and $\theta$ is the angle between the body normal and incident flow. For the results in this chapter, the value of $P_t$ is set to that obtained by processing the incident flow through a normal shock, and then isentropically slowing it until stationary. During this time the fluid is assumed to behave as a perfect gas, with the ratio of specific heats ($\gamma$) remaining at a constant value of 1.4. This process is, in practice, described by the Rayleigh-Pitot formula, which will be discussed in more detail in Chapter 7.

## 6.3 Shock Tunnel Modelling

A model of the HYFLEX forebody was built and tested in the T4 free piston driven shock tunnel by Tuttle and Shimoda.[186] All of the experimental results included in this chapter were produced by Tuttle, and will henceforth not be explicitly referenced.

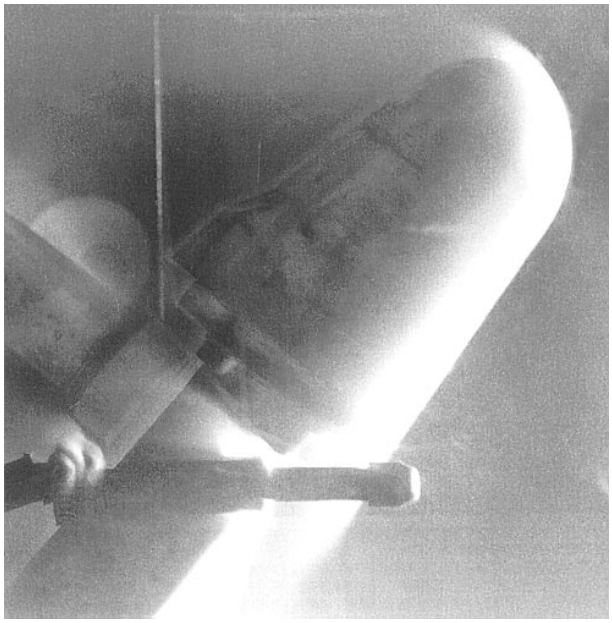


Figure 6.5: The HYFLEX scale model mounted in the T4 shock tunnel. The model Pitot probe is visible in the lower part of the photograph.

The forebody model was designed according to the scaling arguments proposed by Hornung[179] for wind tunnel simulation of flight at hypersonic speeds. To properly model high temperature effects in the experimental facility, it is necessary to reproduce the high shock-layer temperatures encountered during flight; in practice, these temperatures are achieved by matching flight velocity. This ensures that chemical reaction rate coefficients are consistent between both systems.

To correctly simulate the flight chemistry it is also necessary to obtain the same species production rates over the small scale model, so that the chemical composition of the flow during flight is duplicated. Hornung states that for the simple case of a single element gas, the species dissociation rate is proportional to density, while the recombination rate is proportional to the square of density. If recombination can be ignored, then by matching the product of density and length-scale, the composition of the flow can be duplicated (as can the Reynolds number). This is referred to as binary scaling. Binary scaling is only valid when the free-stream gas compositions are matched. The test flow in the shock tunnel experiments, however, was partially dissociated air — the nonequilibrium expansion of flow from the high-enthalpy reservoir caused dissociation of approximately 4% of the $N_2$ and 28% of the $O_2$. The consequences of this are examined later.

Two points on the hypersonic portion of the trajectory of the HYFLEX re-entry glide were selected for the shock tunnel experiments, corresponding to periods of fast flight speed where high-temperature effects most likely occurred. The shock tunnel model was

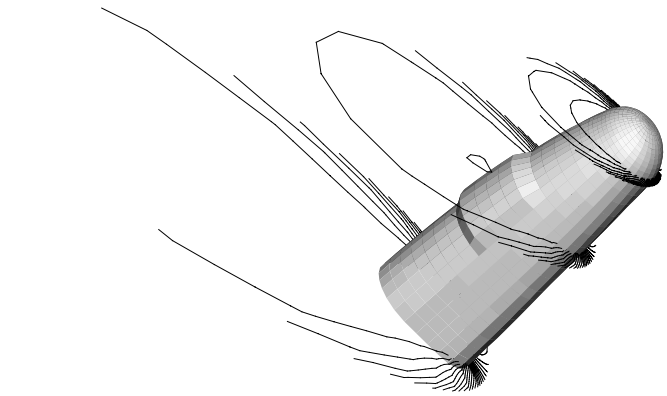
built at one-tenth the size of the flight vehicle and, at this scale, the model fits approximately within the core flow produced by the nozzles in the shock tunnel experiments. Also, the size allows the binary scaling parameter to be matched between flight and experiment, taking into consideration the test gas density achievable in the T4 shock tunnel facility. A Mach 6 nozzle, with exit diameter of 270 mm, was used to produce most of the experimental data presented in this chapter. A result obtained using a Mach 10 nozzle is also presented, however, to examine the effect of Mach number on pressure distribution.

The length of the shock tunnel model is 0.14 m, which corresponds to the first 1.4 m of the flight vehicle. Accordingly, the model ends at a section about half-way along the canopy located on the upper surface of the vehicle. Figure 6.5 is a luminosity photograph of the shock tunnel model on its support in the test section of the T4 shock tunnel. Calculations using SF3D showed that this length ensures that the most downstream sonic point on the model is well upstream from the base and point of attachment to the support. Thus, all characteristics required to correctly reproduce the forebody flow are included in the shock tunnel experiments.

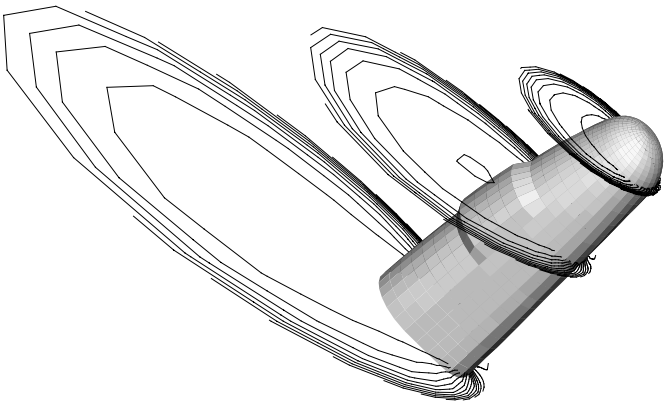
Piezoelectric pressure transducers with 1 $\mu$s rise time measured the surface pressures at locations corresponding to the pressure tapping positions on the flight vehicle. Due to their large size (10 mm diameter, mounted) most of the transducers were located inside the body of the model, and connected to tappings on the model surface by a series of holes. The diameter of each tapping is 2 mm, with location accuracy better than 0.8 mm. Dynamic calibration of the transducers in-situ showed that the average uncertainty in measured pressure was 1%.[186] Mee[187] has examined the uncertainties in test flows produced by the T4 shock tunnel, and found that free-stream static pressure can be determined to an accuracy of about $\pm$7%, density to $\pm$13%, and average flow speed to $\pm$5%.

## 6.4 CFD Simulation of Flight

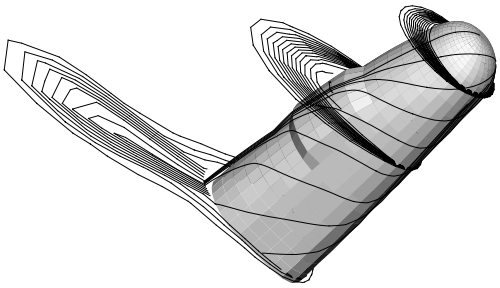
Numerical simulation of the HYFLEX flight was performed over a range of conditions corresponding to between 90 s and 300 s after flight commencement. The flight condition at 120 s shall be considered in detail. At this time, the vehicle was traveling with a 49 deg angle of attack, 1 deg sideslip, and a speed of 3.7 km/s at an altitude of 48 km. Satellite sounding data indicates that, at this altitude, the free-stream static pressure was 97.7 Pa, and the free-stream density was $1.29 \times 10^{-3}$ kg/m$^3$. More details of the vehicle and atmospheric state at this flight time are presented in Table D.1. At these conditions, chemistry effects are vigorous in the shock layer and the vehicle can be considered to be well inside the continuum-flow flight regime.



Isopressure Contours
Minimum: 435 Pa
Maximum: 13100 Pa

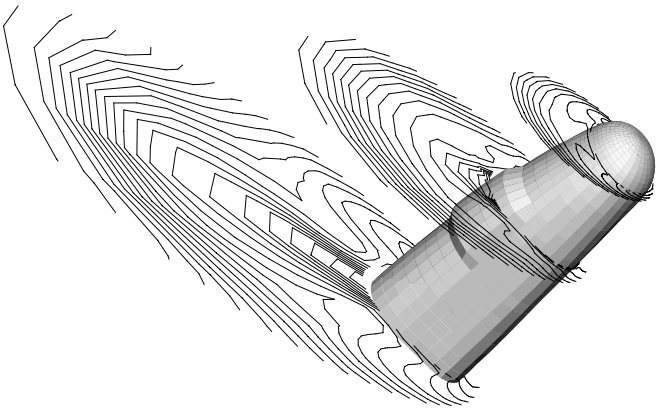


Isopycnic Contours
Minimum: $6.0 \times 10^{-4}$ kg/m$^3$
Maximum: $1.2 \times 10^{-2}$ kg/m$^3$



Monatomic Oxygen Contours
Minimum: 0.000 mass fraction
Maximum: 0.012 mass fraction

Surface Stream Traces



Mach Number Contours
Minimum: 0
Maximum: 9

Figure 6.6: Simulation results for the HYFLEX at 120 s flight time.

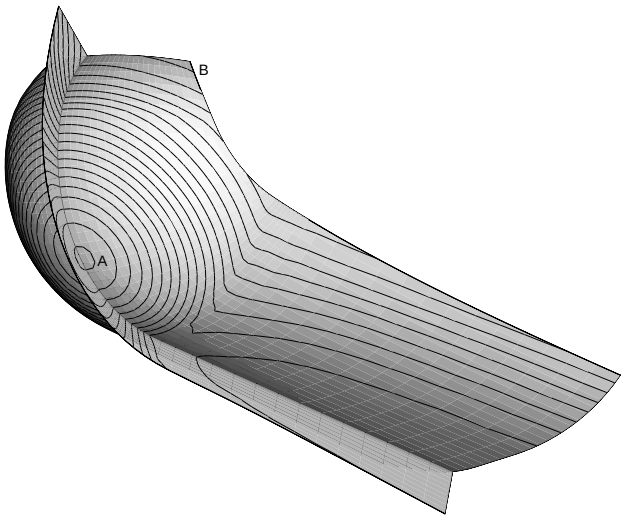



Figure 6.7: Isopressure contours around the HYFLEX surface and in the shock layer. Contour levels are A, 17 kPa; and B, 1 kPa.

The composition of atmospheric air at 48 km altitude was obtained using the MSIS-E-90 model.[188] Species containing elements other than nitrogen and oxygen are present in very small concentrations, typically less than 1% by mass, and are thus ignored. This assumption, however, leads to a slight change of mixture gas constant for the equation of state. The new equation of state constrains the free-stream state variables in a slightly different manner to the equation of state for atmospheric air. To minimize the effect of the approximation on simulated pressure, free-stream static pressure and density are matched between flight and simulation, and free-stream temperature is allowed to depart from the atmospheric value. As indicated in Table D.1 the subsequent error in free-stream temperature is less than 1 K, and negligible.

Results from a viscous, nonequilibrium chemistry simulation of the vehicle at the 120 s flight time are presented in Figures 6.6 and 6.7. Figure 6.6 shows contour plots of results generated using the grid of Figure 6.3, without cell clustering at the body surface. Cell clustering was not used in this case, so that the grid resolution in the shock layer core flow could be maximized, albeit at the expense of boundary layer resolution. The isopressure contours indicate high pressure levels on the windward surface, and a large region of expanded air in the shadow region of the leeward surface, as would be expected. Isopycnic contours also highlight the expansion region, including a low density pocket just downwind of the canopy. Jagged contours are observed where the shock becomes closely aligned with the flow direction, due to the lack of resolution in this area caused by a large expansion of the grid. The coupling of flow physics and chemical reaction is visible in the contours of constant monatomic oxygen concentration. Monatomic oxygen is generated

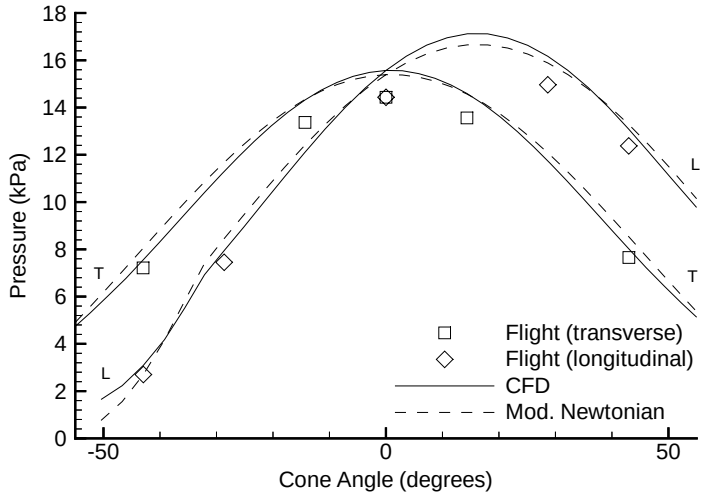



Figure 6.8: Comparison of pressure distributions over the vehicle at 120 s flight time. The longitudinal (L) pressure sensors are drawn in the order (from left to right) ps3, ps2, ps1, ps4, ps5. The transverse (T) pressure sensors are drawn in the order ps9, ps8, ps1, ps6, ps7.

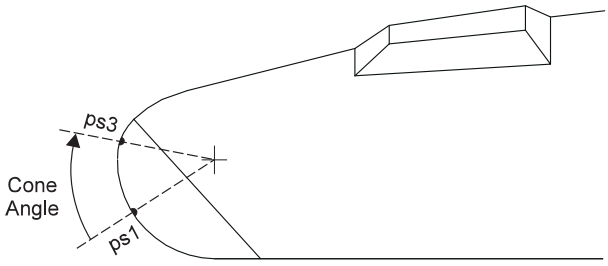



Figure 6.9: Definition of cone angle for pressure sensor 3.

in the high temperature region at the windward surface, and is quenched and transported towards the leeward surface primarily in the circumferential flow defined by the surface streamlines. The Mach number contour plot exhibits a flow disturbance at the centre of the leeward surface, extending from body to shock. The small sideslip angle causes air passing around each side of the vehicle to be processed differently, and the disturbance is thought to result from the recombination of the separated streams. Additionally, agitation caused by the canopy may be partly responsible.

The results in Figure 6.7 were produced using the grid of Figure 6.4, with cell clustering in the boundary layer region. Isopressure contours are shown along the vehicle surface, and across a section of the shock layer. The simulated pressure distribution along the longitudinal pressure sensor array is displayed in Figure 6.8 together with measured flight pressure data, and the results of a modified Newtonian analysis. Additionally, the

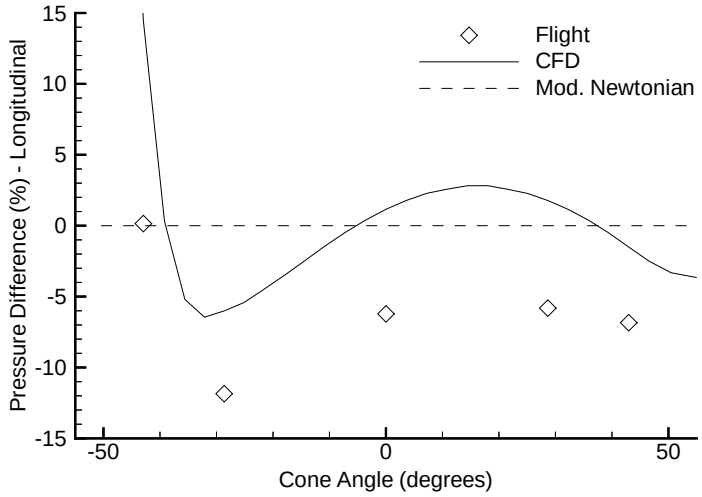



Figure 6.10: Comparison of flight and CFD pressure distributions, relative to modified Newtonian theory, for the longitudinal pressure sensor array.

plot includes the corresponding information for the transverse pressure sensor array. The cone angle referred to on the plot is defined by the angular separation from the ray joining the nose centre with sensor 1. An example of the cone angle definition is shown in Figure 6.9 for pressure sensor 3. At the 120 s flight time, the stagnation point is at a 16 deg cone angle, and lies close to the longitudinal axis of the vehicle. Both the CFD and modified Newtonian techniques are observed to predict pressures that lie reasonably close to, though always exceeding, the flight data.

The CFD and modified Newtonian absolute pressure distributions shown in Figure 6.8 appear to be quite similar. However, the differences between them are more easily seen on a plot showing relative pressure distributions. Shown in Figure 6.10 is the percentage difference between the surface pressures measured in flight, and modified Newtonian theory prediction, for the longitudinal pressure sensor array. Also shown is the percentage difference between CFD surface pressure prediction and modified Newtonian theory prediction. Similarly, Figure 6.11 shows the same information for the transverse pressure sensor array. Both graphs indicate that the CFD results match the trends in flight data more closely than modified Newtonian theory. However, the CFD results appear to be consistently offset from the flight data by about 7%. Possible reasons for this offset include the difficulty in accurately obtaining the free-stream conditions under which the flight occurred (such as density and temperature), as well as the error in measuring the vehicle velocity.[9] For the considered point in flight, these factors alone introduce an average uncertainty in pressure of 2.6%, with a maximum uncertainty of 6.7% at ps3. Al-

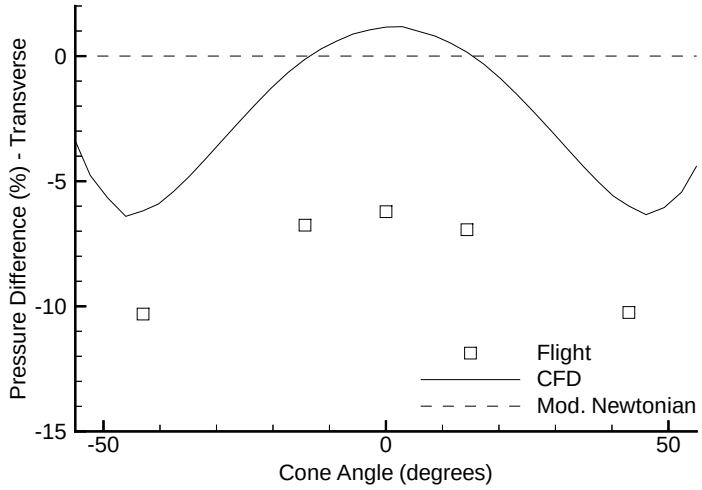



Figure 6.11: Comparison of flight and CFD pressure distributions, relative to modified Newtonian theory, for the transverse pressure sensor array.

though the transducers used to measure the HYFLEX nose pressures were quite accurate (approximately $\pm 22$ Pa), it is possible that flight pressure measurements were adversely affected by pneumatic lag and acoustic noise within the tubes connecting pressure ports to transducers (for example, see Reference 6). Another source of error is the change in cone angle of the pressure ports during flight, caused by thermal deformation of the nosecap.

Flight pressure measurements were also likely to have been affected by atmospheric wind. A 40 m/s wind at a vehicle speed of 3.7 km/s, for example, could perturb pressure measurements by over 2%. For reference, average wind speed for the area of the hypersonic flight experiment is shown in terms of altitude in Figure 6.12. The plot is based on the HWM93 empirical correlation from Reference 189, and does not necessarily reflect the actual wind speed on the day of the flight. However, the correlation does give an idea of typical wind strength encountered at altitude. The combination of wind speed error with the uncertainty sources indicated previously, could easily account for the 7% difference between CFD and flight data.

The relative accuracy of CFD and modified Newtonian predictions over a range of varied flight conditions is best examined by using a quantity that reflects both the trends and magnitudes of the surface pressures. The difference in pressure between ps5 and ps3 (see Figure 6.2) is an indicator that is sensitive to angle of attack, free-stream conditions, and high temperature effects. The error in reproducing this indicator, for a large portion of the flight, is shown in Figure 6.13. The plot spans enough of the trajectory to include both rarefied flow and continuum flow, hypersonic speeds and supersonic speeds, regions of

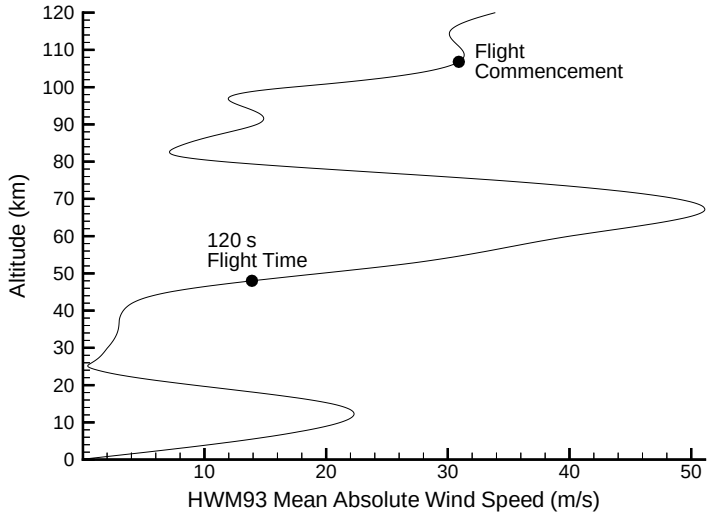



Figure 6.12: Empirical model of wind speed versus altitude.

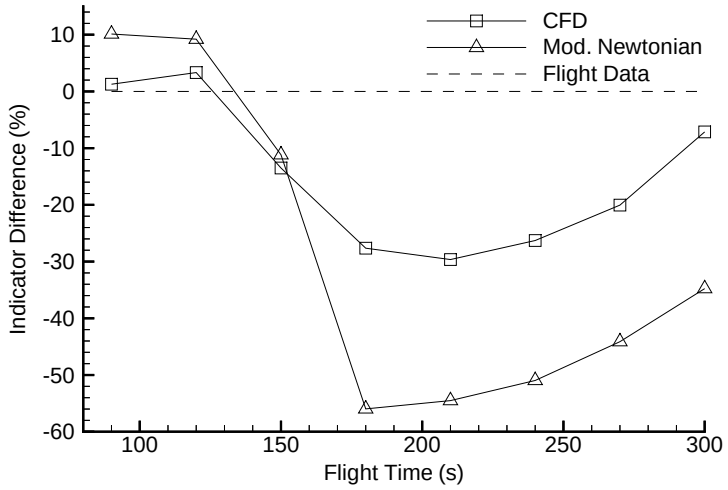



Figure 6.13: Difference in the quantity (ps5-ps3) relative to flight data.

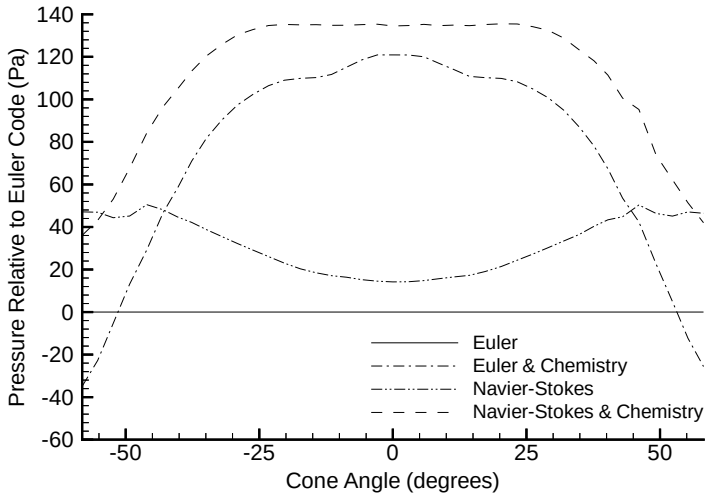



Figure 6.14: Effect of various CFD models on the transverse pressure distribution, for the 120 s flight conditions.

strong high temperature real gas effects, as well as regions of near perfect-gas behaviour (compare with Figure 6.1).

From Figure 6.13, it can be seen that the CFD prediction of the indicator ps5–ps3 is substantially more accurate than the modified Newtonian prediction. The differences observed between the modified Newtonian and CFD pressure values can be attributed to two main causes. First, the CFD method incorporates viscosity and nonequilibrium models. The effects of these models on the transverse pressure distribution can be seen in Figure 6.14. As would be expected, chemistry effects are most prominent near the stagnation region, while viscous effects have most impact on the outer pressure values. The effect of both models combined gives an approximately constant pressure rise over much of the section. While the pressure rise is relatively small, at about 1%, it still comprises a significant portion of the difference from flight data. A second reason for the difference between CFD and modified Newtonian theory is that the CFD method models the flow interactions within the entire shock layer, unlike Newtonian theory. This would appear to be the primary factor causing differences between the two methods.

## 6.5 Shock Tunnel Results and Simulation

The flight condition simulated in the shock tunnel corresponds to 120 s after flight commencement. It was possible for the experimentalist to simultaneously obtain the free-stream velocity and density to within 5% of the values required by binary scaling (the

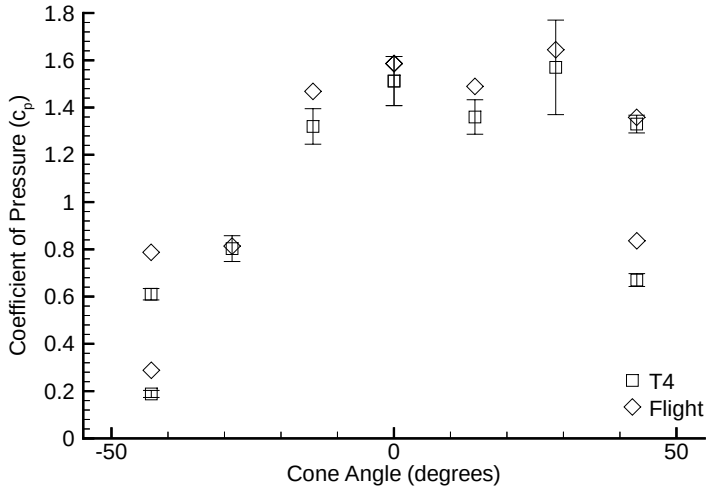



Figure 6.15: Comparison of pressure coefficients derived from the Mach 6 experiment, with those measured in flight. Conditions correspond to the 120 s flight time. Bars denote one standard deviation in the time variation of measured pressures for the T4 data.

velocity was generally matched to within 3% of the flight value).[186] However, for this condition, the flight Mach number of 11.5 was not duplicated in the shock tunnel, since a Mach 6 nozzle was used for the experiment.

A comparison of shock tunnel pressure measurements with flight data is shown in Fig 6.15. The vertical bars on the plot indicate the variation of measured pressure during the experiment test time. The pressure measurements are presented in the form of pressure coefficients, helping to account for not achieving exact free-stream test conditions in the shock tunnel. The pressure coefficient is defined as the difference between measured surface pressure and free-stream static pressure, nondimensionalized by dynamic pressure. Free-stream dynamic pressure is defined as one-half of the product of free-stream density and the square of absolute velocity. Agreement between the two sets of data is reasonable for the pressure sensors located near the stagnation point. Measurements made by the outer pressure sensors compare less favourably with flight data. All pressure measurements on the shock tunnel model are seen to be lower than the corresponding values from the HYFLEX flight vehicle. This same trend and agreement was observed over seven individual shock tunnel experiments at the same nominal condition, simulating the 120 s flight time.[186]

A laser shadowgraph of the subscale model in shock tunnel flow is shown in Figure 6.16. Superimposed onto the figure is the shock position predicted by a CFD simulation of the experiment. Comparison between the shock shapes and standoff distances is

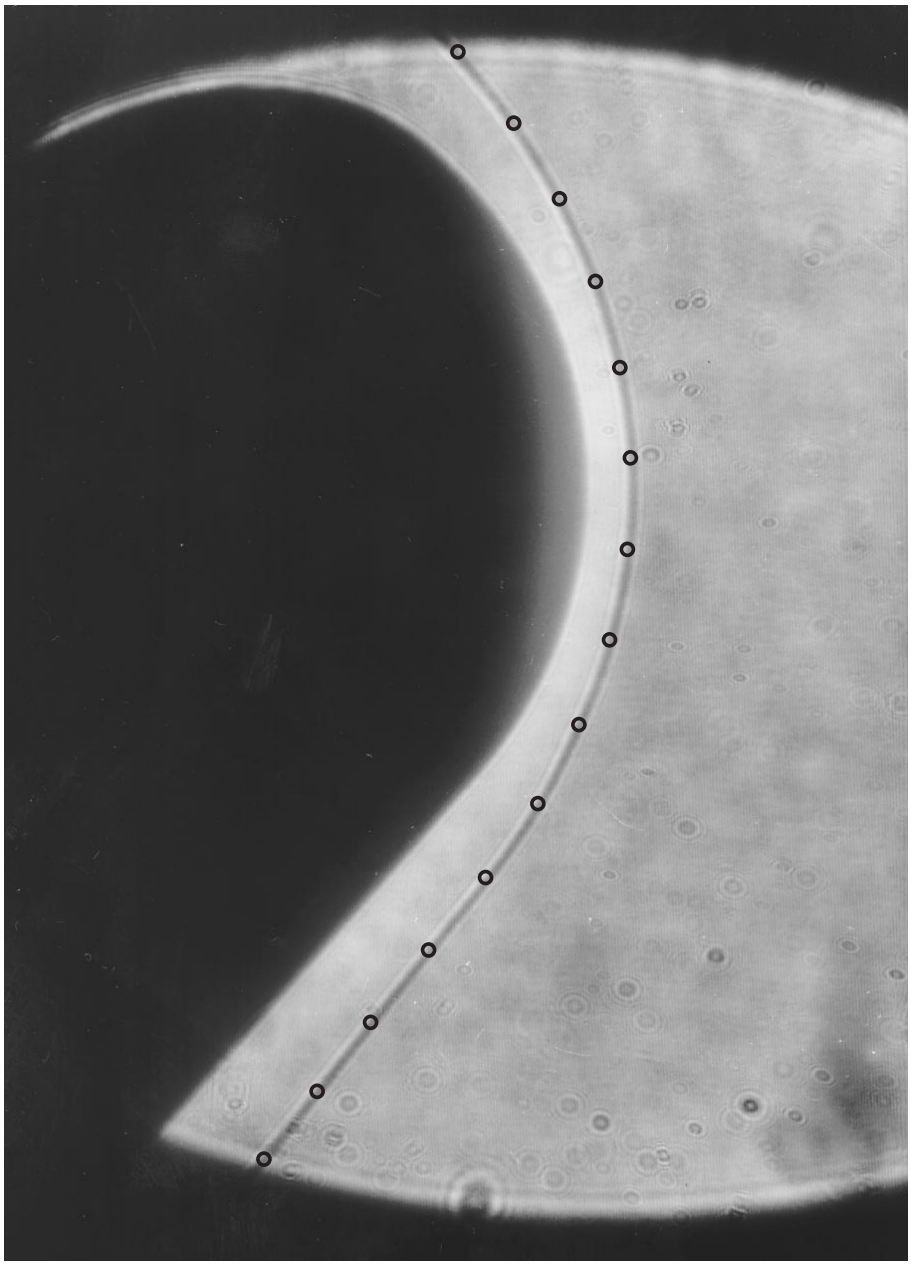


Figure 6.16: Shadowgraph of the bow shock around the HYFLEX scale-model in the shock tunnel at Mach 6. Black circles indicate the calculated CFD shock position.

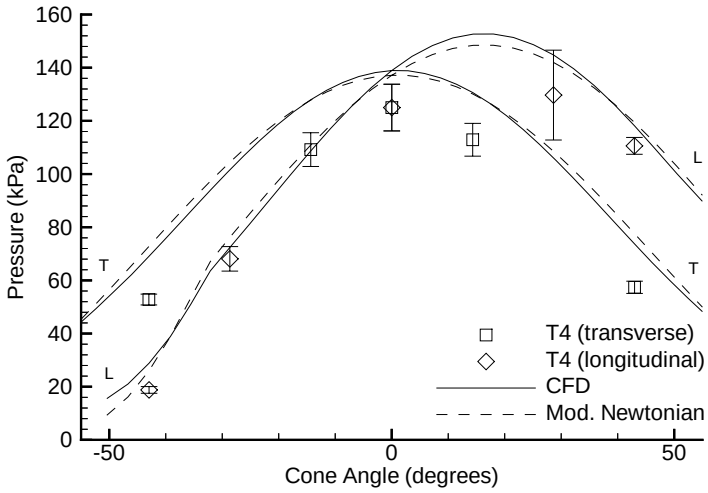



Figure 6.17: Comparison of pressure distributions over the shock tunnel model at Mach 6. Numerical results assume a uniform test flow. Bars denote one standard deviation in the time variation of measured pressures.

excellent. In Section 5.11 it was shown that, for a nitrogen system, standoff distance is reasonably sensitive to the accuracies of the chemistry model and flow condition. In the case of the HYFLEX, though, numerical and shock tunnel experiments revealed shock shape and standoff to be less sensitive to these factors. Thus albeit agreement in shock shape is necessary to validate the numerical and experimental simulations, it is not in itself sufficient for validation of the HYFLEX simulations.

Figure 6.17, in a manner similar to Figure 6.8, compares shock tunnel measurements with the CFD and modified Newtonian predicted pressures along the longitudinal and transverse pressure sensor arrays on the simulated scale model. Again, the variation in measured pressures during the test time is indicated by vertical bars. The numerical results shown in Figure 6.17 assume a uniform distribution of Pitot pressure through the test core. The lack of agreement at the stagnation point may be attributed to the spatial variation in free-stream Pitot pressure, from which the flow conditions are inferred. Differences between the shock tunnel measurements, CFD, and the modified Newtonian estimate, are generally smaller than 10%.

The Newtonian approximation of the flow over the HYFLEX model uses a nominal, constant value of the specific heat ratio to estimate the post-shock stagnation pressure from free-stream flow conditions. Since the temperature behind the shock in the stagnation region is of the order of 5000 K, chemical reactions (such as the oxygen dissociation observed in CFD results) will occur and affect the pressures. The value of $\gamma$ behind the

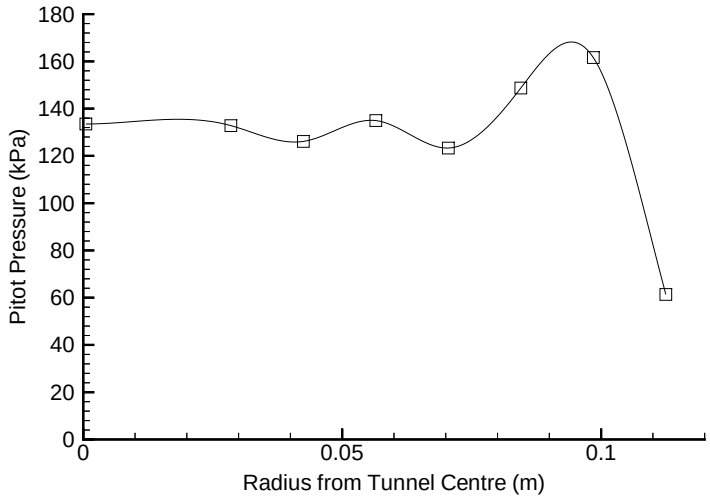



Figure 6.18: Distribution of Pitot pressure across the shock tunnel test section for the Mach 6 nozzle. For reference, the scale-model was mounted in the tunnel such that ps1 was located 80 mm from the tunnel centre line.

shock is thus lower than that in the free stream. When used to calculate the post-shock stagnation pressure for the modified Newtonian model, a reduction in $\gamma$ from 1.4 to 1.3 (for example) causes the stagnation pressure to rise by about 2% at Mach 6. As a consequence, the use of $\gamma = 1.4$ is likely to cause the Newtonian model to estimate slightly lower surface pressures in the stagnation region than CFD. This is generally the case in Figure 6.17. Away from the stagnation region the small discrepancy is reversed, presumably due to the inaccuracy of the Newtonian assumption of a cosine-squared pressure distribution. Better values of $\gamma$ are available from experiment or computation; however these values have intentionally not been used with the Newtonian theory for this study, so that the relative shortcomings and merits of experiment, computation, and Newtonian theory can be individually investigated.

The free-stream conditions used as input to the CFD and modified Newtonian theory are calculated using the Pitot pressure measured during experiment. For the numerical results shown in Figure 6.17 it was assumed that the Pitot pressure across the entire test flow was constant. Pitot pressure surveys across the nozzle exit plane, however, show a variation of approximately 8% through the test core of the Mach 6 nozzle flow (Figure 6.18). This nonuniformity may in part be caused by the use of a contoured nozzle at a condition different to that for which it was designed.

Because most of the nose pressure sensors are located within a subsonic flow region, the spatial variation of flow conditions significantly influences the pressure distribution over the nose of the model and thus the measurement accuracy. The agreement between

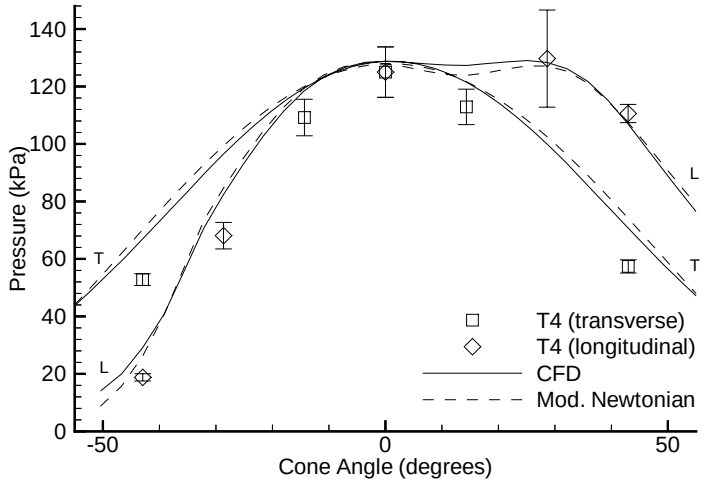



Figure 6.19: Comparison of pressure distributions over the shock tunnel model at Mach 6. Numerical results account for the nonuniform shock tunnel test flow. Bars denote one standard deviation in the time variation of measured pressures.

simulations and experimental measurements was improved by using the nonuniform test flow distribution as the simulation inflow condition. The data presented in Figure 6.19 incorporate this improvement. The test flow distribution was inferred from the Pitot survey results and nozzle stagnation conditions, and assumed to be axisymmetric about the tunnel centre line.

An examination of the chemistry occurring within the shock layer reveals similar qualitative trends in dissociation levels between shock tunnel and flight. Figure 6.20 shows, at flight conditions, the CFD estimated mass fractions of species through the shock layer (from the centre of the first finite-volume cell, to the centre of the last), in a direction normal to the body, near pressure sensor 1. Similarly, Figure 6.21 shows the behaviour of the chemical system under shock tunnel conditions. The most apparent difference between the two sets of data is the concentration of oxygen and nitric oxide just behind the shock. At flight conditions dissociation begins just behind the shock, whereas in the shock tunnel experiment, there are dissociated species in the free-stream resulting from a rapid non-equilibrium expansion through the shock tunnel nozzle. It is also observed that the shock standoff distance has not scaled with model size, with the relative standoff distance on the shock tunnel model being slightly larger. The different standoff distances are consistent with the empirical correlation for shock detachment from a sphere (Equation 5.6), which predicts a larger shock standoff at Mach 6 (the shock tunnel test) than at Mach 11.5 (the flight condition).

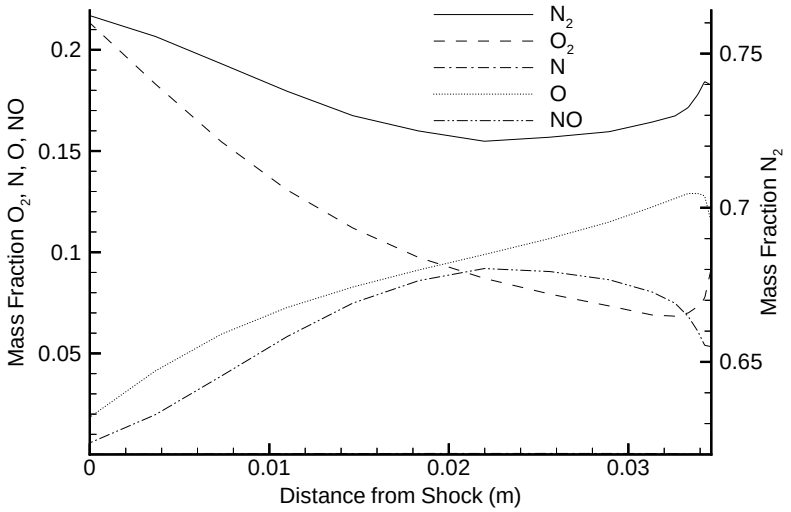



Figure 6.20: Computed species mass fractions in the shock layer, at 120 s flight conditions.

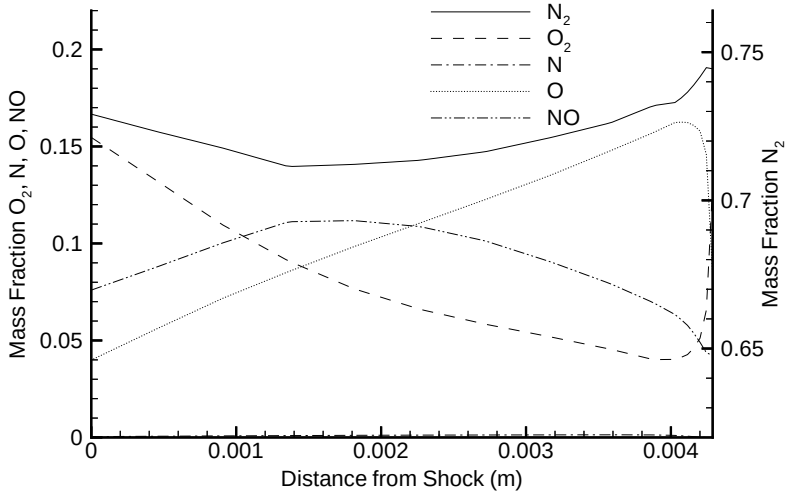



Figure 6.21: Computed species mass fractions in the shock layer, at Mach 6 shock tunnel test conditions.

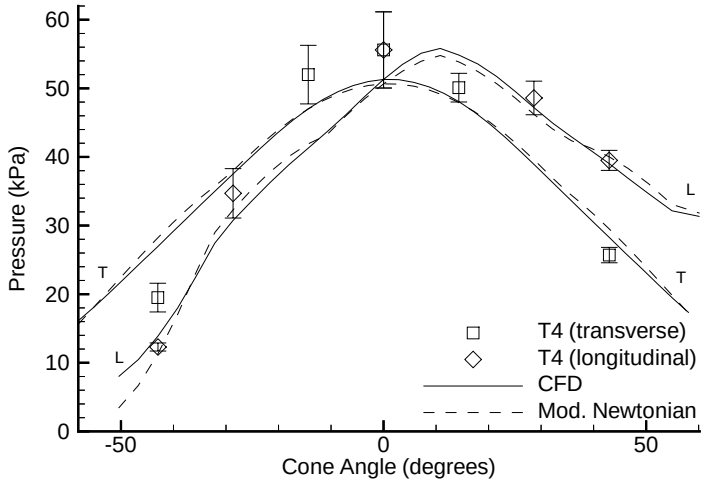



Figure 6.22: Comparison of pressure distributions over the shock tunnel model at Mach 10, simulating the 110 s flight condition. Numerical results account for the nonuniform shock tunnel test flow. Bars denote one standard deviation in the time variation of measured pressures.

The shock tunnel tests simulating the 120 s flight condition were not performed at the flight Mach number, because it was not possible to achieve this Mach number and the correct free-stream conditions at once. Using a Mach 10 nozzle, the required density and velocity could not be obtained without exceeding the maximum rated operating conditions for the shock tunnel.[186] The reduced Mach number that was used in the shock tunnel tests, though, was not expected to produce significant adverse effects on surface pressure simulation. This is suggested by Newtonian theory, which approximates pressure as being Mach number independent. Also, modified Newtonian theory is only weakly Mach number dependent. To verify this near-independence experimentally, the flight conditions at the 110 s flight time were simulated in the shock tunnel. At these conditions, both the flight Mach number and scaled free-stream conditions could be matched in the experimental facility within its rated bounds of operation. The contoured Mach 10 nozzle used for this test produced a pressure distribution less uniform than that generated by the Mach 6 nozzle. Accordingly, the SF3D simulation and modified Newtonian estimates of the Mach 10 experiment account for the nonuniformity. Figure 6.22 shows the comparison between shock tunnel results and numerical techniques. It can be seen that the level of agreement between the shock tunnel measurements and the numerical estimates is comparable to that obtained at Mach 6.

## 6.6 Summary of Simulation Results

In this chapter, the relative merits of numerical simulation, modified Newtonian theory, and shock tunnel experiments were investigated by testing them with flight data obtained from the HYFLEX entry vehicle. As well as examining the usefulness of each simulation method alone, it was found beneficial to use computational fluid dynamics to help explain and alleviate the particular shortcomings of the shock tunnel experiments and modified Newtonian theory. Also, the shock tunnel results were used to help validate the computational fluid dynamics simulations, through the use of a shadowgraph and nose pressure measurements. The key findings were:

(i) Computational fluid dynamics results showed that the effects of nonequilibrium chemistry and viscosity in the flow had a small (about 1%) but significant influence on the pressure distribution at the vehicle surface. The uncertainty in the measurement of model nose surface pressures in the shock tunnel (with a standard deviation of 4%) was too high for real gas effects to be reliably detected.

(ii) Both CFD and modified Newtonian theory predicted the flight-measured pressure data reasonably well, with an average accuracy of 7%. The modified Newtonian theory, however, did not predict trends in the distribution of nose pressures as well as CFD.

(iii) The scale-model shock tunnel results compared acceptably with a CFD shock tunnel simulation. The correct numerical modelling of nonuniformities in the experimental test flow was required to reach this agreement. On average, shock tunnel results were accurate to within 12% of the flight data.

(iv) A laser shadowgraph of the model in the shock tunnel flow showed excellent agreement with the CFD predicted shock shape and standoff distance.

(v) Numerical results showed that binary scaling allowed the shock tunnel tests to reasonably reproduce the shock layer chemistry occurring at flight conditions. It was observed, though, that proper simulation of flight chemistry was limited by the existence of pre-dissociated species in the shock tunnel flow.

(vi) Pressure measurement results suggested that matching the flight Mach number in shock tunnel experiments is not crucial for reproducing flight pressure data, when viscous effects are small.

CHAPTER 7

# Flush Air Data System Calibration

The successful control of a hypersonic vehicle in flight requires knowledge of the vehicle state to sufficient accuracy. Particularly important are the state data that describe the ambient atmosphere, and its interaction with the moving vehicle. These are often called air data, and can be used to evaluate quantities such as the pressure loading and heat loading on the vehicle. Air data may also be used to determine the pressure altitude and attitude of the vehicle, assisting the control system to keep the vehicle trajectory within the desired flight envelope. Examples of air data parameters include angle of attack, angle of sideslip, free-stream dynamic pressure, and free-stream static pressure. For flight experiments, accurate knowledge of these kinds of air data parameters is crucial in the post-flight reconciliation of flight measurements with ground based experiments and computational fluid dynamics (CFD) predictions, as conducted in the previous chapter.

There are several types of instrumentation available for measuring air data, many of them reviewed in Reference 190. Laser velocimeter systems are reported to have good accuracy, but do not work well at high altitude and are unable to produce a full set of air data information.

Alternatively, information from onboard inertial measurement units (IMU) can be used to infer air data. IMU gyroscopes and accelerometers are used to compute estimates of vehicle velocity, altitude, and attitude, with respect to a fixed coordinate system. In conjunction with an aerodynamic model of the vehicle, the IMU data can also be processed to estimate atmospheric conditions. The IMU computed air data are prone to a number of error sources: apart from the effects of IMU instrument drift and inaccuracies

in the vehicle aerodynamic model, the computed air data does not usually account for local wind speed. Supplementing the IMU computed air data with positional and meteorological information derived from satellites can improve its accuracy.

A third method for the determination of air data is a technique based on flow field pressure measurement. For subsonic and supersonic craft, this may be performed with a flow-intrusive boom instrumented with a Pitot tube and mechanical vanes.[191] However, the high energy nature of hypersonic flow makes an intrusive boom impractical, since it would quickly become damaged. Additionally, an intrusive boom disrupts the flow pattern around the vehicle and can lead to flight instabilities for some craft.[192] A remedy to this is the flush air data system concept (FADS), which consists of a number of pressure tappings located flush with the vehicle surface, usually near the nose. The measured distribution of the pressure field around the nose is then used to infer the air data. A minimum of four pressure ports are needed to obtain a complete set of air data parameters, although more ports will increase accuracy. Improvements in accuracy start to diminish when more than nine ports are used.[193]

For hypersonic flows, modified Newtonian theory is able to reasonably predict vehicle surface pressures as a function of air data. Thus by solving an inverse problem, the theory can be used to determine the air data in terms of the known surface pressure measurements. The simplicity and robustness of modified Newtonian theory make it ideal for use in a hypersonic FADS. However, the accuracy of modified Newtonian theory may not be sufficient for all FADS applications.[194]

There are two main approaches in achieving increased accuracy in air data prediction. The first is to apply correction factors to the air data parameters predicted by the FADS, but still retain the underlying modified Newtonian pressure model. This method has been used extensively in the past.[195] The second approach is more fundamental — a better pressure model is created by applying correction factors directly to the Newtonian theory. It is this second technique that will be used to calibrate an entry vehicle FADS in this chapter. The calibration involves finding sets of correction factors applicable to different flight conditions experienced by the vehicle. The correction factors compensate for high temperature gas effects, boundary layer growth, the bow shock wave, and other real flow phenomena not modelled by modified Newtonian theory.

For both approaches the correction factors can be determined from flight experiments, ground based experiments, CFD, or analytical flow theory. Undoubtedly flight-generated correction factors are the most desirable, but they are unavailable for maiden flights, and single-use vehicles such as the HYFLEX. Correction factors may be obtained from ground based experiments, such as wind-tunnel tests,[196] but these may not cover the entire flight envelope nor always provide sufficient accuracy, and are subject to the effects of scale, as

evidenced in Chapter 6. Additionally, a large number of ground based experiments are needed, and performing them can be time-consuming and costly. Using a validated CFD solver to generate a suite of correction factors is an attractive option, due to its wide range of applicability, low cost, and very high precision. The accuracy of a CFD solver, though, must be ascertained before it is relied upon.

In this chapter, we will examine a FADS calibration technique that involves the determination of correction factors using only CFD. The proposed method involves correcting the underlying pressure model exclusively, rather than correcting the predicted air data. This procedure will subsequently be applied to generate air data parameters from FADS pressure data collected by the HYFLEX entry vehicle. The air data parameters predicted by the CFD-calibrated FADS will be compared to IMU results. It will be shown that this solely numerical calibration procedure provides air data that are generally accurate enough for hypersonic vehicle control requirements. Comparison will also be made against air data parameters predicted by an uncalibrated FADS based on modified Newtonian theory.

## 7.1 The HYFLEX FADS

Although the HYFLEX was instrumented with nine nose pressure sensors capable of performing the function of a flush air data system, they were not employed during the flight to determine air data parameters. Instead, the nose pressure measurements were transmitted to ground receivers for post-flight analysis. During flight, an IMU was used to compute air data parameters for control purposes.[197]

The nose pressure tappings, however, are ideally located for air data system use. Pressures measured at the outlying tappings across the beam (ps9 and ps7 in Figure 6.2) are sensitive to vehicle sideslip. Similarly, the tappings ps2–ps5 provide good resolution for determining the vehicle angle of attack. The three pressure tappings near the centre of the nose, ps1, ps6, and ps8, are positioned in the stagnation region for much of the HYFLEX flight, and give a reliable indication of post-shock stagnation pressure, without the influence of upstream disturbance. A genetic algorithm for optimization of FADS pressure sensor locations was developed by Deshpande *et al.*, and applied to the aeroassist flight experiment FADS in Reference 198. It turns out that the optimal pressure tapping layout is similar to the the configuration on the HYFLEX nosecap.

For the scope of this thesis, the HYFLEX FADS will be calibrated for operation over the range of flight conditions encountered during the period 120 s to 300 s after separation. In the low density flow experienced before 120 s, pressure sensor error constitutes a significant proportion of the measured pressures. Pressure measurements made during this

time are thus less useful as test data for the comparison of FADS calibration techniques. For the low supersonic and transonic flight regimes experienced after 300 s, subsonic flow at the nose expands to encompass much of the vehicle. Because low speed air data systems can be calibrated through conventional means, and the HYFLEX computational grid of Figure 6.4 is no longer accurate after 300 s of flight time, we ignore the HYFLEX data recorded in the brief period 300–340 s after separation.

## 7.2 The Air Data Inverse Problem

The surface pressure distribution over the HYFLEX is a function of a large set of variables, including the entire vehicle geometry. However, we may approximate the surface pressures at the nose pressure ports by the following functional relationship:

$$P_i \approxeq F_i(P_{\text{gt}}, P_\infty, \alpha, \beta; \hat{\mathbf{n}}_i). \tag{7.1}$$

Here $i$ represents the pressure port, and $F_i$ is a function estimating the surface pressure $P_i$ at this pressure port location. The vector $\hat{\mathbf{n}}_i$ is the unit normal to the surface at location $i$, measured in a vehicle frame of reference. This vector is easily measured before flight, and can be considered a known quantity. $P_{\text{gt}}$ is the gauge pressure at the stagnation point on the vehicle surface, equal to the stagnation pressure $P_{\text{t}}$ minus the free-stream static pressure $P_\infty$. The vehicle angle of attack is $\alpha$, and the sideslip angle is $\beta$. Most other air data of interest, including free-stream dynamic pressure and Mach number, can be calculated from these four parameters. Note that vehicle roll angle $\phi$, as presented in Table D.1, is not an air data parameter. Roll angle relates the orientation of the vehicle relative to the earth's surface, and does not directly affect surface pressure or aerodynamics.

A procedure for solving for the four unknown air data parameters $(P_{\text{gt}}, P_\infty, \alpha, \beta)$ is now presented. Given that nine pressure observations are available on the HYFLEX, we may construct the following system of nine equations.

$$P_i = F_i(P^*_{\text{gt}}, P^*_\infty, \alpha^*, \beta^*; \hat{\mathbf{n}}_i) + E^*_i, \;\; i = 1, 2, \ldots, 9. \tag{7.2}$$

The quantity $P_i$ is the surface pressure measured by sensor $i$, and $P^*_{\text{gt}}$, $P^*_\infty$, $\alpha^*$ and $\beta^*$ represent the unknown air data parameters which best fit the available set of pressure observations. For nine pressure sensors, there are more equations than unknowns and solution requires the inversion of an overconstrained system. In general, it is not possible to solve the system exactly and residual errors ($E^*_i$) will remain. Thus, some kind of error minimization technique must be employed to invert the system. The method of least squares is commonly used for minimizing error in overdetermined systems,[199] and this technique has also been used before in FADS applications.[193,194]

Least squares can be performed quickly and easily on linear systems. However, for the non-linear Equation Set 7.2, the equations must first be linearized, and an iterative technique applied. Let the $j$th guess for the best state vector of air data parameters be denoted as $\mathbf{q}^j = [P_{\text{gt}}^j \; P_\infty^j \; \alpha^j \; \beta^j]^T$. The guess is then improved iteratively by applying

$$\mathbf{q}^{j+1} = \mathbf{q}^j + \delta\mathbf{q}, \tag{7.3}$$

until $\mathbf{q}^j = \mathbf{q}^*$. In practice this is attained when estimate $j$ is close to estimate $j+1$.

At each iteration step, the incremental term $\delta\mathbf{q}$ in Equation 7.3 must be evaluated. Consider a Taylor series expansion of the $F_i$ about state $j$ and neglect terms higher than first order. We may then write

$$\mathbf{b} = \mathbf{A}\,\delta\mathbf{q}, \tag{7.4}$$

with

$$\mathbf{A} = \begin{bmatrix} \frac{1}{\sigma_1}\frac{\partial F_1}{\partial P_{\text{gt}}} & \frac{1}{\sigma_1}\frac{\partial F_1}{\partial P_\infty} & \frac{1}{\sigma_1}\frac{\partial F_1}{\partial \alpha} & \frac{1}{\sigma_1}\frac{\partial F_1}{\partial \beta} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{\sigma_i}\frac{\partial F_i}{\partial P_{\text{gt}}} & \frac{1}{\sigma_i}\frac{\partial F_i}{\partial P_\infty} & \frac{1}{\sigma_i}\frac{\partial F_i}{\partial \alpha} & \frac{1}{\sigma_i}\frac{\partial F_i}{\partial \beta} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{\sigma_9}\frac{\partial F_9}{\partial P_{\text{gt}}} & \frac{1}{\sigma_9}\frac{\partial F_9}{\partial P_\infty} & \frac{1}{\sigma_9}\frac{\partial F_9}{\partial \alpha} & \frac{1}{\sigma_9}\frac{\partial F_9}{\partial \beta} \end{bmatrix}_{\mathbf{q}^j} \tag{7.5}$$

where, as indicated, the partial derivatives are evaluated at state $\mathbf{q}^j$. Also,

$$\mathbf{b} = \begin{bmatrix} \frac{E_1^j}{\sigma_1} & \cdots & \frac{E_i^j}{\sigma_i} & \cdots & \frac{E_9^j}{\sigma_9} \end{bmatrix}^T. \tag{7.6}$$

In Equations 7.5 and 7.6, $\mathbf{A}$ and $\mathbf{b}$ are referred to as the design matrix and residual vector respectively. The weighting terms $\sigma_i$ represent the standard deviation in pressure measurement uncertainty for sensor $i$. For this study, it is assumed that all pressure sensors have equal uncertainty. To find $\delta\mathbf{q}$, we multiply both sides of Equation 7.4 by $\mathbf{A}^{-1}$:

$$\delta\mathbf{q} = \mathbf{A}^{-1}\,\mathbf{b}. \tag{7.7}$$

Since $\mathbf{A}$ is a non-square matrix, its inverse is undefined. The matrix $\mathbf{A}^{-1}$ is thus referred to as the pseudoinverse of $\mathbf{A}$. To solve the least squares problem for $\delta\mathbf{q}$, we now need the pseudoinverse of the design matrix that minimizes the 2-norm of the residual vector.

There are several techniques available to accomplish this task. Singular value decomposition (SVD) is one of the most robust, and can be applied to decompose $\mathbf{A}$ into the product of a series of component matrices which are easily inverted. The technique is

described in many numerical analysis texts[199] and will not be repeated here; in practice, the SVD code of Reference 200 was used to perform decomposition. Although SVD is not the fastest way to solve the least squares problem, it provides useful information about the conditioning of the design matrix and the relative importance of solution components.

The partial derivatives in Equation 7.5 are easily found if $F_i$ is a simple analytical function. For more complex functions, the derivatives are evaluated numerically. The numerical approximation to the partial derivative of $F_i$ with respect to any air data parameter $x$ is

$$\left.\frac{\partial F_i}{\partial x}\right|_j \approxeq \frac{F_i(x^j+\Delta)-F_i(x^j)}{\Delta}, \tag{7.8}$$

where $\Delta$ is a small number, such that $\Delta/x^j$ is a few orders of magnitude greater than machine precision (for $x^j \neq 0$).

If the measured pressure data contain spikes or irregularities, or a very poor initial guess $\mathbf{q}^1$ is made, then the iterations may not converge. Non-convergence is usually manifested in the guesses for $\mathbf{q}$ oscillating about (but not approaching) $\mathbf{q}^*$. If convergence is not reached within a reasonable number of iterations, a semi-implicit technique is employed to attempt to regain stability. This technique involves calculating all the partial derivatives in $\mathbf{A}$ with Equation 7.8, while setting $\Delta$ to the corresponding value of the air data parameter in the most recently computed $\delta\mathbf{q}$. Equation 7.7 is then used to find a better $\delta\mathbf{q}$. New derivatives for $\mathbf{A}$ are recalculated using the new $\delta\mathbf{q}$ as values of $\Delta$ in Equation 7.8, and the procedure repeated several times. Once the optimum value of $\delta\mathbf{q}$ is finally found, it is substituted into Equation 7.3 to update the state vector estimate. The semi-implicit technique then remains in use until convergence is obtained.

Once the best air data state vector $\mathbf{q}^*$ is determined, the known air data parameters are used to compute other air data of interest. The free-stream dynamic pressure, $q_\infty$, is determined by solving the Rayleigh-Pitot equation. This equation describes the process of a thermally perfect gas passing through a normal shock, and then isentropically slowing until stationary. It relates the free-stream dynamic and static pressures, to the post-shock gauge stagnation pressure by

$$1+\frac{P_{\text{gt}}}{P_\infty} = \left[\frac{(\gamma+1)q_\infty/P_\infty}{\gamma}\right]^{\left(\frac{\gamma}{\gamma-1}\right)} \left[\frac{\gamma+1}{4(q_\infty/P_\infty)-(\gamma-1)}\right]^{\left(\frac{1}{\gamma-1}\right)} \tag{7.9}$$

where $\gamma$ is the ratio of specific heats. Real air, when processed through a strong shock and stagnated, does not keep constant $\gamma$ and is not accurately described by the Rayleigh-Pitot equation. Allowances for this are described in a later section, when CFD-determined correction factors are used to remove this error source.

Mach number, if required, is easily calculated from the estimates of free-stream static and dynamic pressure:

$$M_\infty = \sqrt{\frac{2q_\infty}{\gamma P_\infty}}. \tag{7.10}$$

Recently, some alternative techniques for estimating air data have been proposed. The method of triples, where several sets of three pressure measurements are individually and analytically inverted to obtain air data estimates, has some inherent stability advantages over the iterative method presented here.[2] The technique, however, still requires a function relating nose pressure to air data parameters. After roughly estimating air data parameters using pressure triples, least squares iteration can be applied to obtain final, accurate FADS predictions. Another option is to train an artificial neural network with large sets of information that relate air data parameters to nose pressures.[201–204] The information pool can be obtained from flight tests, computational results, or ground based experimental data. Unfortunately, the reliability and accuracy of neural networks is difficult to quantify, and massive amounts of data are generally required for the network to accurately learn the relationships between air data parameters and surface pressures. Also, such neural networks have no direct fluid dynamical basis, and generally do not extrapolate accurately.

Ultimately, though, the technique used to perform the ADS inversion is not of major importance to this thesis. The methods of least squares, pressure triples, and neural networks all require knowledge of the relationship between air data and surface pressure. The primary aim of this chapter is to test the use of numerical simulation in calibrating a FADS, independent of the inversion technique itself.

## 7.3 Surface Pressure Models

In the described air data system algorithm, the most critical factor affecting accuracy is the surface pressure model. If the assumed relationship between air data parameters and surface pressure is not correct, then the air data estimates will be poor.

As observed in Chapter 6, modified Newtonian theory is a simple and reasonably accurate surface pressure model. The theory may be rewritten in the form

$$F_i = P_{\text{gt}} \cos^2\theta + P_\infty, \quad \text{where} \tag{7.11}$$

$$\cos\theta = \begin{bmatrix} -\cos\alpha\cos\beta \\ \sin\beta \\ -\sin\alpha\cos\beta \end{bmatrix} \cdot \hat{\mathbf{n}}_i. \tag{7.12}$$

As before, the variable $\theta$ denotes the angle between the local (outward facing) surface normal, and the free-stream flow.

Numerical simulations and shock tunnel results showed that the actual pressure distribution around the HYFLEX nose does not follow the Newtonian cosine-squared law. We introduce a correction factor function $f$ into Equation 7.11 to further improve its accuracy.

$$F_i = P_{\text{gt}}\left[\cos^2\theta + f(\theta, R, \alpha, i)\right] + P_\infty, \quad \text{where} \tag{7.13}$$

$$R = \frac{P_\infty}{P_{\text{gt}} + P_\infty}. \tag{7.14}$$

The correction function is introduced as an additive (rather than multiplicative) term, so that the accuracy of the formulation is not degraded when $\cos\theta$ approaches zero. The functional dependence of $f$ on $\theta$ reflects the fact that the correction factor varies with flow incidence angle. Likewise, nonuniformity in some of the HYFLEX nose geometry necessitates the dependence of $f$ on the pressure port identification number *i* and angle of attack $\alpha$. The variable *R* is used to give the correction function an indication of the severity of high temperature gas effects, and nature of the flow field. The variable is conveniently expressed in terms of already available air data parameters, through Equation 7.14. In the hypersonic limit, it can be deduced that *R* roughly scales with $M_\infty^{-2}$. Since stagnation enthalpy is approximately proportional to $M_\infty^2$, it can be concluded that *R* is roughly inversely proportional to the total temperature in the shock layer.

The value of the pressure correction function $f$ for a particular set of arguments $(\theta, R, \alpha, i)$, can be found by evaluating the difference between the modified Newtonian pressure prediction and the pressure determined from another, more accurate, source. By repeating this procedure over a large argument domain, the complete correction function will eventually be described. In this work, the correction function will be developed by comparing the modified Newtonian theory with a number of CFD simulations. It is emphasized that no ground based experiment data or flight pressure data are used to augment the pressure correction model. Accordingly, for the remainder of the thesis Equation 7.13 will be referred to as a CFD pressure model.

To determine the form of $f$ over a large enough domain, SF3D was used to compute the HYFLEX nose pressures for a range of different flight conditions that occurred on the actual trajectory. Figure 6.1 shows the seven points on the flight trajectory that were simulated. Thus, we know that seven flight conditions are guaranteed to be in the domain of $f$. The question then arises: Is it valid for pressure data, recorded during the flight, to be used as input to test a FADS algorithm which is based on a CFD pressure model that was calibrated for that same flight's trajectory? It is an important question, since fair evaluation of the CFD pressure model in the results section depends on it. The validity of the on-trajectory calibration can be justified with two arguments. First, the

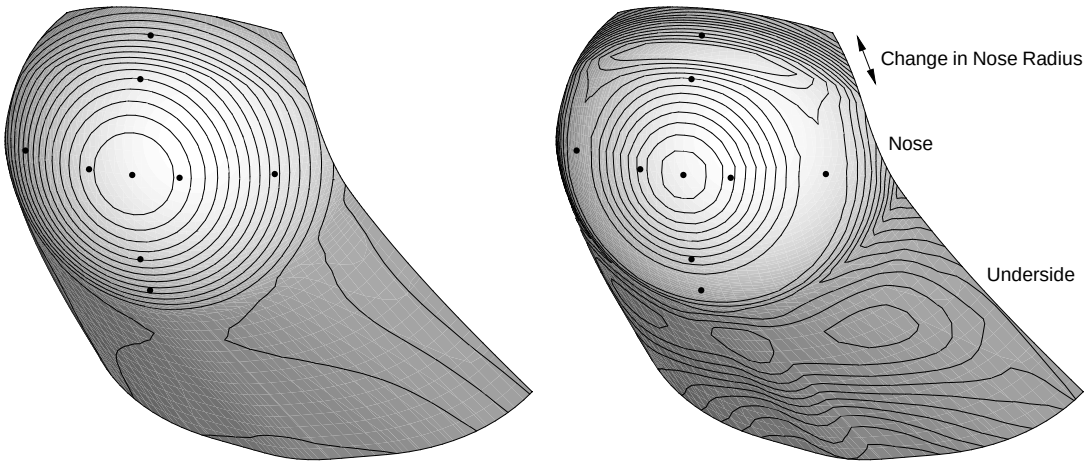



Figure 7.1: Contour plots of CFD results for pressure (left) and pressure correction factor (right) over the HYFLEX nose and underside. Filled circles mark the locations of pressure ports.

actual flight trajectory was so close to the planned trajectory[178] that no real *a posteriori* accuracy advantage is gained by calibrating on flight trajectory conditions. Second, it will later be demonstrated that the CFD pressure correction function produces accurate results for sample flight conditions that are significantly off-trajectory. The calibration was performed on the flight trajectory simply to allow convenient comparison of CFD and flight measured pressure results without the need for further simulations.

## 7.4 Calibration Simulations

The SF3D code was again used to numerically simulate flow over the HYFLEX vehicle forebody. Although the code is time accurate, it is assumed (for simplicity) that the flow around the HYFLEX is instantaneously steady at all points in flight. The assumption is reasonable, since only small changes in flow conditions occur over the time required for full development of the flow field. For example, at 120 s after flight commencement, the free-stream velocity changes by only 0.1 m/s (0.003%) in the equivalent of four body lengths of flow time. Importantly, the assumption allows the generation of CFD flow solutions at any point on the trajectory, independent of previous flight conditions.

An example calibration simulation is presented in Figure 7.1. Contours of pressure and pressure correction factor $f$ are drawn on the vehicle surface. Pressure port locations are indicated by dots. For these results, the vehicle was simulated as travelling at 3.7 km/s at an altitude of 48 km, with a 32 deg angle of attack and no sideslip. Examination of the figure reveals that the correction factor contours are axisymmetric (about the stagnation

point) over much of the nose. The axisymmetry is to be expected, since the nosecap is mostly spherical. The upper pressure port (ps3), though, is located on a region of the nose where the radius of curvature decreases. This decrease in nose radius causes an increased rate of flow expansion, and the CFD result is observed to quickly depart from the modified Newtonian approximation. Pressure port ps2 is also affected by the change in nose radius, but to a lesser extent.

The axisymmetry of the correction factor contours about the stagnation point may be exploited in several ways. First, the axisymmetry indicates that $f$ should be dependent on only one spatial variable, describing position on the nose relative to the centre of symmetry. On the spherical parts of the nose, the flow incidence angle is also a measure of angular separation from the stagnation point, and is an ideal choice for this variable. Second, the axisymmetry allows $f$ to be built using fewer and simpler simulations than would be required for an asymmetric body. This is because the flow around a sphere (of radius equal to that of the HYFLEX nose) is analogous to the flow around the HYFLEX nose itself. In terms of computational requirements, the simulation of a sphere is considerably faster and easier than the simulation of the HYFLEX forebody. Results from CFD tests showed that the correction factor distribution on the sphere matched the distribution on the spherical parts of the HYFLEX nosecap. Thus, with the exception of ps2 and ps3, just one sphere flow field is required to determine the complete form of $f$ for a given flight condition. In total, the flow over a sphere was simulated at seven different $R$ values, corresponding to seven flight conditions in half-minute increments from 120 s to 300 s. These simulations were used to define the core of function $f$.

A different strategy was used to modify $f$, to make allowance for the two pressure ports affected by the nonuniformity in nosecap curvature. Simulations of the HYFLEX were performed at various angles of attack and flight conditions, and the correction factors at pressure port locations ps2 and ps3 were recorded. The correction function was then amended by evaluating the difference between these values, and the correction factors at equivalent incidence angles on a sphere. Overall, only about 20 simulations were used to construct $f$. By comparison, recent calibrations of neural network air data systems with ground based experimental results have required well over an order of magnitude more data.[202] Still other calibrations used as many as five thousand frames of experimental results.[204] The 20 CFD simulations took the equivalent of 60 days of computation on a single MIPS R10000 processor running at 195 MHz. In practice, multiple processors were used to reduce the actual time required to perform all the simulations.

The SF3D computed correction factors are fairly small, with magnitudes typically not exceeding 0.04. As a consequence, the correction factors are especially prone to error introduced by the discretization of the flow domain. Thus for the correction factors to be

reliable, the CFD solver must be significantly more accurate than 4%. The grid convergence analysis in Chapter 6 showed that the average error in computed pressure (over the parts of the flow domain containing the pressure ports) was equal to 0.08% of the post-shock stagnation pressure. This accuracy is good enough to render the correction factors valid. Also, the small magnitude of the correction factors renders the possibility of shock tunnel calibration of air data systems an unfeasible option. In Section 6.4, it was observed that boundary layer development and nonequilibrium chemical reactions typically have a 1% influence on nose pressure. These physical processes are thus an important part of the CFD model, since they account for roughly one-quarter of the correction factor.

## 7.5 The CFD-Calibrated Surface Pressure Model

It is convenient to express the pressure correction function in terms of several component functions:

$$f = f_{\mathrm{a}}(R)\; p(\lambda(R),\theta) + f_{\mathrm{t}}(R) - g(R,\alpha,i). \tag{7.15}$$

The core of Equation 7.15 is the polynomial $p$. It is given by

$$p = \sum_{k=1}^{4} c_k (\lambda(R)\,\theta)^{2k}, \qquad |\theta| \leqslant 1.2 \text{ rad}, \tag{7.16}$$

with the constants $c_1 = -4.7$, $c_2 = 6.82$, $c_3 = -3.02$, and $c_4 = 0.45$ derived from CFD results. The polynomial describes the variation of the correction factor with flow incidence angle. Since the flow around the spherical part of the HYFLEX nose is axisymmetric, $p$ is thus necessarily symmetric about the stagnation point ($\theta = 0$). Hence $p$ contains only even powers of $\theta$.

A striking feature of Equation 7.16 is that the curve fit coefficients $c_k$ are constants. That is to say, the underlying shape of the correction function does not depend on flight conditions. Flight conditions are only needed to dictate the scaling and ordinate-position of the basic polynomial $p$. The sensitivity to flight conditions is introduced with a single variable, $R$ (defined in Equation 7.14), which is a measure of proximity to the hypersonic limit and the strength of high temperature effects. This variable is used as an argument to the function $\lambda(R)$, which is used to scale $p$ with respect to $\theta$. In a similar manner, $f_{\mathrm{a}}(R)$ is used to set the amplitude of the polynomial $p$, according to the flight conditions. Thus $f_{\mathrm{a}}(R)$ can be thought of as a measure of the variation in $f$ over the body. Table 7.1 and Figure 7.2 show the values of these functions at various $R$, as derived from CFD simulations.

The function $f_{\mathrm{t}}(R)$ is used provide an ordinate offset, so that the correct value of $f$ is obtained at the stagnation point ($\theta = 0$). A first examination of the CFD pressure model

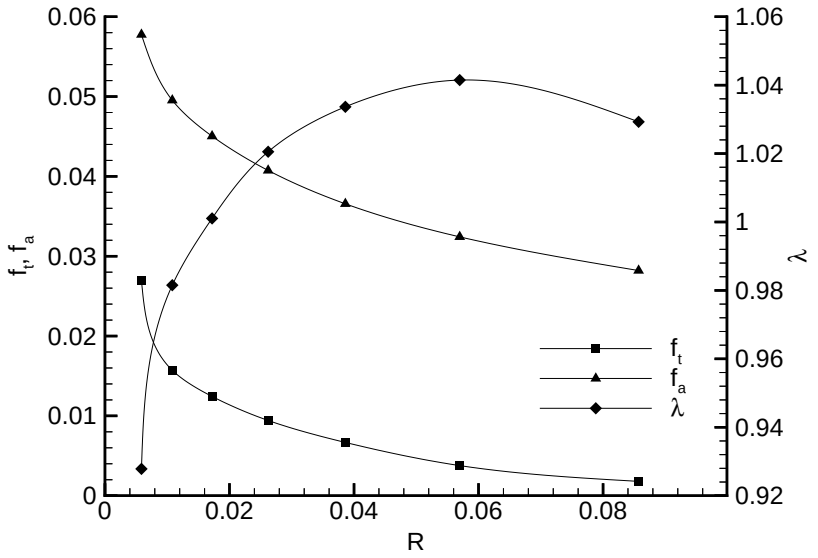



Figure 7.2: Plot of various functions, against $R$.

(Equation 7.13) indicates that $f_t$ should be set to zero for all values of $R$. However, due to the inaccuracy of the Rayleigh-Pitot equation (Equation 7.9), this is not the case. The function $f_t(R)$ is used to cause the air data inversion algorithm to compute an adjusted value of $P_{gt}$, which, when substituted into the Rayleigh-Pitot equation, will produce the correct value of the air data parameter $q_\infty$. In a physical sense, $f_t(R)$ may be thought of as the relative error in calculating gauge stagnation pressure with the Rayleigh-Pitot equation, as compared with a CFD calculation. Table 7.1 and Figure 7.2 show the computed values of $f_t(R)$ for various $R$.

The last term in Equation 7.15 is used to account for the nonuniformity in the HYFLEX nose radius. It takes the form:

$$g = \begin{cases} \dfrac{g(R,\alpha_\circ,i)}{g(R_\circ,\alpha_\circ,i)}\, g(R_\circ,\alpha,i) & \text{for } i = 2,3,\ \beta \approx 0 \\ 0 & \text{otherwise.} \end{cases} \tag{7.17}$$

For the pressure ports unaffected by the change in nose radius, the term vanishes. For pressure ports 2 and 3 though, the value of $g(R,\alpha,i)$ is determined using a system of two look up tables. Table 7.1 shows how $g$ varies with $R$, at a constant value of $\alpha$ ($\alpha_\circ = 32.7$ deg). In a like manner, Table 7.2 shows how $g$ varies with angle of attack $\alpha$, at a constant value of $R$ ($R_\circ = 0.0108$). Using spline fits, the two tables are interpolated and the results combined according to Equation 7.17, to find $g(R,\alpha,i)$. This procedure relies on the assumption that the behaviour of $g$, with respect to $\alpha$, scales with $R$. By making the

Table 7.1: Values of various functions, with respect to $R$.

| $R$ | $f_t$ | $f_a$ | $\lambda$ | $\frac{g(R,\alpha_\circ,2)}{g(R_\circ,\alpha_\circ,2)}$ | $\frac{g(R,\alpha_\circ,3)}{g(R_\circ,\alpha_\circ,3)}$ |
|---|---|---|---|---|---|
| 0.0059 | 0.0270 | 0.0578 | 0.9278 | 0.7993 | 0.8295 |
| 0.0108 | 0.0157 | 0.0495 | 0.9815 | 1.0000 | 1.0000 |
| 0.0172 | 0.0123 | 0.0450 | 1.0010 | 1.2172 | 1.4454 |
| 0.0262 | 0.0094 | 0.0407 | 1.0205 | 1.4695 | 2.0202 |
| 0.0386 | 0.0067 | 0.0366 | 1.0337 | 1.7992 | 2.5477 |
| 0.0570 | 0.0038 | 0.0324 | 1.0415 | 2.3999 | 3.2533 |
| 0.0857 | 0.0018 | 0.0282 | 1.0292 | 3.2730 | 4.0409 |

Table 7.2: Values of $g$, with respect to angle of attack.

| $\alpha$ | 20° | 30° | 32.7° | 40° | 50° |
|---|---|---|---|---|---|
| $g(R_\circ,\alpha,2)\times10^3$ | 11.95 | 6.82 | 4.73 | 1.33 | 1.06 |
| $g(R_\circ,\alpha,3)\times10^3$ | 16.21 | 8.75 | 4.17 | -2.70 | -5.04 |

assumption, it is possible to calibrate $g$ without needing CFD results at all angles of attack at all values of $R$. Computational effort is thus significantly reduced. Equation 7.17 also requires that the vehicle angle of sideslip is small, since Table 7.2 was calculated with $\beta = 0$. This requirement is satisfied for the hypersonic part of the HYFLEX trajectory, since $\beta$ never exceeds 1 deg.

Examples of the pressure correction function at two different flight conditions are now presented. In both examples, $g$ is ignored. The first set of conditions corresponds to 120 s after flight commencement, where the vehicle velocity is 3.7 km/s, and the free-stream static pressure is 97.7 Pa. For these conditions, Figure 7.3 shows the pressure correction factors calculated by CFD at discrete values of $\theta$. Since these CFD results were already used in the calibration of $f$, we consequently see very good agreement with Equation 7.15. A second set of conditions are now chosen, with a vehicle velocity of 2.5 km/s and free-stream static pressure of 97.7 Pa. These conditions are significantly off the HYFLEX flight trajectory, and the pressure correction function has not been calibrated at this condition. Figure 7.4 shows that very good agreement between CFD results and the pressure correction function is still achieved at off-trajectory conditions.

A brief verification of the complete FADS algorithm, including the new surface pressure model, is now conducted. Providing the surface pressure model is properly constructed, the inversion algorithm should select air data parameters that match the flight data with CFD pressure predictions as close as is possible. Air data predictions for angle of attack, angle of sideslip, and dynamic pressure made by the CFD-calibrated FADS at

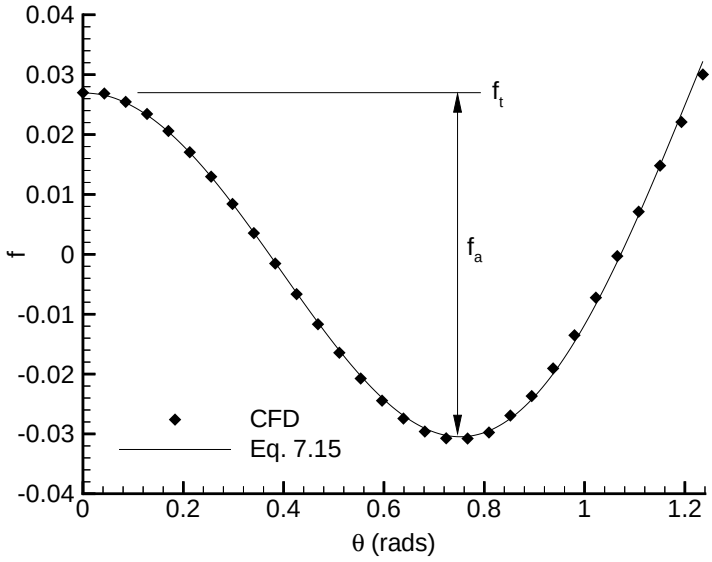



Figure 7.3: Plot of pressure correction factors for $R = 0.0059$.

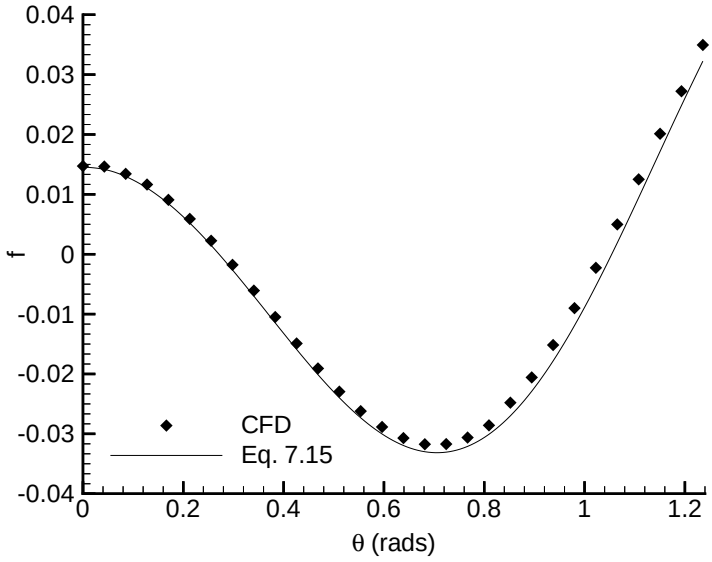



Figure 7.4: Plot of pressure correction factors for $R = 0.0130$.

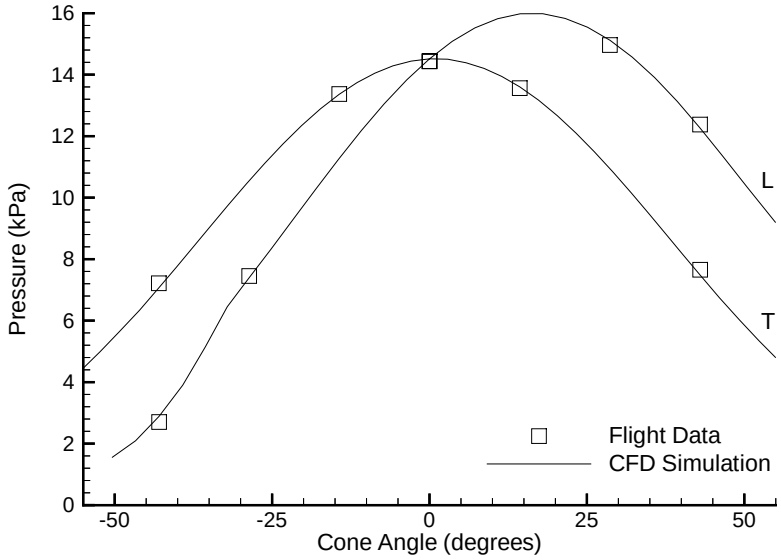



Figure 7.5: Reproduction of simulated pressures based on FADS air data predictions. The longitudinal (L) pressure sensors are drawn in the order (from left to right) ps3, ps2, ps1, ps4, ps5. The transverse (T) pressure sensors are drawn in the order ps9, ps8, ps1, ps6, ps7.

120 s flight time were used as input for a SF3D HYFLEX simulation. Free-stream static pressure and density, however, were constrained to meteorologically determined values for reasons that will become clear in the next section. Results from the CFD simulation are compared with flight measured pressure data in Figure 7.5. It is observed that the least squares inversion algorithm has correctly predicted air data parameters that align the simulated nose pressure distribution with flight measurements. Because a perfect fit could not be obtained, the least squares method has formed a compromise, where some sensor measurements slightly exceed the simulated pressures, while others fall a little below. The overall good agreement in Figure 7.5 implies that the surface pressure model of Equation 7.15 must be accurately describing the characteristics of CFD simulations.

## 7.6 Results and Discussion

We now test and compare the performance of the CFD-calibrated FADS algorithm, with a FADS algorithm based on the modified Newtonian theory pressure model. The comparison is conducted over the hypersonic part of the HYFLEX trajectory, using the HYFLEX pressure measurements recorded in flight as input.

Figure 7.6 shows the FADS predictions and IMU measurement of vehicle angle of attack. The CFD-FADS prediction is observed to agree closely with the IMU result, while

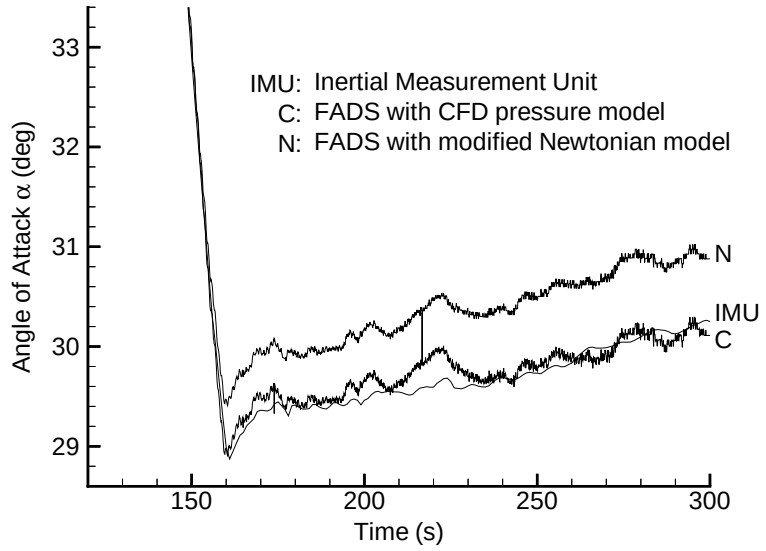



Figure 7.6: Comparison of angle of attack predictions made by the Inertial Measurement Unit (IMU), and the FADS using the CFD pressure model (C), and the modified Newtonian pressure model (N). Results are shown at a 10 Hz frequency.

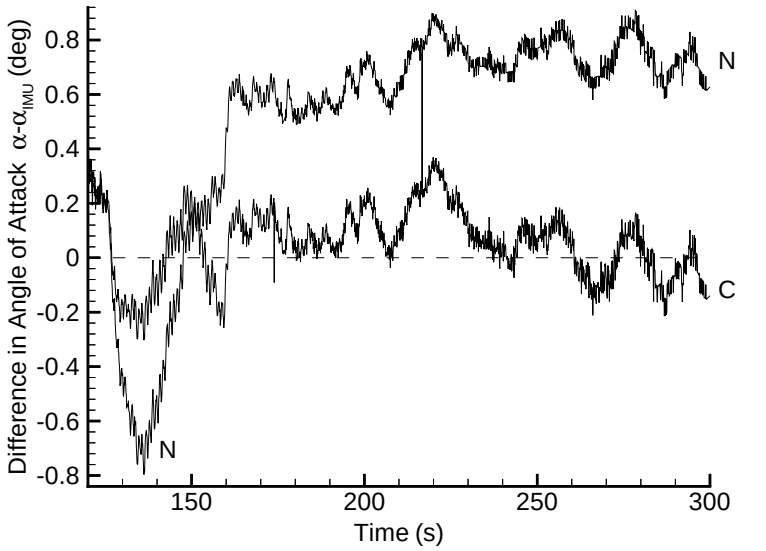



Figure 7.7: The difference between the angle of attack predicted by the FADS, and the IMU.

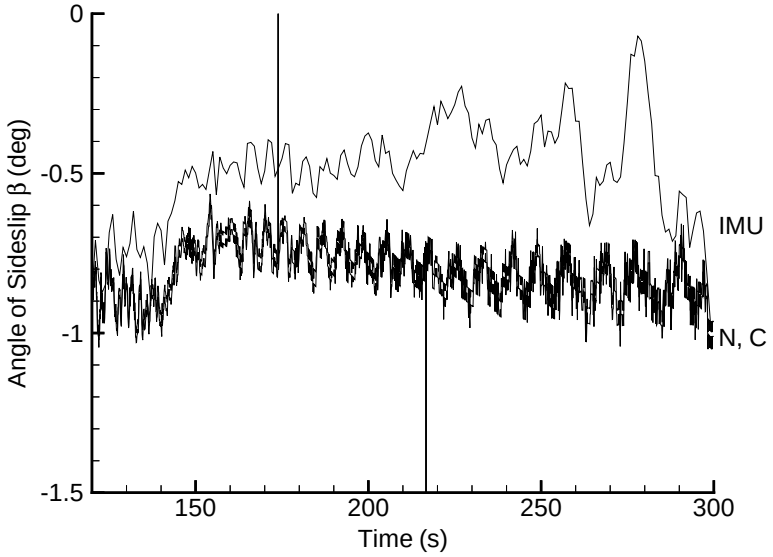



Figure 7.8: Comparison of angle of sideslip predictions made by the IMU and the FADS.

the Newtonian-FADS result deviates considerably from the IMU signal. A plot showing the difference between the FADS predictions and IMU result is presented in Figure 7.7. The HYFLEX handbook[9] reports that the accuracy of the IMU measurement is $\pm 0.65$ deg at the point of maximum dynamic pressure (140 s after flight commencement), decreasing to $\pm 0.45$ deg at Mach 3 (about 300 s after flight commencement). These quoted accuracies represent a variation of $\pm 3$ standard deviations. Thus, with the exception of a single data spike, the CFD-FADS result is completely contained within the quoted error band, while the Newtonian-FADS result lies mostly outside the band. The CFD-FADS estimates of $\alpha$ vary from the IMU with an average bias of $+0.05$ deg and standard deviation of 0.13 deg. The Newtonian-FADS shows an average bias of $+0.51$ deg and standard deviation of 0.39 deg. For a generic, broad-envelope, hypersonic vehicle, accuracies of 0.5 deg in angle of attack are typically required.[190] The results show that this kind of accuracy can be achieved with a CFD-calibrated FADS.

The accurate CFD-FADS prediction of $\alpha$ is strongly dependent on the $g(R, \alpha, i)$ term in the pressure correction function (Equation 7.15). As previously described, $g$ is used to account for the nonuniformity in nose radius near pressure ports 2 and 3. Without the inclusion of this term, it was found that the CFD-FADS estimates of $\alpha$ were of an accuracy similar to the Newtonian-FADS results. It thus appears that the inability of modified Newtonian theory to properly model ps2 and ps3, is responsible for its degraded performance in the prediction of $\alpha$.

Figure 7.8 compares the angle of sideslip estimates made by the FADS and the IMU.

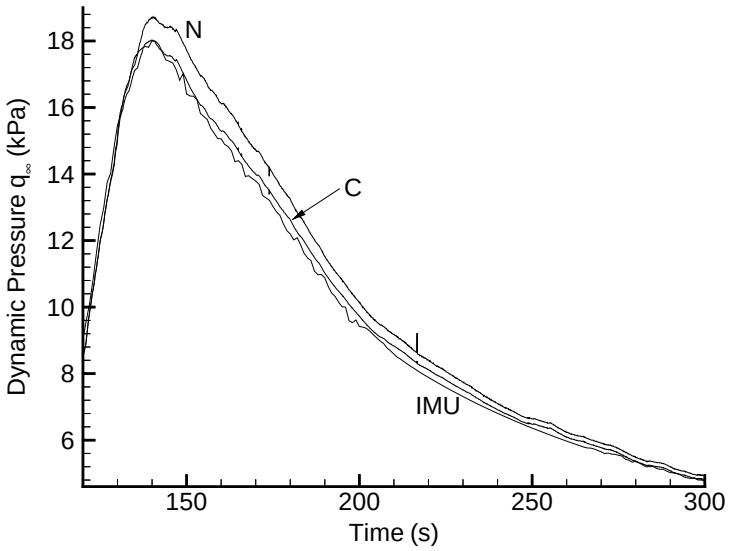



Figure 7.9: The dynamic pressure predicted by the IMU, compared with the predictions made by the FADS.

The estimates made by the CFD-FADS and Newtonian-FADS algorithms are close to identical. Relative to the IMU results, the CFD-FADS has an average bias in $\beta$ of $-0.34$ deg and a 0.17 deg standard deviation. The Newtonian-FADS has an average bias of $-0.30$ deg and a standard deviation of 0.16 deg. The IMU has a $\pm 3\sigma$ uncertainty in $\beta$ of $\pm 0.41$ deg at 140 s, and $\pm 0.60$ deg at 300 s.[9] For the control of a hypersonic vehicle, $\beta$ is typically required to an accuracy of 0.5 deg.[190] For both of the FADS results, accuracy is better than 0.5 deg for the most part.

Figure 7.9 displays the FADS and IMU predictions of free-stream dynamic pressure. Differences between the sets of data are more clearly seen in Figure 7.10, which shows the FADS predictions relative to the IMU. The $\pm 3\sigma$ uncertainty in the IMU estimated $q_\infty$ is at least $\pm 360$ Pa at 140 s, and $\pm 120$ Pa at 300 s.[9] The CFD-FADS $q_\infty$ estimates straddle the edge of the IMU error band, and have an average bias of $+130$ Pa with a 200 Pa standard deviation. The Newtonian-FADS estimates are well outside the quoted band, and have an average bias of $+500$ Pa and standard deviation of 410 Pa, with respect to the IMU data. Typically, a hypersonic vehicle requires dynamic pressure to within $\pm 1\%$ accuracy.[190] This requirement is not met by the Newtonian-FADS. Since the resolution of the IMU roughly corresponds to $\pm 2.5\%$ of $q_\infty$, it is not possible to say whether or not this goal was reached by the CFD-FADS.

The wind-tunnel calibration of the Shuttle Entry Air Data System (SEADS) is described in Reference 196. The paper includes an accuracy analysis of the SEADS for a

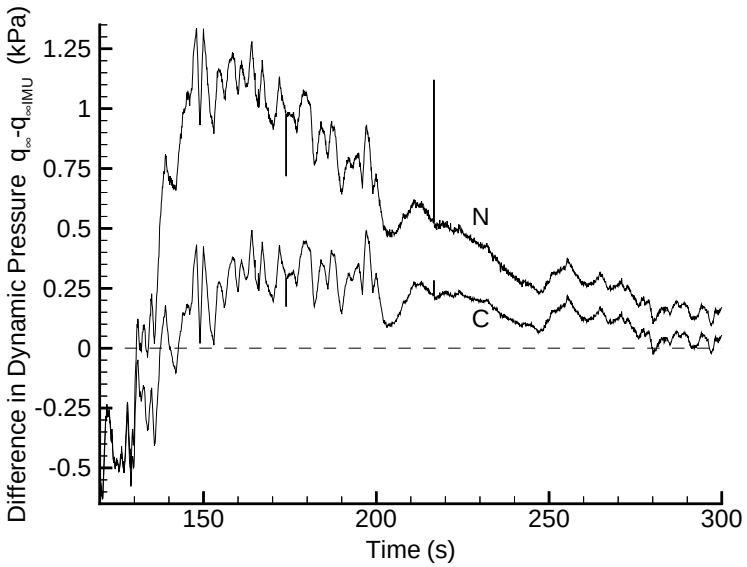



Figure 7.10: The difference between the dynamic pressure predictions made by the FADS, and the IMU.

shuttle entry, over the range Mach 0.5 to Mach 26. The air data accuracies are presented as average values over 100 s time intervals. Over the range Mach 2.8 to Mach 11.9, which corresponds to the HYFLEX flight speeds considered in this work, the maximum averaged error in SEADS $\alpha$ and $\beta$ estimates are reported to be 0.69 deg and 0.33 deg respectively. The maximum averaged error in SEADS stagnation pressure is 540 Pa (the absolute stagnation pressure corresponding to this error is not stated in Reference 196). The results from the wind-tunnel calibrated SEADS are thus of comparable accuracy to the CFD-calibrated HYFLEX FADS.

The final air data parameter to be investigated is the free-stream static pressure $P_\infty$. Figure 7.11 shows the FADS predictions of $P_\infty$, along with a value based on the HYFLEX trajectory and meteorological data. The CFD-FADS estimate is observed to be poor, while the Newtonian-FADS estimate is much worse. After examination of Equation 7.13 (or Equation 7.11), the cause of the bad $P_\infty$ estimates becomes apparent. At hypersonic flight conditions, the $P_\infty$ term is dwarfed by $P_{gt}$. Because of the disparity in relative magnitudes, it is very difficult to accurately resolve both quantities simultaneously. Small amounts of error or noise in the surface pressure measurements further compounds the problem. As discussed in Chapter 6, examples of this kind of error include pneumatic lag in the tubing connecting pressure ports to sensors, pressure port angular misalignment resulting from the deformation of the nosecap at high temperature, and sensor error itself.

The error in predicting $P_\infty$ consequently causes error in $R$ (Equation 7.14). Since $R$

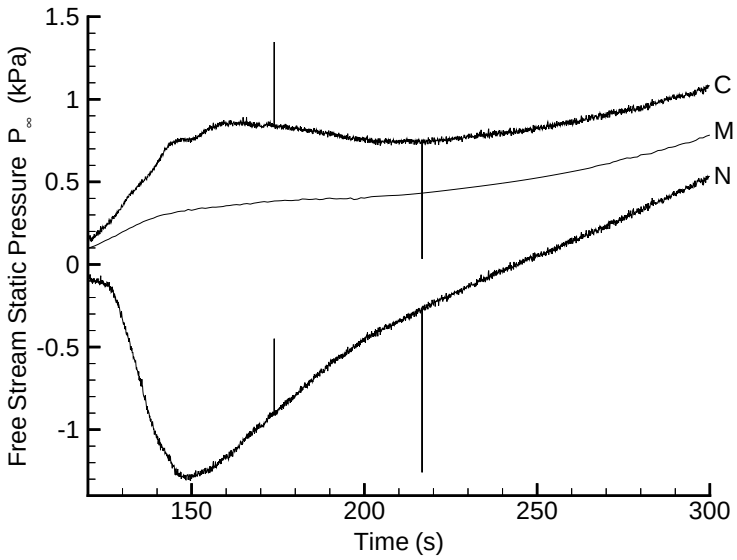



Figure 7.11: The free-stream static pressure estimated by the FADS, compared with the atmospheric pressure determined from meteorological data (M).

is used to determine many of the parameters in the CFD pressure correction model, the CFD-FADS prediction of other air data parameters (besides $P_\infty$) might thus be expected to suffer. Contrarily, Figs. 7.6–7.10 show that the CFD-FADS results are quite good. Good results are still obtained because the core curve-fit coefficients in the CFD pressure model ($c_k$ in Equation 7.16) are independent of $R$.

By using satellite positioning and meteorological data, it is possible to make a reasonably accurate estimate of $P_\infty$. If this data is available onboard the vehicle, it can be used as a supplement to help improve the accuracy of the FADS air data estimates. To investigate this technique, the FADS algorithm was tested with $P_\infty$ constrained to meteorological estimates. All the results shown in Figs. 7.6–7.8 did not change appreciably. However, the Newtonian-FADS prediction of $q_\infty$ did show significant improvement. By constraining $P_\infty$, the quality of the pressure model fit becomes less important in determining the magnitude of $P_{\text{gt}}$, and thus $q_\infty$.

The size of the pressure residuals ($E_i^*$ in Equation 7.2) are a useful indicator of how well the pressure models fit the measured data. A residual of zero implies a perfect fit. Shown in Figure 7.12 are the root mean square pressure residuals for the nine nose pressure sensors, expressed as a percentage of free-stream dynamic pressure. For both pressure models, erratic results are observed early in the flight, where the air density is low, the vehicle has greatest speed, and high temperature gas effects are strong. It may thus be concluded that either the pressure measurements, the pressure models, or both, lose ac-

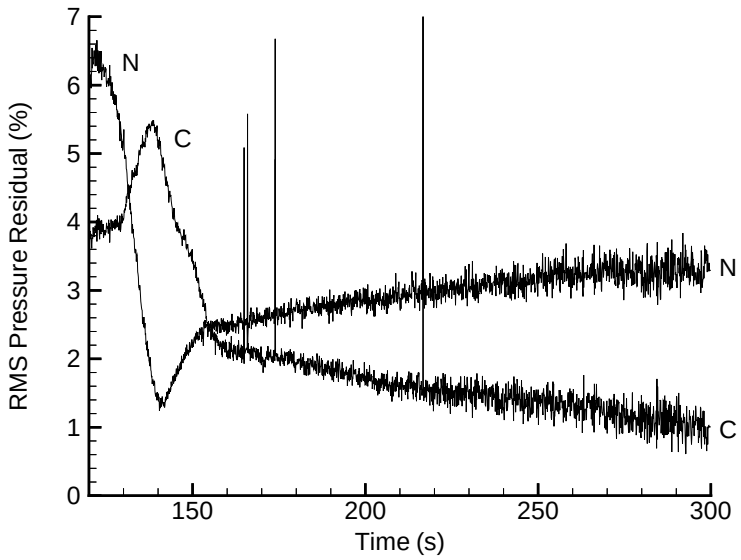



Figure 7.12: The RMS pressure residual for the FADS algorithm, expressed as a percentage of free-stream dynamic pressure.

curacy at the more extreme flight conditions. Later in the flight, the CFD pressure model displays a residual consistently smaller than the modified Newtonian pressure model.

## 7.7 Algorithm Convergence and Stability

Figure 7.13 shows the number of iterations required for convergence of the air data inversion algorithm. All air data parameters from state estimate $j+1$ must differ by less than 0.001% (or $1 \times 10^{-4}$ deg) from estimate $j$ before convergence is obtained (see Equation 7.3). The Newtonian-FADS takes about three iterations to reach convergence, while the CFD-FADS requires at least an extra two iterations. To understand why, consider the first order Taylor series expansion in Equation 7.5. By using this expansion, we are assuming that the pressure model behaves linearly about a particular air data state. The CFD pressure model, being more complex (and further from linear) than the modified Newtonian model, is adversely affected by the linearization to a greater extent. Thus the absence of high order terms in Equation 7.5 means that the inversion algorithm requires extra iterations to achieve convergence when the CFD pressure model is used. The overall speed of the CFD-FADS running on a MIPS R10000 195 MHz processor was 370 Hz. The corresponding speed for the Newtonian-FADS algorithm was 1270 Hz. Both speeds are in excess of 50 Hz, which is the typical in-flight requirement for a hypersonic vehicle air data system.[190]

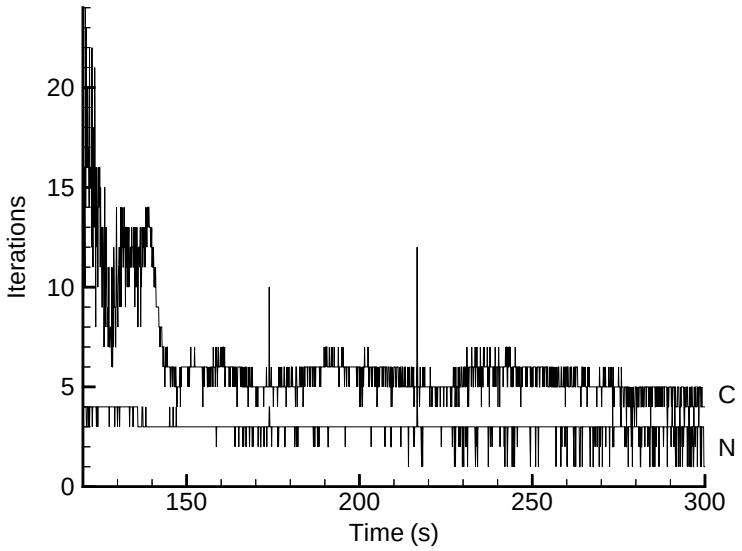



Figure 7.13: Number of iterations required for convergence of the FADS algorithm using different pressure models.

The singular values of the design matrix **A** provide a convenient way to check the stability of the air data solution technique. Singular values of zero indicate that column degeneracies exist in **A**, and that there are linear combinations of air data parameters that are ill-determined by Equation 7.4. To obtain meaningful and stable solutions from the air data system algorithm, it is important that all singular values remain significantly larger than machine precision. The condition number of **A** is also a useful indicator of solution stability, and is defined as the ratio of its largest and smallest singular value. A low condition number is desirable, and a condition number of infinity implies that the matrix is singular. Using the modified Newtonian pressure model, the worst (largest) design matrix condition number encountered was $1.24 \times 10^5$, while the worst (smallest) singular value was 0.49. For the CFD-calibrated pressure model, the worst condition number and singular value were $7.23 \times 10^4$ and 0.53 respectively. Thus the CFD-calibrated pressure model poses a better conditioned air data inversion problem. Additionally, the condition numbers and singular values of **A** lie well within machine precision for both pressure models.

In testing, the FADS algorithm proved to be robust, and converged for all the sets of pressure data, for both pressure models. This is despite a number of data spikes being present in the input data. The effect of these input spikes on air data prediction is clearly seen in the displayed results. Previous investigators[192] have encountered instability problems when the edge of the bow sonic zone nears or crosses pressure port locations. For

the HYFLEX flight, the sonic line crosses the outer transverse pressure ports (ps7 and ps9) at about 270 s after flight commencement. No perturbations in results or algorithm stability problems are observed at this time.

## 7.8 Summary of Calibration Results

A technique for calibrating the HYFLEX vehicle flush air data system (FADS) was described. The calibration involved using CFD simulation to provide the FADS algorithm with an accurate pressure model. No experimental or flight pressure data were used to augment the CFD pressure model. Using the pressure data recorded during the HYFLEX flight, air data predictions from the calibrated FADS were compared to those obtained from a FADS based on the modified Newtonian pressure model. The main findings were:

(i) The CFD-calibrated FADS predicted the vehicle angle of attack and angle of sideslip to approximately the accuracy required for control of a broad-envelope hypersonic vehicle. Dynamic pressure was predicted to an accuracy better or equal to that of the onboard inertial measurement unit.

(ii) The modified Newtonian FADS estimates of angle of attack and dynamic pressure did not meet the accuracy required for vehicle control. The modified Newtonian FADS angle of sideslip estimates were close to those made by the CFD-calibrated FADS.

(iii) Both the uncalibrated and calibrated FADS gave poor estimates of the free-stream static pressure.

(iv) The experimental calibration of air data system pressure models is not mandatory. Numerical results alone were sufficient to generate a good pressure model for the HYFLEX FADS in hypersonic flight.

(v) The numerical calibration procedure produced results of comparable accuracy to that obtained by procedures which use wind-tunnel data, such as in the Shuttle Entry Air Data System calibration.

(vi) The nonuniformity in radius of the HYFLEX nose made the CFD FADS calibration more difficult. Also, the nonuniformity was the predominate cause of error in the angle of attack predictions made by the uncalibrated FADS. In the future, it would be beneficial to position all pressure ports on a symmetric region of the nose.

CHAPTER 8

# Conclusion

This thesis was about the development of a Navier-Stokes solver optimized for blunt-body simulations, and its application to the calibration of a flush air data system. In the course of the work, some original computational fluid dynamics algorithms and air data system calibration techniques were presented, and new numerical simulation results were generated and analysed. Also, the occurrence of spurious bow-shock instability in simulations was investigated, and cured without loss of numerical accuracy.

The principal conclusion of this thesis, is that it is indeed possible to accurately calibrate a flush air data system for hypersonic flight using computational fluid dynamics techniques alone. Other conclusions, as well as a summary of ideas and experiences drawn from the work, are now listed in detail. Some recommendations for further research in the area are also proposed.

## 8.1 Computational Fluid Dynamics Code Development

The calibration of air data systems for blunt-nosed vehicles requires the establishment of an accurate relationship between surface pressure and free-stream variables. Thus, to achieve the thesis aims, it was necessary to develop a computational fluid dynamics (CFD) solver exhibiting the requisite pressure accuracy. Since an array of simulations is required for air data system calibration, it was also desired to choose CFD techniques that were not overly complicated or computationally expensive. To this end, a number of algorithms were investigated, in order to determine which were optimum for blunt-body flows. When no suitable algorithm already existed, some new techniques were developed.

Among the flux solvers tested, the advection upwind splitting method combining flux vector and difference splitting (AUSMDV), was found to be close to ideal for blunt-body simulations. The AUSMDV exhibits very little noise in the pressure distributions at shock and body, and also remains robust, stable, and accurate in the shock layer for gases with non-ideal equations of state. Boundary layers are accurately resolved by the AUSMDV, and the technique is computationally inexpensive. In common with the other low-dissipation flux solvers tested, however, the AUSMDV was found to suffer from an instability in captured bow shocks. The instability was aggravated at higher grid resolution, increased body bluntness, and with chemically reacting gases.

The bow-shock instability was found to be closely related to the carbuncle effect and odd-even flow variable decoupling, which have been reported in the literature previously. Results in this thesis confirmed postulates that the instability was caused by noise generated at flow states within the unphysically thick, captured shock wave. Noise in the flow velocity component tangential to the shock was shown to generate shock-layer vorticity, ultimately producing sustained vortices which perturbed the shock itself. Previous to this thesis, adding more dissipation during flux calculation was the commonly suggested cure.

Rather than add dissipation and decrease simulation accuracy, the low-dissipation AUSMDV was retained, and bow-shock stability was maintained using a shock fitting technique. Besides providing stability, shock fitting also enhances solution accuracy and boosts efficiency; shocks were resolved crisply, and post-shock noise was eliminated. Original approaches for interpolation and limiting at the shock, and shock speed upwinding, were presented. Such algorithms were found necessary to prevent the appearance of shock wave corrugations and post-shock oscillations. In practice, shock fitting allowed simulations of the HYFLEX (hypersonic flight experiment) to be easily produced for any angle of attack or sideslip, without having to manually adjust the computational grid to account for variation in shock position. A hybrid shock fitting and capturing strategy allowed transition between the two techniques at simulation runtime.

A second-order explicit time marching technique was used to advance HYFLEX simulations to convergence. It was decided that an implicit technique was not justified, due to the associated complexity and memory requirements. Further, the small timescales required to resolve nonequilibrium chemistry effects tend to negate the speed advantages of implicit techniques. The time accuracy of an explicit method was required to simulate flow establishment processes in shock tunnel experiments of the HYFLEX.

Flow reconstruction was performed using MUSCL interpolation and a van Albada limiter as the base scheme. By modifying the reconstruction to account for unequal cell spacing, it was demonstrated that good accuracy could be obtained on highly and abruptly clustered grids. The new reconstruction scheme causes a 15% increase in code execution

time compared to the often used MUSCL interpolation with minimum-modulus limiter, but the expense is justified by accuracy improvement in the case of the HYFLEX grid.

A range of general verification test cases showed that the code was solving the Navier-Stokes equations correctly, that conservation laws were observed, and that sufficient convergence rates were being obtained. Specific blunt-body validation tests showed that the physical system was being correctly modelled. Good agreement in drag, shock shape, and standoff distance, was achieved between simulation and experimental results.

## 8.2 Flush Air Data System Calibration

Modified Newtonian theory is quite simple, yet predicts surface pressure on hypersonic blunt bodies to a reasonable accuracy. A pressure model with higher accuracy, however, is needed for the calibration of hypersonic flush air data systems (FADS). In this thesis, the HYFLEX flush air data system (FADS) was calibrated by developing and applying a correction function to account for flow phenomena that are not modelled by modified Newtonian theory. These phenomena include physical flow interaction in the shock layer, boundary layer development, and nonequilibrium chemistry effects. Three possible calibration techniques for generating the correction function were discussed: flight data calibration, wind tunnel calibration, and computational fluid dynamics calibration. Flight data calibration was ruled out due to expense, and because it is a technique that is not available for a first flight.

Due to the high speeds at which the HYFLEX FADS was to be calibrated, the possibility of ground-based experimental calibration could only be investigated using an impulse facility, such as a shock tunnel. The uncertainty in measuring surface pressure on a HYFLEX scale model in the T4 shock tunnel was 4%. Since CFD simulations showed that modified Newtonian theory is already in error by approximately 4%, it is unrealistic to try to calibrate a hypersonic FADS using shock tunnel results. Difficulties in obtaining a spatially and temporally uniform free stream flow, problems of scale, cost, and uncertainty in free stream conditions, further prohibit the use of a shock tunnel for FADS calibration.

It was thus decided to calibrate the HYFLEX FADS using only computational fluid dynamics results. It is believed that FADS calibration for a hypersonic flight vehicle, based exclusively on CFD results, has never before been performed. A grid convergence test for CFD simulations of the HYFLEX showed that the governing equations were being solved to 0.08% accuracy, which is much less than the total modified Newtonian error, and deemed sufficient. However, the convergence test does not determine the accuracy at which the CFD code models the physical system. The lack of accurate experimental

data for blunt-body flows, though, makes strict validation difficult at hypersonic speeds. While available for comparison, the nose pressures measured during the HYFLEX flight experiment were not used as validation evidence or for code tuning, since this would have defeated the aim of conducting a CFD-only FADS calibration.

Results from simulations of the HYFLEX showed that, at hypersonic speeds, unmodelled flow interaction within the shock layer is the largest single factor contributing to error in modified Newtonian theory. The remaining error, roughly one-quarter of the total, is due to the aforementioned viscous and nonequilibrium effects. It is thus concluded that, although boundary layer development and chemical nonequilibrium do not strongly affect surface pressure, their impact is significant enough to warrant inclusion in a hypersonic FADS calibration. It was found that vehicle surface temperature had a negligible effect on simulated surface pressures.

When tested with flight data, the CFD-calibrated FADS reduced error in the prediction of vehicle angle of attack and dynamic pressure by approximately a factor of two, compared to the uncalibrated FADS. Angle of sideslip prediction was not improved. Accuracy of the CFD-calibrated FADS mostly satisfied the level required for the control of a broad-envelope flight vehicle, and was generally at least as good as the onboard inertial measurement unit (IMU). Also, accuracy of the CFD-calibrated HYFLEX FADS was similar to that obtained from the experimentally calibrated shuttle entry air data system. Hence, it is concluded that experimental data is not required to obtain an accurate FADS calibration. Additionally, the CFD calibration has the advantages of low cost and repeatability. Once a suitable code has been developed and validated, the calibration procedure is also relatively quick.

Some of the pressure sensors on the HYFLEX bow were located in a region of nonuniform nose radius. CFD results showed that a decrease in the HYFLEX nose radius generated a sudden expansion, and this was accounted for in the FADS calibration. The expansion process was not correctly modelled by modified Newtonian theory and, as a consequence, contributed to the degraded performance of the uncalibrated FADS. It is hence a recommendation that, where possible, pressure sensors be located on symmetric regions of the nose in future FADS designs.

The calibration pressure correction function was developed by examining simulation results, and fitting curves to the observed trends. Hence, the FADS inversion algorithm relies upon the correction function to describe simulation behaviour in terms of air data parameters, rather than directly using CFD itself. Although a novel idea at present, it would be more convenient to actually incorporate a CFD solver within the inversion routine. However, operating at 20 Hz, with roughly four iterations required for each air data state prediction, eighty simulations would need to be performed every second! With

currently available processors, it requires about two days to perform a single HYFLEX simulation. Computer speed, though, is roughly doubling every eighteen months. If this exponential rate is maintained, it is imaginable that onboard real-time CFD could be feasible within forty years. Until then, however, the calibration procedure described in this thesis provides a good substitute.

The reliability and accuracy of air data reporting for a hypersonic vehicle in flight is critical for successful control. Although the presented calibration procedure works well, there are a number of matters that need further development or investigation before the system can be relied upon in flight: (i) CFD-calibrated FADS results need to be compared against air data of higher accuracy than can be obtained from an IMU. Unfortunately, since FADS and IMU air data are the most accurate available, it is not immediately clear how this can be accomplished without leading to circular arguments. (ii) While the least squares inversion technique exhibited robustness and accuracy in this thesis, the author is dubious about relying on the convergence of Newton iterations for such a flight critical system, especially in the case of low quality pressure data. It would be interesting to compare the error tolerance and efficiency of the least squares technique, with a CFD-calibrated neural network, or the method of pressure triples. Provided the calibrations are correctly performed, it is maintained that accuracy should be nearly independent of the inversion technique. (iii) It is possible to accurately measure wind tunnel and flight pressure data at supersonic, transonic, and subsonic speeds. Previous investigators have traditionally used such data sources to calibrate lower speed flush air data systems. It would be worthwhile to investigate the use of CFD to calibrate a FADS for these regimes as well, with a view to eventually developing numerical calibrations that span all speeds, from subsonic to hypersonic.

APPENDIX A

# Gas Data

Viscosity curve fit parameters for various species are shown in Table A.1, and are valid over the temperature range $1,000\,\text{K} \leq T \leq 30,000\,\text{K}$. The parameters for $N_2$, $O_2$, N, O, and NO were sourced from Reference 63, $CO_2$ from Reference 205, and CO from Reference 206. The viscosity of species *i* is calculated using

$$\mu_i = 0.1\, e^{C_{\mu,i}} T^{B_{\mu,i}+A_{\mu,i}\ln T} \text{ Pa s.} \tag{A.1}$$

Table A.2 lists thermodynamic data for various gases at standard atmospheric conditions. Under the calorifically perfect assumption, these data are taken to be constant and independent of the gas state.

Molecular weights and enthalpies of formation (referenced to absolute zero temperature) for the component species of air are shown in Table A.3.

Table A.4 shows the polynomial curve fit coefficients for the thermodynamic properties of air component species, in the format described in Section 2.9.1. For each species, the first set of coefficients are valid over the temperature range 300 K$\leq T \leq$1,000 K, the second set for 1,000 K$\leq T \leq$6,000 K, and the third set for 6,000 K$\leq T \leq$15,000 K. The $A_{i,j}$ coefficients are based on those presented in Reference 63, but have been modified in order to maintain continuity between temperature intervals.

Table A.1: Viscosity curve fit parameters.

| Species | $A_{\mu,i}$ | $B_{\mu,i}$ | $C_{\mu,i}$ |
|---|---|---|---|
| $CO_2$ | $2.0400E-02$ | $4.3120E-01$ | $-1.1826E+01$ |
| $N_2$ | $2.0300E-02$ | $4.3290E-01$ | $-1.1815E+01$ |
| $O_2$ | $4.8400E-02$ | $-1.4550E-01$ | $-8.9231E+00$ |
| CO | $2.0500E-02$ | $4.2890E-01$ | $-1.1818E+01$ |
| NO | $4.5200E-02$ | $-6.0900E-02$ | $-9.4596E+00$ |
| N | $1.2000E-02$ | $5.9300E-01$ | $-1.2381E+01$ |
| O | $2.0500E-02$ | $4.2570E-01$ | $-1.1580E+01$ |

Table A.2: Thermodynamic data for gases at 300 K and 100 kPa.

| Gas | $c_p$ (J/kg K) | $c_v$ (J/kg K) | $R$ (J/kg K) | $\gamma$ |
|---|---|---|---|---|
| Air | 1004.5 | 717.5 | 287.0 | 1.40 |
| $CO_2$ | 864.0 | 675.0 | 189.0 | 1.28 |
| He | 5191.0 | 3114.0 | 2077.0 | 1.67 |

Table A.3: Molecular weights and formation enthalpies.

| Species | $M_i$ (kg/mol) | $h_i^{f(0K)}$ (J/kg) |
|---|---|---|
| $N_2$ | $2.8013E-02$ | $0.0000E+00$ |
| $O_2$ | $3.1999E-02$ | $0.0000E+00$ |
| N | $1.4007E-02$ | $4.7082E+05$ |
| O | $1.5999E-02$ | $2.4679E+05$ |
| NO | $3.0006E-02$ | $8.9775E+04$ |

Table A.4: Polynomial curve fit coefficients for species thermodynamics.

| $A_{i,0}$ | $A_{i,1}(\mathrm{K}^{-1})$ | $A_{i,2}(\mathrm{K}^{-2})$ | $A_{i,3}(\mathrm{K}^{-3})$ | $A_{i,4}(\mathrm{K}^{-4})$ | $s^{\mathrm{c}}(\mathrm{J/kgK})$ |
|---|---|---|---|---|---|
| $N_2$ | | | | | |
| $3.6748E+00$ | $-1.2081E-03$ | $2.3240E-06$ | $-6.3218E-10$ | $-2.2577E-13$ | $6.9985E+02$ |
| $3.1846E+00$ | $1.0137E-03$ | $-3.0467E-07$ | $4.1091E-11$ | $-2.0170E-15$ | $1.3522E+03$ |
| $3.1603E+00$ | $8.9745E-04$ | $-2.0216E-07$ | $1.8266E-11$ | $-5.0334E-16$ | $1.4167E+03$ |
| $O_2$ | | | | | |
| $3.6146E+00$ | $-1.8598E-03$ | $7.0814E-06$ | $-6.8070E-09$ | $2.1628E-12$ | $1.1336E+03$ |
| $3.6009E+00$ | $7.5213E-04$ | $-1.8732E-07$ | $2.7913E-11$ | $-1.5774E-15$ | $9.7255E+02$ |
| $3.8679E+00$ | $3.2510E-04$ | $-9.2131E-09$ | $-7.8684E-13$ | $2.9426E-17$ | $6.0328E+02$ |
| N | | | | | |
| $2.5031E+00$ | $-2.1800E-05$ | $5.4205E-08$ | $-5.6476E-11$ | $2.0999E-14$ | $2.4739E+03$ |
| $2.4766E+00$ | $6.9258E-05$ | $-6.3065E-08$ | $1.8387E-11$ | $-1.1747E-15$ | $2.5517E+03$ |
| $2.7355E+00$ | $-3.9090E-04$ | $1.3380E-07$ | $-1.1910E-11$ | $3.3690E-16$ | $1.7542E+03$ |
| O | | | | | |
| $2.8236E+00$ | $-8.9478E-04$ | $8.3060E-07$ | $-1.6837E-10$ | $-7.3205E-14$ | $1.8202E+03$ |
| $2.5444E+00$ | $-2.7551E-05$ | $-3.1028E-09$ | $4.5511E-12$ | $-4.3681E-16$ | $2.5491E+03$ |
| $2.5519E+00$ | $-5.9520E-05$ | $2.7010E-08$ | $-2.7980E-12$ | $9.3800E-17$ | $2.5189E+03$ |
| NO | | | | | |
| $3.5887E+00$ | $-1.2479E-03$ | $3.9786E-06$ | $-2.8651E-09$ | $6.3015E-13$ | $1.4269E+03$ |
| $3.2092E+00$ | $1.2705E-03$ | $-4.6603E-07$ | $7.5007E-11$ | $-4.2314E-15$ | $1.8436E+03$ |
| $3.8401E+00$ | $2.3409E-04$ | $-2.1354E-08$ | $1.6689E-12$ | $-4.9070E-17$ | $9.1580E+02$ |

APPENDIX B

# Interferometry

Interferometry is a non-intrusive flow visualization technique, and gives information about the distribution of density and species concentration throughout the flowfield. Conventional Mach-Zehnder interferometry uses a coherent laser beam of fixed wavelength, which is initially split in two. One beam is passed through the disturbed test flow, while the other, termed the reference beam, is passed around the flow using periscopes. The final phase of each beam is affected by the density and composition of the gas it travels through, as well as its path length. After the beams are recombined, phase differences between them cause the light to interfere either constructively or destructively, forming observable fringes. Because the reference beam is passed through a gas of known properties, it is possible to infer information about the test flow from the resulting image (the interferogram). This technique is called infinite-fringe Mach-Zehnder interferometry.

If the test flow conditions do not vary markedly from the reference conditions, few fringes will be visible and quantitative data gleaned from the interferogram will likely be inaccurate. To remedy this effect, carrier fringes can be introduced into the image by varying the path length of the reference beam across its cross section. After beam recombination, this procedure forms a set of background fringes upon which small flow disturbances may easily be observed. The use of background fringes is called finite-fringe interferometry.

A problem with Mach-Zehnder interferometry is that high quality optical equipment is required. To avoid phase shift errors, the reference beam must be passed through optical windows of the exact same thickness and properties as the test beam. Because the test section windows on hypervelocity wind tunnels are easily damaged, this is a difficult requirement to satisfy. Holographic interferometry avoids the problem, by passing both

reference and test beams along the same path. Before flow begins, a hologram of the test section is recorded on a holographic plate. During the test flow, a second hologram is recorded on the same plate, and interference between the two holograms results. By firing a laser through the holographic plate after the experiment, the holographic image can be reconstructed and photographed. Since the reference and test beams are recorded at different times, it is important to ensure that the optics do not move between exposures. More information on holographic interferometry can be obtained from References 125 and 207.

There are two approaches to comparing computational results with experimental interferograms. The first is to process the experimental interferogram image, by hand or on computer, to determine the distribution of density throughout the field of view. This technique is time-consuming, difficult, and error-prone. The second approach is to process the simulation results using computational flow imaging, to generate a simulated interferogram which can be compared directly to the experimental interferogram. We use the second approach in this thesis.

Generation of the computational interferogram is quite straight-forward for both two- and three-dimensional flows. Extending the Gladstone-Dale equation[208] yields the fringe shift as

$$F = \int_{z_1}^{z_2} \left( \frac{\rho - \rho_{\text{ref}}}{\lambda} \sum_{i=1}^{NS} K_i C_i \right) dz, \tag{B.1}$$

if the test beam passes through the flow in the $z$ direction. The limits of integration, $z_1$ and $z_2$, define the path length of the beam. The reference density is denoted by $\rho_{\text{ref}}$, and does not vary with $z$ in most practical interferograms. The test flow density $\rho$ and species mass fractions $C_i$ vary with the Cartesian coordinates $x$, $y$, and $z$ in three-dimensional and axisymmetric flows, but only vary with $x$ and $y$ in two-dimensional flows. The wavelength of the laser light is $\lambda$, and the Gladstone-Dale constant for species $i$ is $K_i$. Terming $K_i$ a constant is somewhat of a misnomer, since it does vary slightly with wavelength. We will, however, ignore this effect. Table B.1 gives a listing of the Gladstone-Dale constants used to produce the simulated interferograms in this thesis, sourced from References 125 and 208.

The final step in forming a computational interferogram is to calculate the light intensity across the image. Intensity, $I$, is given by

$$I = \cos^2(\pi F + \delta), \tag{B.2}$$

where $\delta$ is an offset to compensate for any background phase shift between test and reference beams in the experimental interferogram.

We make special mention of the procedure used for the calculation of axisymmetric interferograms. In this case, a full three-dimensional simulation is not required to produce the interferogram. The simple mapping

$$(x, y, z) \rightarrow (x, \sqrt{y^2 + z^2}, 0) \tag{B.3}$$

is used to extend results from axisymmetric simulations performed on two dimensional grids to an equivalent representation in three-dimensional space, before application of Equation B.1.

Table B.1: Gladstone-Dale constants.

| Species | $K$ ($m^3$/kg) |
|---|---|
| $CO_2$ | $2.30 \times 10^{-4}$ |
| $N_2$ | $2.46 \times 10^{-4}$ |
| $O_2$ | $1.90 \times 10^{-4}$ |
| CO | $2.70 \times 10^{-4}$ |
| N | $3.22 \times 10^{-4}$ |
| O | $1.82 \times 10^{-4}$ |

APPENDIX C

# SF3D Version 4.1

The SF3D computer program was written expressly for this thesis, and thus has the primary purpose of performing computational fluid dynamics simulations of blunt body flows. Version 1.0 was finished in October 1995, and as at January 1999 the latest version is 4.1. Code development was conducted on a Silicon Graphics Origin 2000 system (shown in Figure C.1) running IRIX Release 6.4. Most of the simulations presented in this thesis were also performed on the Origin. The program is available on CD-ROM from the Department of Mechanical Engineering at the University of Queensland. This appendix describes the code components, and presents a brief user guide.

Figure C.1: The Silicon Graphics Origin 2000.

## C.1 Compiling SF3D

The SF3D program is coded entirely in the C programming language,[209] and should port to any computer loaded with a relatively recent ANSI C compiler. SF3D has been successfully compiled and executed under the MS-DOS, Linux, DEC UNIX, SunOS, AIX and IRIX operating systems.

A makefile is included with the SF3D distribution to allow easy compilation. The makefile should be invoked using

`make [`*`target`*`] [opt=`*`level`*`]`

With no arguments, SF3D will be made for a general system at the compiler default optimization level. If desired, a different optimization *`level`* can be used. A good compromise between speed and floating point accuracy is `opt=3` on most systems. On Silicon Graphics IRIX systems, the makefile will take advantage of parallel compilers if the target is specified as `sg`. For DEC UNIX machines, use `dec` as the target. The arguments `debug` and `profile` can be used to include symbolic debugging labels or implement performance profiling, respectively. If a `make clean` is performed, all executables and object files will be cleared from the current working directory. After a successful compilation, the binary executable `sf3d.x` should be available for immediate use. Table C.1 lists the main files which comprise the source code for SF3D, and gives a description of the purpose of the functions contained within.

## C.2 Using SF3D

Before running SF3D, several simulation definition files must be present in, or linked to, the directory containing the executable.

The `profile.3d` file contains most options and switches pertaining to the flow solver, and administrative details about the simulation. Rarely adjusted tuning parameters and options are hard coded in `sf3d.h`. Included in `profile.3d` are details about the grid size, simulation run time, gas chemistry model, and simulation project name. The specific format of the file is defined by a sample template in the SF3D distribution.

The `flowspec.3d` file contains the initial and boundary conditions for the flow. The first five lines specify the initial density, internal energy, and velocity components in SI units, in that order. Subsequently, each of the six boundary conditions are defined according to the codes in Table C.2. Note that SF3D will report an error if a boundary condition is incompatible with the selected `profile.3d` simulation options. A non-slip wall should not be used in an inviscid simulation, for example. Code `9` ensures

Table C.1: Descriptions of some of the SF3D source code files.

| Source file | Description |
|---|---|
| `aireq.c` | Equilibrium air equation of state routines.[a] |
| `chem.c` | Calculates nonequilibrium reaction kinetics.[b] |
| `co2eq.c` | Equilibrium carbon dioxide equation of state routines.[56] |
| `engine.c` | Performs time integration of the Navier-Stokes equations. |
| `flux.c` | Contains flow reconstruction algorithms and flux solvers. |
| `gas.c` | Determines thermodynamic and transport properties. |
| `geom.c` | Calculates geometric properties of the grid and performs vector operations. |
| `grid.c` | Reads, writes, transforms and creates grids. |
| `icbc.c` | Sets initial and boundary conditions. |
| `io.c` | Loads and saves solution files, reads simulation definition files, and displays status reports. |
| `main.c` | Contains the parent calling function. |
| `mem.c` | Allocates, clears and frees heap memory as necessary. |
| `moving.c` | Performs shock fitting, grid movement, and GCL calculations. |
| `n2eq.c` | Equilibrium nitrogen equation of state routines.[a] |
| `rivp.c` | Approximate Riemann solver.[a,96] |
| `sf3d.h` | Header file, including definitions of adjustable parameters. |
| `visc.c` | Calculates viscous stresses and heat conduction. |

[a] Written by Peter Jacobs.

[b] Rewritten and modified versions of original routines by Chris Craddock.

Table C.2: Boundary condition codes for `flowspec.3d`.

| Code | Description |
|---|---|
| 1 | Inflow. |
| 2 | Outflow. |
| 3 | Adiabatic frictionless wall. |
| 4 | Isothermal frictionless wall. |
| 5 | Adiabatic non-slip wall. |
| 6 | Isothermal non-slip wall. |
| 9 | Boundary adjoins another boundary. |

compatibility between mating boundaries, and is a special option that can only be used on boundaries 5 and 6 (see Figure 2.1). The format of the `flowspec.3d` file is defined by a sample template included in the SF3D distribution.

If shock fitting has been selected in `profile.3d`, boundary 4 will be used as the shock boundary by default. An inflow condition on this boundary, containing the upstream flow state, should be specified in `flowspec.3d` in this case.

If a nonequilibrium chemistry model has been selected, the reaction pathways, reactions rates, and species thermodynamics should be defined in `chemistry.3d`. The format of this file is defined by a template included in the distribution.

For new simulations, the user has the option of loading a grid from disk, or using a grid generation routine hard coded in `grid.c`. Grids on disk should have the filename *`project.mode`*`.g`, where `project` is the name of the simulation defined in `profile.3d`, and *`mode`* is either `cap` or `fit`, depending on whether shock capturing or shock fitting is being performed. The first line of a grid file should contain the grid dimensions (number of cells) in the order $i$, $j$, $k$. The remainder of the grid file must contain Cartesian triples $x$, $y$, $z$ on separate lines, defining the cell vertices. Vertices should be listed so that the $i$ counter increments most frequently, followed by $j$, and then by $k$. If SF3D complains that negative volumes have been calculated, check that the vertices define grid cells which conform to the right-hand geometry convention. Newly generated grids, or those resulting from shock fitting, are saved under the filename *`project.mode`*`.g.new`, so as not to overwrite existing gridfiles.

It is a good idea to specify that SF3D should write solution files regularly during simulations, to allow the code to be restarted should a system shutdown occur, and to monitor solution progress. Solution files have the unique filename *`project_n.mode`*`.pl`, where *`n`* is the timestep number . The solutions are in text format and can be visually inspected, and all flow and grid information required for a simulation restart are contained

within. To avoid postprocessing delays, solution files are written in a format directly readable by the TECPLOT data visualization program.[210,211] Solution files converted to TECPLOT binary format, however, may no longer be used to restart a simulation.

An example of screen output from SF3D is shown in Figure C.2. At startup, the code reads the simulation input files and echoes some of the more important selected options to the screen, for confirmation. The abbreviation to the left of each message indicates the function which generated it. When SF3D knows the grid resolution and simulation options, it will attempt to allocate the required memory. The names of all large arrays and their memory requirements are listed during allocation, to allow diagnosis if insufficient memory is available.

The user is provided with a progress update approximately once per minute, via the display function. With reference to Figure C.2, the first line of the update contains the current simulation time in seconds, the run time of the code in seconds, and the current timestep number. The timestep (in seconds; see Equation 2.22) is shown on the second line. Also shown is the current computer speed, expressed in microseconds required per cell per step. Because other users may be also be running jobs on the system, the maximum speed obtainable may be better than the achieved speed. To the right of the current speed is the maximum speed, and the corresponding percentage of CPU time used by SF3D. The third line of the progress update indicates which cell is limiting the timestep. The cell indices include a ghost cell offset, and the number of ghost cells used (usually three) should be subtracted from the displayed values to obtain a raw grid reference. The index direction causing the limitation is also displayed, and the limiting wave speed and cell length are shown on the line below in SI units. The fifth line contains the maximum normalized density residual and its location, to help indicate when convergence to steady-state is achieved. If shock fitting is used, the final display line will contain the number of boundary cell interfaces currently being fitted or tracked.

When the program finishes executing, SF3D will report if any errors or exceptions were encountered. If so, it is advisable to review the screen output to locate possible problems. For this reason, capturing the screen output to a log file is a good idea for long simulations.

```
main:      SF3D - Shock Fitting and Capturing 3D
main:      -------------------------------------
main:      Rev 4.1, 1995-1998.      Ian Johnston

loadprof:  PROJECT ID = hyflex
loadprof:  loading profile
loadprof:  grid is 15i x 46j x 45k
loadprof:  solution will be resumed from hyflex.fit.pl
loadprof:  ...done
chemgnum:  Using a 5 species, 6 reaction model
memget:    requesting 32023200 bytes of memory (gc)
memget:    ...allocated O.K.
                   ⋮
cheminit:  initialising finite rate chemistry
cheminit:  ...done
loadfs:    loading flow specification file
loadfs:    ...done
loadsoln:  loading solution hyflex_29324.fit.pl
loadsoln:  solution will be resumed from
loadsoln:  time = 4.500534e-03,  step = 29324,  dt = 1.485805e-07
loadsoln:  solution succesfully loaded
geomgrid:  updating cell geometry
geomgrid:  ...done
main:      starting main loop
main:      solution will proceed in FIT mode
                   ⋮
display:   simtime = 5.248925e-03, wallclock = 87594, step = 34347
display:   dt = 1.489704e-07, speed = 543us/542us/100%, eta = 8.28hrs
display:   timestep limited by 15i 29j 26k jk_face
display:   lspeed = 1.230301e+03,  llen = 3.665569e-04
display:   maximum normalized residual 9.250066e-07 at 16i 6j 4k
display:   shock elements fitted = 2070,   tracked = 0
                   ⋮
tplot:     writing a tecplot solution
tplot:     simtime = 5.5003e-04, step = 36057
tplot:     ...done

memfree:   releasing allocated memory
memfree:   ... done

main:      SF3D program normal exit
```

Figure C.2: Sample SF3D screen output.

APPENDIX D

# HYFLEX Data

This appendix contains drawings of the HYFLEX vehicle, including details of pressure tapping locations on the nosecap. Also included in the appendix are various data recorded in flight, and NOAA atmospheric sounding data relevant to the time and location of the flight experiment. The experiment commenced with a launch at 8:00 am on 12 February 1996, with the vehicle following an entry trajectory contained approximately within the bounds 29°–30.5° North (latitude) and 135.7°–143.5° East (longitude).

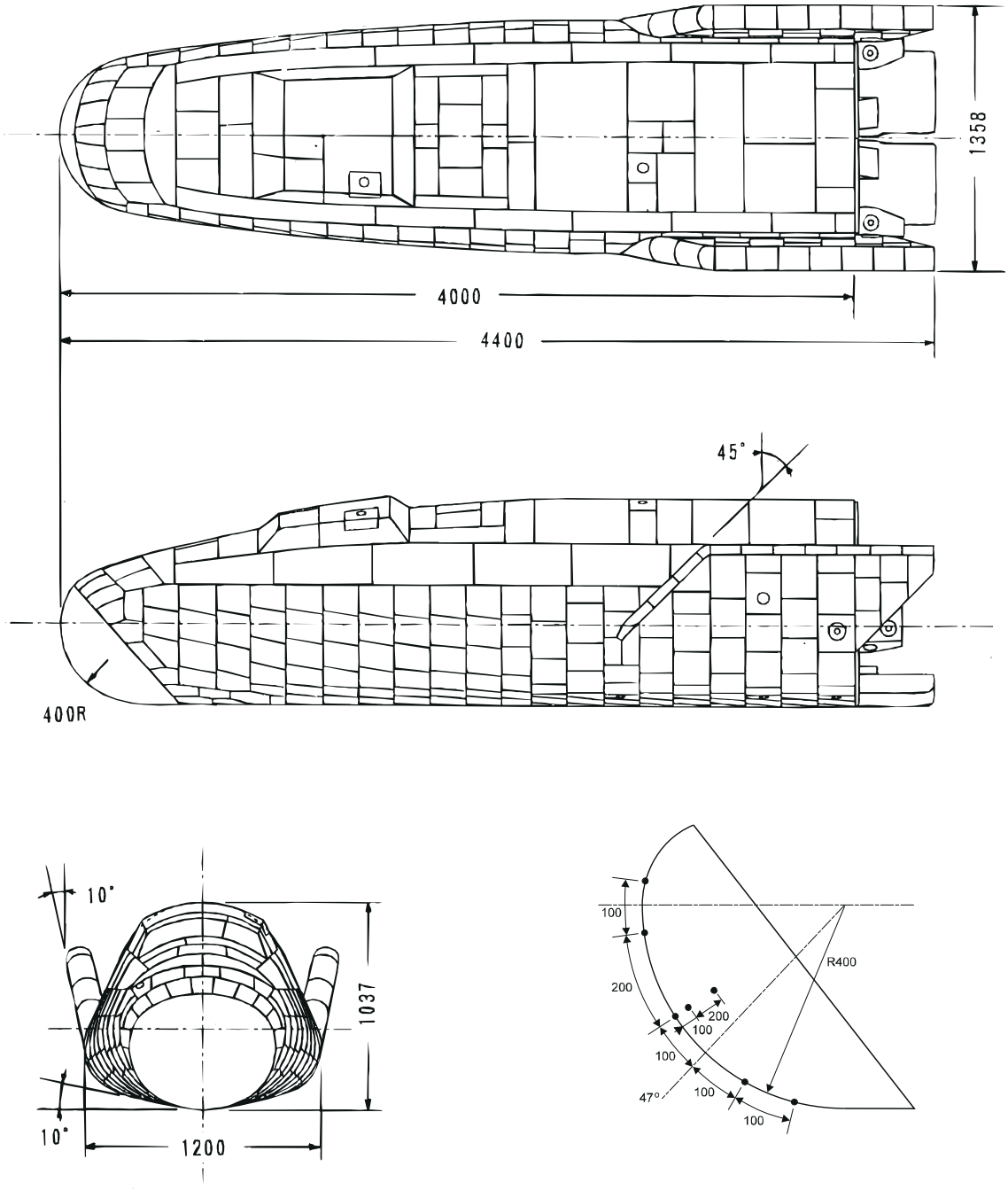



Figure D.1: From top to bottom, drawings of the plan, side, and front views of the HYFLEX (Source: Reference 9). A side view of the locations of the nose pressure tappings is pictured lower right. All dimensions are in mm.

Table D.1: Vehicle state and atmospheric conditions during the HYFLEX descent.

| Quantity | $t = 90$ s | $t = 120$ s | $t = 150$ s | $t = 180$ s |
|---|---|---|---|---|
| $\rho^{a,b}$ (kg/m$^3$) | $7.606\times10^{-5}$ | $1.295\times10^{-3}$ | $4.553\times10^{-3}$ | $5.464\times10^{-3}$ |
| $P^{a,b}$ (Pa) | 4.9 | 97.7 | 327.9 | 389.0 |
| $T^{a}$ (K) | 223.4 | 262.8 | 250.9 | 248.0 |
| $T^{b}$ (K) | 222.6 | 261.8 | 250.0 | 247.1 |
| $C^{b}_{N_2}$ | 0.766 | 0.765 | 0.765 | 0.765 |
| $C^{b}_{O_2}$ | 0.234 | 0.235 | 0.235 | 0.235 |
| $V^{b,c}$ (m/s) | 3965.8 | 3736.7 | 2684.5 | 2112.7 |
| $\alpha^{b,c}$ (deg) | 50.00 | 48.79 | 32.90 | 29.41 |
| $\beta^{b,c}$ (deg) | -0.94 | -0.87 | -0.50 | -0.54 |
| $h$ (km) | 69.8 | 48.0 | 38.7 | 37.4 |

| Quantity | $t = 210$ s | $t = 240$ s | $t = 270$ s | $t = 300$ s |
|---|---|---|---|---|
| $\rho^{a,b}$ (kg/m$^3$) | $5.921\times10^{-3}$ | $7.062\times10^{-3}$ | $8.902\times10^{-3}$ | $1.180\times10^{-2}$ |
| $P^{a,b}$ (Pa) | 419.1 | 492.1 | 608.3 | 783.5 |
| $T^{a}$ (K) | 246.6 | 242.8 | 238.1 | 231.4 |
| $T^{b}$ (K) | 245.7 | 241.9 | 237.2 | 230.5 |
| $C^{b}_{N_2}$ | 0.765 | 0.765 | 0.765 | 0.765 |
| $C^{b}_{O_2}$ | 0.235 | 0.235 | 0.235 | 0.235 |
| $V^{b,c}$ (m/s) | 1703.1 | 1388.3 | 1126.6 | 899.1 |
| $\alpha^{b,c}$ (deg) | 29.54 | 29.69 | 29.99 | 30.25 |
| $\beta^{b,c}$ (deg) | -0.55 | -0.47 | -0.47 | -0.96 |
| $h$ (km) | 36.7 | 35.7 | 34.2 | 32.4 |

[a] Atmospheric value.

[b] Simulated condition.

[c] According to the inertial measurement unit.

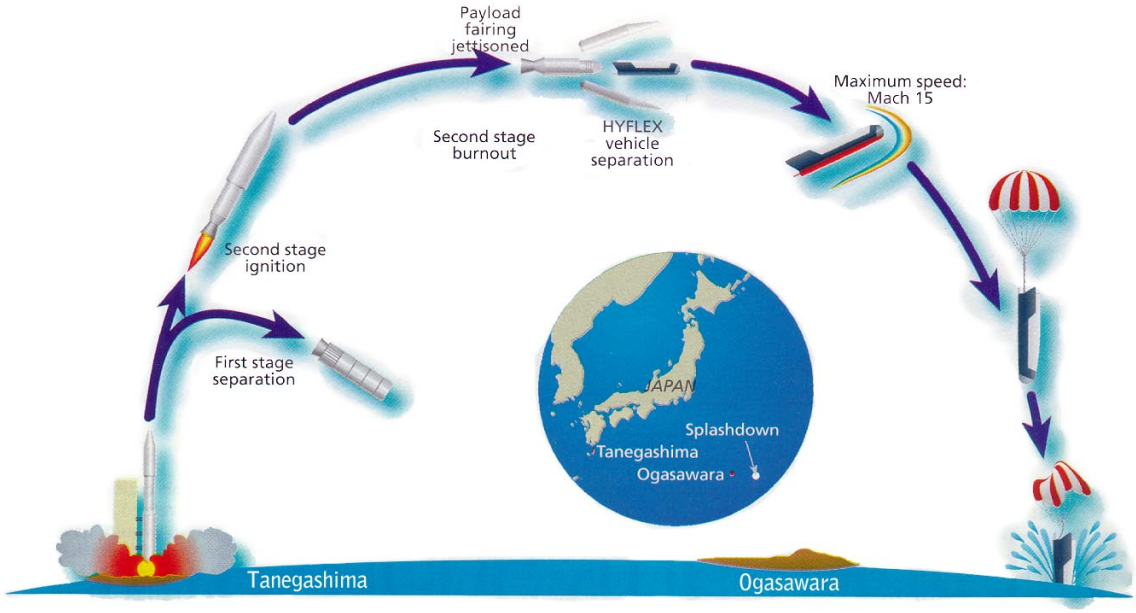



Figure D.2: Diagram of the HYFLEX flight, adapted from Reference 10.

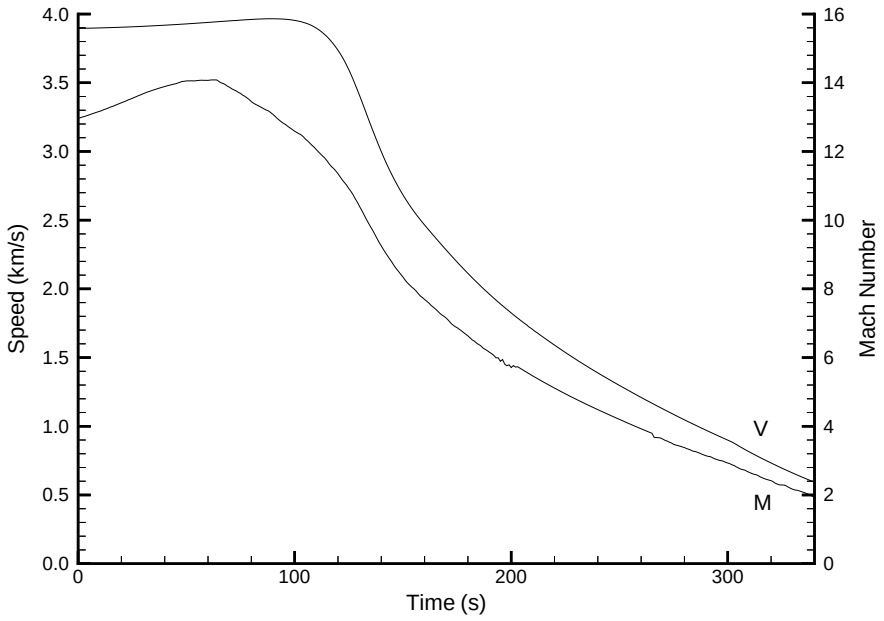



Figure D.3: Speed $V$ and Mach number $M$ of the HYFLEX versus time after separation.

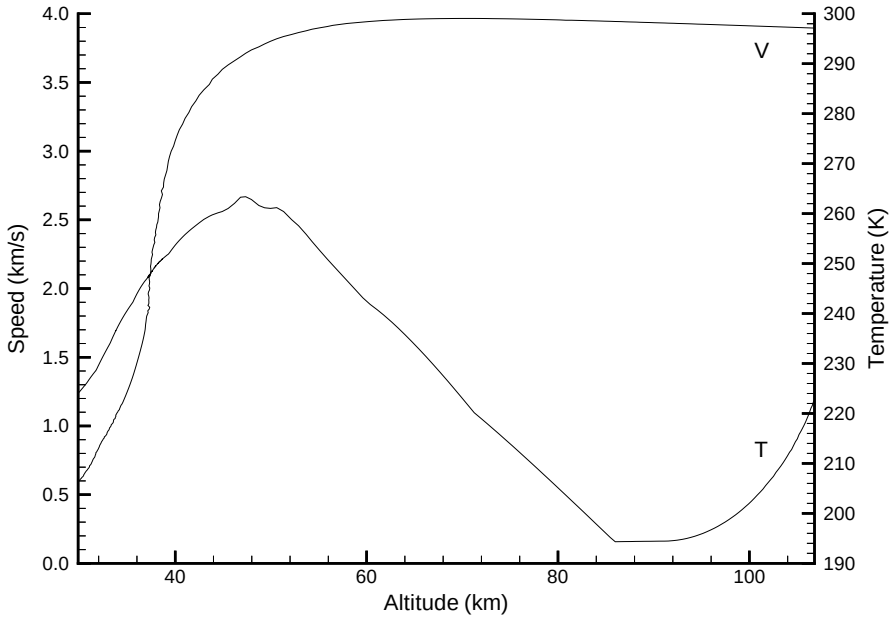



Figure D.4: HYFLEX speed $V$ and atmospheric temperature $T$ versus altitude.

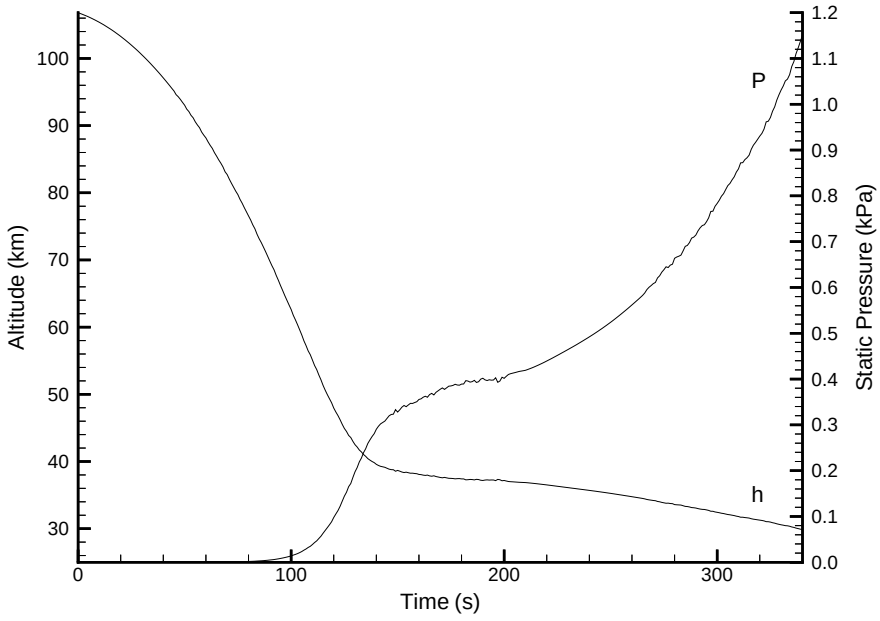



Figure D.5: Altitude of the HYFLEX $h$ and atmospheric pressure $P$ versus time after separation.

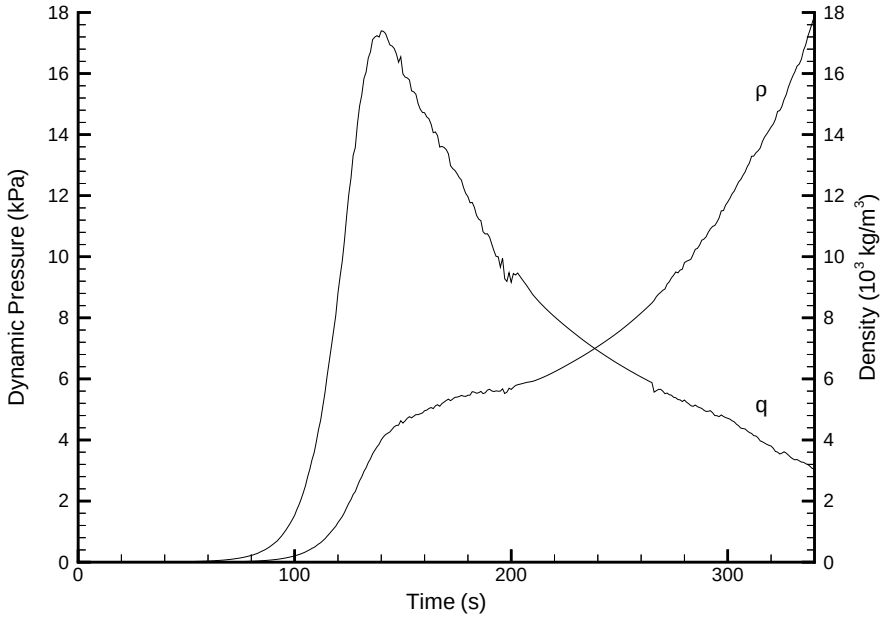



Figure D.6: Vehicle dynamic pressure $q$ and atmospheric density $\rho$ versus time after separation.

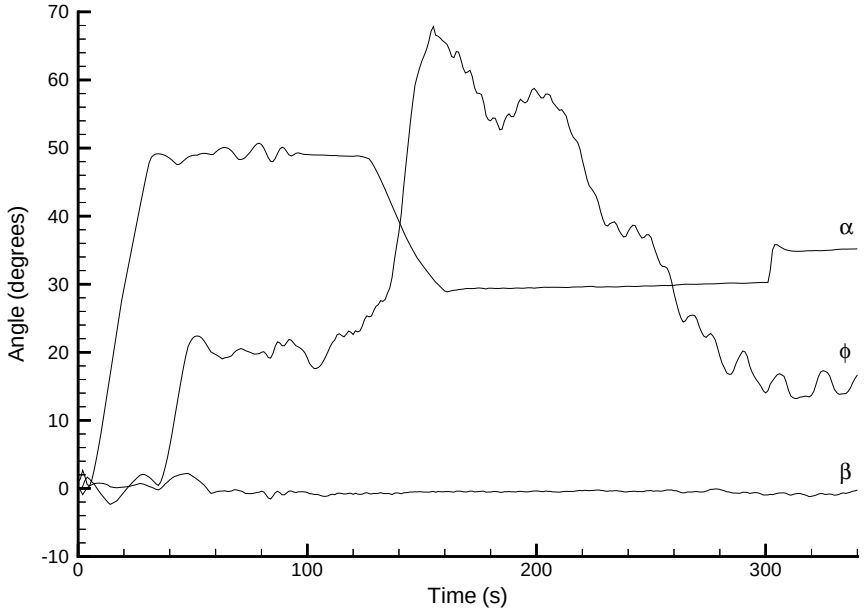



Figure D.7: Angle of attack $\alpha$, angle of sideslip $\beta$, and roll angle $\phi$ of the HYFLEX versus time after separation.

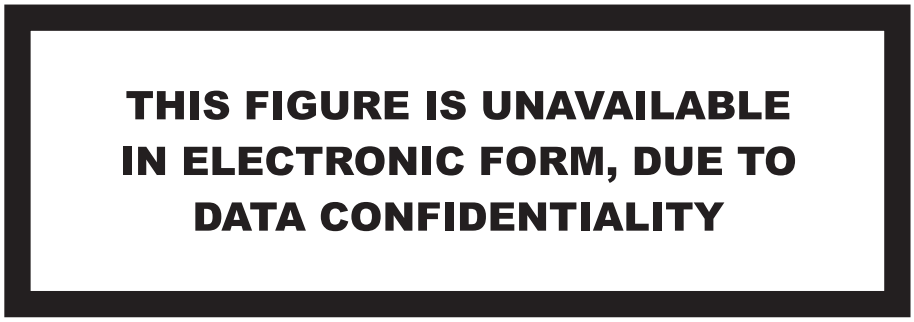



Figure D.8: Pressure sensor readings from the HYFLEX nosecap (ps1–ps5) versus time.

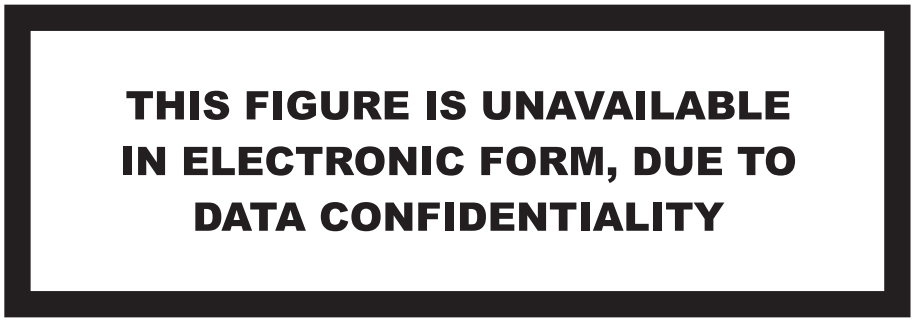



Figure D.9: Pressure sensor readings from the HYFLEX nosecap (ps1, ps6–ps9) versus time.

# References


[1] Mason, W. H. and Lee, J., “On Optimal Supersonic/Hypersonic Bodies,” AIAA Paper 90-3072, 1990.

[2] Whitmore, S. A., Cobleigh, B. R., and Haering, E. A., “Design and Calibration of the X-33 Flush Airdata Sensing (FADS) System,” NASA TM-1998-206540, 1998.

[3] Dornheim, M. A., “X-33 Design Gets Go-ahead,” *Aviation Week & Space Technology*, Vol. 147, No. 19, Nov. 1997, pp. 50–52.

[4] Lindberg, R. E. and Feconda, R. T., “X-34: A Test Bed for RLV Technology,” *Aerospace America*, Vol. 36, No. 8, Aug. 1998, pp. 30–33.

[5] Siemers, P. M., Wolf, H., and Flanagan, P. F., “Shuttle Entry Air Data System Concepts Applied to Space Shuttle Orbiter Flight Pressure Data to Determine Air Data — STS 1–4,” AIAA Paper 83-0118, 1983.

[6] Whitmore, S. A. and Moes, T. R., “The Effects of Pressure Sensor Acoustics on Air-data Derived from a High Angle-of-Attack Flush Airdata Sensing (HI-FADS) System,” AIAA Paper 91-0671, 1991.

[7] Chapman, D. R., Mark, H., and Pirtle, M. W., “Computers vs. Wind Tunnels for Aerodynamic Flow Simulations,” *Astronautics and Aeronautics*, Vol. 13, No. 4, Apr. 1975, pp. 23–35.

[8] Babbage, C., *Passages From the Life of a Philosopher*, IEEE Press, Piscataway, 1994, originally published 1864.

[9] NAL/NASDA HOPE Cooperative Research Team HYFLEX Subgroup, *Hypersonic Flight Experiment Handbook*, 1994, NN-95-3009 (in Japanese).

[10] Mullins, J., “Faith, Hope and Rocketry,” *New Scientist*, Vol. 149, No. 2014, Jan. 1996, pp. 38–40.

[11] Rusanov, V. V., "A Blunt Body in a Supersonic Stream," *Annual Review of Fluid Mechanics*, Vol. 8, 1976, pp. 377–404.

[12] Mason, W. H. and Lee, J., "Aerodynamically Blunt and Sharp Bodies," *Journal of Spacecraft and Rockets*, Vol. 31, No. 3, May–Jun. 1994, pp. 378–382.

[13] Anderson, J. D., *Hypersonic and High Temperature Gas Dynamics*, McGraw-Hill, New York, 1989.

[14] Hirsch, C., "Part V: The Numerical Computation of Potential Flows," *Numerical Computation of Internal and External Flows*, Vol. 2, Wiley, Chichester, 1988, pp. 1–117.

[15] Anderson, J. D., *Computational Fluid Dynamics: The Basics with Applications*, McGraw-Hill, New York, 1995.

[16] Bird, G. A., *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford Engineering Science Series 42, Clarendon Press, Oxford, 1994.

[17] Zhong, X. and Furumoto, G. H., "Augmented Burnett-Equation Solutions over Axisymmetric Blunt Bodies in Hypersonic Flow," *Journal of Spacecraft and Rockets*, Vol. 32, No. 4, Jul.–Aug. 1995, pp. 588–595.

[18] Rizzi, A. W. and Inouye, M., "Time-Split Finite-Volume Method for Three-Dimensional Blunt-Body Flow," *AIAA Journal*, Vol. 11, No. 11, Nov. 1973, pp. 1478–1485.

[19] Dick, E., *Introduction to Finite Volume Techniques in Compuational Fluid Dynamics*, chap. 11, Wendt,[212] 2nd ed., 1996, pp. 269–297.

[20] Hirsch, C., "Finite Volume Method and Conservative Discretizations," *Numerical Computation of Internal and External Flows*, Vol. 1, chap. 6, Wiley, Chichester, 1988, pp. 237–264.

[21] MacCormack, R. W., "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354, 1969.

[22] Hirsch, C., "The Finite Difference Method," *Numerical Computation of Internal and External Flows*, Vol. 1, chap. 4, Wiley, Chichester, 1988, pp. 167–200.

[23] Hirsch, C., "The Finite Element Method," *Numerical Computation of Internal and External Flows*, Vol. 1, chap. 5, Wiley, Chichester, 1988, pp. 201–236.

[24] Dick, E., *Introduction to Finite Element Techniques in Computational Fluid Dynamics*, chap. 10, Wendt,[212] 2nd ed., 1996, pp. 230–268.

[25] Fornberg, B., "Introduction," *A Practical Guide to Pseudospectral Methods*, chap. 1, Cambridge University Press, New York, 1996, pp. 1–3.

[26] Hussaini, M. Y. and Zang, T. A., "Spectral Methods in Fluid Dynamics," *Annual Review of Fluid Mechanics*, Vol. 19, 1987, pp. 339–367.

[27] Dimitropoulos, C. D., Edwards, B. J., Chae, K.-S., and Beris, A. N., "Efficient Pseudospectral Flow Simulations in Moderately Complex Geometries," *Journal of Compuational Physics*, Vol. 144, No. 2, Aug. 1998, pp. 517–549.

[28] Selmin, V., "The node-centred finite volume approach: Bridge between finite differences and finite elements," *Computer Methods in Applied Mechanics and Engineering*, Vol. 102, No. 1, Jan. 1993, pp. 107–138.

[29] Chakravarthy, S., "On Some Unification Themes in Computational Fluid Dynamics," *Fifteenth International Conference on Numerical Methods in Fluid Dynamics*, edited by R. Kutler, J. Flores, and J.-J. Chattot, No. 490 in Lecture Notes in Physics, Springer, Berlin, Jun. 1996, pp. 213–218.

[30] Hirsch, C., "The Mathematical Nature of the Flow Equations and their Boundary Conditions," *Numerical Computation of Internal and External Flows*, Vol. 1, chap. 3, Wiley, Chichester, 1988, pp. 133–166.

[31] Hirsch, C., "Finite Volume Method and Conservative Discretizations," *Numerical Computation of Internal and External Flows*, Vol. 1, chap. 12, Wiley, Chichester, 1988, pp. 457–504.

[32] Degrez, G., *Implicit Time-Dependent Methods for Inviscid and Viscous Compressible Flows, With a Discussion of the Concept of Numerical Dissipation*, chap. 9, Wendt,[212] 2nd ed., 1996, pp. 180–229.

[33] Hirsch, C., "Part II: Basic Discretization Techniques," *Numerical Computation of Internal and External Flows*, Vol. 1, Wiley, Chichester, 1988, pp. 161–166,424.

[34] Sheffer, S. G., Martinelli, L., and Jameson, A., "An Efficient Multigrid Algorithm for Compressible Reactive Flows," *Journal of Computational Physics*, Vol. 144, No. 2, Aug. 1998, pp. 484–516.

[35] Gerald, C. F. and Wheatley, P. O., *Applied Numerical Analysis*, chap. 5, Addison-Wesley, 1989, pp. 347–410.

[36] Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes," AIAA Paper 81-1259, 1981.

[37] Swanson, R. and Turkel, E., "A Multistage Time-Stepping Scheme for the Navier-Stokes Equations," Report 84-62, ICASE, Feb. 1984, Also NASA CR-172527.

[38] Pike, J. and Roe, P. L., "Accelerated Convergence of Jameson's Finite-Volume Euler Scheme using Van Der Houwen Integrators," *Computers and Fluids*, Vol. 13, No. 2, 1985, pp. 223–236.

[39] Courant, R., Friedrichs, K., and Lewy, H., "Über die partiellen Differenzengleichungen der mathematischen Physik," *Mathematische Annalen*, Vol. 100, 1928, pp. 32–74, (in German).

[40] Courant, R., Friedrichs, K., and Lewy, H., "On the Partial Difference Equations of Mathematical Physics," *IBM Journal*, Vol. 11, No. 2, Mar. 1967, pp. 215–234, (English translation).

[41] Gottlieb, S. and Shu, C.-W., "Total Variation Diminishing Runge-Kutta Schemes," Tech. Rep. CR-201591, NASA, Jul. 1996, Also ICASE Report 96-50.

[42] Thomas, P. D. and Lombard, C. K., "Geometric Conservation Law and Its Application to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, No. 10, Oct. 1979, pp. 1030–1037.

[43] Gaitonde, A. L., Jones, D. P., and Fiddes, S. P., "A 2D Navier-Stokes Method for Unsteady Compressible Flow Calculations On Moving Meshes," *The Aeronautical Journal*, Vol. 2, No. 1012, Feb. 1998, pp. 89–97.

[44] Holman, J. P., *Heat Transfer*, McGraw Hill, London, 7th ed., 1992.

[45] Swanson, R. C., Turkel, E., and White, J. A., "An Effective Multigrid Method for High-Speed Flows," *Fifth Copper Mountain Conference on Multigrid Methods*, Mar.–Apr. 1991.

[46] Wahls, R. A., Barnwell, R. W., and DeJarnette, F. R., "Finite-Difference Outer-Layer, Analytic Innter-Layer Method for Turbulent Boundary Layers," *AIAA Journal*, Vol. 27, No. 1, Jan. 1989, pp. 15–22.

[47] Barnwell, R. W., "Nonadiabatic and Three-Dimensional Effects in Compressible Turbulent Boundary Layers," *AIAA Journal*, Vol. 30, No. 4, Apr. 1992, pp. 897–904.

[48] Thompson, R. A., "Comparison of Nonequilibrium Viscous-Shock-Layer Solutions with Shuttle Heating Measurements," *Journal of Thermophysics and Heat Transfer*, Vol. 4, No. 2, Apr. 1990, pp. 162–169.

[49] Tannehill, J. C., Anderson, D. A., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, chap. 8, Taylor & Francis, Washington DC, 2nd ed., 1997.

[50] Hirsch, C., "The Dynamic Levels of Approximation," *Numerical Computation of Internal and External Flows*, Vol. 1, chap. 2, Wiley, Chichester, 1988, pp. 26–132.

[51] Gupta, R. N., "Two- and Three-Dimensional Analysis of Hypersonic Nonequilibrium Low-Density Flows," *Journal of Thermophysics and Heat Transfer*, Vol. 10, No. 2, Apr.–Jun. 1996, pp. 267–276.

[52] Wilke, C. R., "A Viscosity Equation for Gas Mixtures," *The Journal of Chemical Physics*, Vol. 18, No. 4, Apr. 1950, pp. 517–519.

[53] Reid, R. C., Prausnitz, J. M., and Poling, B. E., *The Properties of Gases and Liquids*, McGraw Hill, New York, 4th ed., 1987.

[54] Wylen, G. J. V. and Sonntag, R. E., *Fundamentals of Classical Thermodynamics*, Wiley, New York, 3rd ed., 1985.

[55] Thivet, F., Candel, S., and Perrin, M. Y., "Hypersonic Viscous Shock Layer in Thermochemical Nonequilibrium," *Proceedings of the 1st International Symposium on Aerothermodynamics of Space Vehicles*, Noorwijk, The Netherlands, 1991, pp. 325–333, ESA SP-318.

[56] Johnston, I. A., "A Thermodynamic Model of the Martian Atmosphere for Computational Fluid Dynamics Analyses," Honours Thesis, Department of Mechanical Engineering, The University of Queensland, 1994.

[57] Jacobs, P. A., "Single-Block Navier-Stokes Integrator," ICASE Interim Report 18, NASA CR-187613, Jul. 1991.

[58] Prabhu, R. K. and Erickson, W. D., "A Rapid Method for the Computation of Equilibrium Chemical Composition of Air to 15000 K," NASA TP-2792, 1988.

[59] Srinivasan, S., Tannehill, J. C., and Weilmuenster, K. J., "Simplified Curve Fits for the Thermodynamics Properties of Equilibrium Air," NASA RP-1181, 1987.

[60] Fedkiw, R. P., Merriman, B., and Osher, S., "High Accuracy Numerical Methods for Thermally Perfect Gas Flows with Chemistry," *Journal of Computational Physics*, Vol. 132, No. 2, Apr. 1997, pp. 175–190.

[61] Craddock, C. S., "A Quasi-One-Dimensional Space-Marching Flow Solver with Finite Rate Chemical Effects," Department of Mechanical Engineering, The University of Queensland, Technical Report 7/96, 1996.

[62] Vincenti, W. G. and Kruger, C. H., *Introduction to Physical Gas Dynamics*, Wiley, New York, 1975.

[63] Gupta, R. N., Yos, J. M., Thompson, R. A., and Lee, K.-P., "A Review of Reaction Rates and Thermodynamic and Transport Properties for an 11-Species Air Model for Chemical and Thermal Nonequilibirum Calculations to 30000K," NASA RP-1232, Aug. 1990.

[64] Blottner, F. G., "Viscous Shock Layer at the Stagnation Point with Nonequilibrium Air Chemistry," *AIAA Journal*, Vol. 7, No. 12, Dec. 1969, pp. 2281–2288.

[65] Godunov, S. K., "A Difference Scheme for Numerical Computation of Discontinuous Solution of Hydrodynamic Equations," *Matematicheskii Sbornik*, Vol. 47, 1959, pp. 271–306, (in Russian).

[66] Venkatakrishnan, V., "Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters," *Journal of Computational Physics*, Vol. 118, No. 1, Apr. 1995, pp. 120–130.

[67] van Leer, B., "Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov's Method," *Journal of Computational Physics*, Vol. 32, No. 1, Jul. 1979, pp. 101–136.

[68] Choi, H. and Liu, J.-G., "The Reconstruction of Upwind Fluxes for Conservation Laws: Its Behavior in Dynamics and Steady State Calculations," *Journal of Computational Physics*, Vol. 144, No. 2, Aug. 1998, pp. 237–256.

[69] Hirsch, C., "Second-order Upwind and High-Resolution Schemes," *Numerical Computation of Internal and External Flows*, Vol. 2, chap. 21, Wiley, Chichester, 1988, pp. 493–594.

[70] Arora, M. and Roe, P. L., "A Well-Behaved TVD Limiter for High-Resolution Calculations of Unsteady Flow," *Journal of Computational Physics*, Vol. 132, No. 1, Mar. 1997, pp. 3–11.

[71] Spekreijse, S., "Multigrid Solution of Monotone Second-Order Discretizations of Hyperbolic Conservation Laws," *Mathematics of Computation*, Vol. 49, No. 179, Jul. 1987, pp. 135–155.

[72] Shu, C.-W., "TVB Uniformly High-Order Schemes for Conservation Laws," *Mathematics of Computation*, Vol. 49, No. 179, Jul. 1987, pp. 105–121.

[73] Suresh, A. and Huynh, H. T., "Compuations of Slowly Moving Shocks," *Journal of Computational Physics*, Vol. 136, No. 1, Sep. 1997, pp. 83–99.

[74] Anderson, W. K., Thomas, J. L., and van Leer, B., "A Comparison of Finite Volume Flux Vector Splittings for the Euler Equations," AIAA Paper 85-0122, Jan. 1985.

[75] Sweby, P. K., "High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws," *SIAM Journal on Numerical Analysis*, Vol. 21, No. 5, Oct. 1984, pp. 995–1011.

[76] van Albada, G. D., van Leer, B., and Roberts, W. W., "A Comparative Study of Computational Methods in Cosmic Gas Dynamics," *Anstronomy and Astrophysics*, Vol. 108, 1982, pp. 76–84.

[77] Swanson, R. C., Radespiel, R., and Turkel, E., "On Some Numerical Dissipation Schemes," *Journal of Computational Physics*, Vol. 147, No. 2, Dec. 1998, pp. 518–544.

[78] Jameson, A., "Analysis and Design of Numerical Schemes for Gas Dynamics. 2. Artificial Diffusion and Discrete Shock Structure," *International Journal of Computational Fluid Dynamics*, Vol. 5, No. 1–2, 1995, pp. 1–38.

[79] Roe, P., "A Survey of Upwind Differencing Techniques," *Eleventh International Conference on Numerical Methods in Fluid Dynamics*, edited by D. Dwoyer, M. Hussaini, and R. Voigt, No. 323 in Lecture Notes in Physics, Springer-Verlag, Berlin, Jul. 1988, pp. 69–78.

[80] Sengupta, T. K., "Compressible Flows," *Computational Fluid Flow and Heat Transfer*, edited by K. Muralidhar and T. Sundararajan, chap. 11, Narosa, New Delhi, 1995, pp. 387–428.

[81] Fey, M., "Multidimensional Upwinding. Part 1. The Methods of Transport for Solving the Euler Equations," *Journal of Computational Physics*, Vol. 143, No. 1, Jun. 1998, pp. 159–180.

[82] Fey, M., "Multidimensional Upwinding. Part 2. Decomposition of the Euler Equations into Advection Equations," *Journal of Computational Physics*, Vol. 143, No. 1, Jun. 1998, pp. 181–199.

[83] Lax, P. D. and Liu, X.-D., "Solution of Two-Dimensional Riemann Problems of Gas Dynamics by Positive Schemes," *SIAM Journal on Scientific Computing*, Vol. 19, No. 2, Mar. 1998, pp. 319–340.

[84] Steger, J. L. and Warming, R., "Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite-Difference Methods," *Journal of Computational Physics*, Vol. 40, No. 2, Mar. 1981, pp. 263–293.

[85] van Leer, B., "Flux-Vector Splitting for the Euler Equations," *Eighth International Conference on Numerical Methods in Fluid Dynamics*, edited by E. Krause, Vol. 170 of *Lecture Notes in Physics*, Springer-Verlag, Aachen, 1982, pp. 507–512.

[86] Amaladas, J. R. and Kamath, H., "Accuracy Assessment of Upwind Algorithms for Steady-State Computations," *Computers & Fluids*, Vol. 27, No. 8, Nov. 1998, pp. 941–962.

[87] van Leer, B., Thomas, J. L., Roe, P. L., and Newsome, R. W., "A Comparison of Numerical Flux Formulas for the Euler and Navier-Stokes Equations," AIAA Paper 87-1104, 1987.

[88] Edwards, J. R., "A Low-Diffusion Flux Splitting Scheme for Navier-Stokes Calculations," *Computers and Fluids*, Vol. 26, No. 6, 1997, pp. 635–659.

[89] Radespiel, R. and Kroll, N., "Accurate Flux Vector Splitting for Shocks and Shear Layers," *Journal of Computational Physics*, Vol. 121, No. 1, Oct. 1995, pp. 66–78.

[90] Donat, R. and Marquina, A., "Capturing Shock Reflections: An Improved Flux Formula," *Journal of Computational Physics*, Vol. 125, No. 1, Apr. 1996, pp. 42–58.

[91] Liou, M.-S. and Steffen, C. J., "Development of a New Flux Splitting Scheme," *AIAA Tenth Computational Fluid Dynamics Conference*, Jun. 1991, pp. 967–968.

[92] Larrouturou, B., "How to Preserve the Mass Fractions Positivity when Computing Compressible Multi-Component Flows," *Journal of Computational Physics*, Vol. 95, No. 1, Jul. 1991, pp. 59–84.

[93] Gottlieb, J. J. and Groth, C. P. T., "Assessment of Riemann Solvers for Unsteady One-Dimensional Inviscid Flows of Perfect Gases," *Journal of Computational Physics*, Vol. 78, No. 2, Oct. 1988, pp. 437–458.

[94] Quirk, J. J., "A Contribution to the Great Riemann Solver Debate," *International Journal for Numerical Methods in Fluids*, Vol. 18, No. 6, Mar. 1994, pp. 555–574.

[95] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, Oct. 1981, pp. 357–372.

[96] Jacobs, P. A., "Approximate Riemann Solver for Hypervelocity Flows," *AIAA Journal*, Vol. 30, No. 10, Oct. 1992, pp. 2558–2561.

[97] Pullin, D. I., "Direct Simulation Methods for Compressible Inviscid Ideal-Gas Flow," *Journal of Computational Physics*, Vol. 34, No. 2, Feb. 1980, pp. 231–244.

[98] Macrossan, M. N., "The Equilibrium Flux Method for the Calculation of Flows with Non-Equilibrium Chemical Reactions," *Journal of Computational Physics*, Vol. 80, No. 1, Jan. 1989, pp. 204–231.

[99] Xu, K., Martinelli, L., and Jameson, A., "Gas-Kinetic Finite Volume Methods, Flux-Vector Splitting and Artificial Diffusion," *Journal of Computational Physics*, Vol. 120, No. 1, Aug. 1995, pp. 48–65.

[100] Moschetta, J.-M. and Pullin, D. I., "A Robust Low Diffusive Kinetic Scheme for the Navier-Stokes/Euler Equations," *Journal of Computational Physics*, Vol. 133, No. 2, May 1997, pp. 193–204.

[101] Liou, M.-S. and Steffen, C. J., "A New Flux Splitting Scheme," *Journal of Computational Physics*, Vol. 107, No. 1, Jul. 1993, pp. 23–39.

[102] Wada, Y. and Liou, M.-S., "A Flux Splitting Scheme with High-Resolution and Robustness for Discontinuities," AIAA Paper 94-0083, Jan. 1994.

[103] Wada, Y. and Liou, M.-S., "An Accurate and Robust Flux Splitting Scheme for Shock and Contact Discontinuities," *SIAM Journal on Scientific Computing*, Vol. 18, No. 3, May 1997, pp. 633–657.

[104] Liou, M.-S., "A Continuing Search for a Near-Perfect Numerical Flux Scheme," Technical Memorandum 106524, NASA, Mar. 1994.

[105] Kim, K. H., Lee, J. H., and Rho, O. H., "An Improvement of AUSM Schemes by Introducing the Pressure-Based Weight Functions," *Computers and Fluids*, Vol. 27, No. 3, Mar. 1998, pp. 311–346.

[106] Liou, M.-S., "Progress Towards an Improved CFD Method: AUSM+," *AIAA Twelfth Computational Fluid Dynamics Conference*, Jun. 1995, pp. 606–625, AIAA Paper 95-1701.

[107] Liou, M.-S. and Wada, Y., "A Quest Towards Ultimate Numerical Flux Schemes," *Computational Fluid Dynamics Review*, edited by M. Hafez and K. Oshima, Wiley, 1995, pp. 251–278.

[108] Edwards, J. R. and Liou, M.-S., "Low-Diffusion Flux-Splitting Mehtods for Flows at All Speeds," AIAA Paper 97-1862, May 1997.

[109] Smith, A. L., Johnston, I. A., and Austin, K. J., "Comparison of Numerical and Experimental Drag Measurement in Hypervelocity Flow," *The Aeronautical Journal*, Vol. 100, No. 999, Nov. 1996, pp. 385–388.

[110] Gupta, R. N., Lee, K. P., and Scott, C. D., "Aerothermal Study of Mars Pathfinder Aeroshell," *Journal of Spacecraft and Rockets*, Vol. 33, No. 1, Jan.–Feb. 1996, pp. 61–69.

[111] Smith, B. A., "Lockheed Martin Wins Mars Contract," *Aviation Week & Space Technology*, Vol. 142, No. 13, Mar. 1995, pp. 25.

[112] Walberg, G., "How Shall We Go to Mars? A Review of Mission Scenarios," *Journal of Spacecraft and Rockets*, Vol. 30, No. 2, Jul.–Aug. 1993, pp. 129–151.

[113] Johnston, I. A. and McGhee, A., "A CFD Study of Flight in a Martian Atmosphere," *International Aerospace Congress 1995 Proceedings. PICAST 2 – AAC 6*, Vol. 1, The Institution of Engineers, Australia, Melbourne, Mar. 1995, pp. 361–365.

[114] Bashkirov, A. G., "Influence of Viscous Stresses on Shock Wave Stability in Gases," *Physics of Fluids. A: Fluid Dynamics*, Vol. 3, No. 5, May 1991, pp. 960–966.

[115] D'Yakov, S. P., "Investigation of the Problem of the Stability of Shock Waves in Arbitrary Media," *Zhurnal Eksperimental'noi i Teoreticheskoi Fiziki*, Vol. 27, 1954, pp. 288–295.

[116] Swan, G. W. and Fowles, G. R., "Shock Wave Stability," *The Physics of Fluids*, Vol. 18, No. 1, 1975, pp. 28–35.

[117] Barmin, A. A. and Egorushkin, S. A., "Stability of Shock Waves," *Advances in Mechanics*, Vol. 15, No. 1, 1992, pp. 3–37.

[118] Johnston, I. A. and Jacobs, P. A., "Hypersonic Blunt Body Flows in Reacting Carbon Dioxide," *Twelfth Australasian Fluid Mechanics Conference*, edited by R. W. Bilger, Vol. 2, Sydney, Dec. 1995, pp. 807–810.

[119] Griffiths, R. W., Sandeman, R. J., and Hornung, H. G., "The Stability of Shock Waves in Ionizing and Dissociating Gases," *Jounral of Physics. D: Applied Physics*, Vol. 8, 1975.

[120] Paull, A., *A Theoretical and Experimental Investigation into the Test Flow Disturbances in an Expansion Tube*, Master's thesis, Department of Mechanical Engineering, The University of Queensland, 1994.

[121] Glass, I. I. and Liu, W. S., "Effects of Hydrogen Impurities on Shock Structure and Stability in Ionizing Monatomic Gases. Part 1. Argon," *Journal of Fluid Mechanics*, Vol. 84, 1978, pp. 55–77.

[122] Houwing, A. F. P., Fowles, G. R., and Sandeman, R. J., "Shock Wave Instability and Spontaneous Acoustic Emission for Arbitrary Disturbances in Real Gases," *Proceedings of the 14th International Symposium on Shock Tubes and Waves*, edited by R. D. Archer and B. E. Milton, Sydney, Aug. 1983, pp. 277–284.

[123] Houwing, A. F. P., McIntyre, T. J., Taloni, P. A., and Sandeman, R. J., "On the Population of the Metastable States Behind Unstable Shock Waves in Ionizing Argon," *Journal of Fluid Mechanics*, Vol. 170, 1986, pp. 319–337.

[124] Mond, M. and Rutkevich, I. M., "Spontaneous Acoustic Emission from Strong Ionizing Shocks," *Journal of Fluid Mechanics*, Vol. 275, 1994, pp. 121–146.

[125] McIntyre, T. J., Wegener, M. J., Bishop, A. I., and Rubinsztein-Dunlop, H., "Simultaneous Two-Wavelength Holographic Interferometry in a Superorbital Expansion Tube Facility," *Applied Optics*, Vol. 36, No. 31, Nov. 1997, pp. 8128–8134.

[126] Neely, A. J. and Morgan, R. G., "The Superorbital Expansion Tube Concept, Experiment and Analysis," *The Aeronautical Journal*, Vol. 98, No. 973, 1994, pp. 97–105.

[127] Allen, C., "The reduction of numerical entropy generated by unsteady shockwaves," *The Aeronautical Journal*, Vol. 100, No. 1001, Jan. 1997, pp. 9–16.

[128] Roberts, T. W., "The Behaviour of Flux Difference Splitting Schemes near Slowly Moving Shock Waves," *Journal of Computational Physics*, Vol. 90, No. 1, Sep. 1990, pp. 141–160.

[129] Karni, S. and Čanić, S., "Computations of Slowly Moving Shocks," *Journal of Computational Physics*, Vol. 136, No. 1, Sep. 1997, pp. 132–139.

[130] Hayes, W. D., "The Vorticity Jump Across a Gasdynamic Discontinuity," *Journal of Fluid Mechanics*, Vol. 2, Aug. 1957, pp. 595–600.

[131] Truesdell, C., "On Curved Shocks in Steady Plane Flow of an Ideal Fluid," *Journal of the Aeronautical Sciences*, Vol. 19, 1952, pp. 826–828.

[132] Guichard, L., Vervisch, L., and Domingo, P., "Two-Dimensional Weak Shock-Vortex Interaction in a Mixing Zone," *AIAA Journal*, Vol. 33, No. 10, Oct. 1995, pp. 1797–1802.

[133] Rusak, Z., Giddings, T. E., and Cole, J. D., "Interaction of a Weak Shock with Freestream Disturbances," *AIAA Journal*, Vol. 33, No. 6, Jun. 1995, pp. 977–984.

[134] Hornung, H. G., "Shock Layer Instability Near the Newtonian Limit of Hypervelocity Flows," *13th Australasian Fluid Mechanics Conference*, edited by M. C. Thompson and K. Hourigan, Vol. 1, Melbourne, Dec. 1998, pp. 111–118.

[135] Cambier, J.-L., Tokarcik, S., and Prabhu, D. K., "Numerical Simulations of Unsteady Flow in a Hypersonic Shock Tunnel Facility," AIAA Paper 92-4029, 1992.

[136] von Neumann, J. and Richtmyer, R. D., "A Method for the Numerical Calculation of Hydrodynamic Shocks," *Journal of Applied Physics*, Vol. 21, 1950, pp. 232–237.

[137] Caramana, E. J., Shashkov, M. J., and Whalen, P. P., "Formulations of Artificial Viscosity for Multi-dimensional Shock Wave Computations," *Journal of Computational Physics*, Vol. 144, No. 1, Jul. 1998, pp. 70–97.

[138] de Nicola, C., Iaccarino, G., and Tognaccini, R., "Rotating Dissipation for Accurate Shock Capture," *AIAA Journal*, Vol. 34, No. 6, 1996, pp. 1289–1291.

[139] Nasuti, F. and Onofri, M., "Analysis of Unsteady Supersonic Viscous Flows by a Shock Fitting Technique," *AIAA Journal*, Vol. 34, No. 7, Jul. 1996, pp. 1429–1434.

[140] Courant, R. and Friedrichs, K. O., *Supersonic Flow and Shock Waves*, Vol. 1 of *Pure and Applied Mathematics*, Interscience Publishers, New York, 1948.

[141] MacCormack, R. W., "A Perspective on a Quarter Century of CFD Research," AIAA Paper 93-3291-CP, 1993.

[142] Morton, K. W. and Paisley, M. F., "A Finite Volume Scheme with Shock Fitting for the Steady Euler Equations," *Journal of Computational Physics*, Vol. 80, No. 1, Jan. 1989, pp. 168–203.

[143] Salas, M. D., "Shock Fitting Method for Complicated Two-Dimensional Supersonic Flows," *AIAA Journal*, Vol. 14, No. 5, May 1976, pp. 583–588.

[144] Harwich, P. M., "Comparison of Coordinate-Invariant and Coordinate-Aligned Upwinding for the Euler Equations," *AIAA Journal*, Vol. 32, No. 9, Sep. 1994, pp. 1791–1799.

[145] Kim, K. H. and Rho, O. H., "An Accurate Simulation of Hypersonic Flow Using AUSMPW+ Scheme and Shock-Aligned Grid Technique," AIAA Paper 98-2442, 1998.

[146] Trépanier, J.-Y., Paraschivoiu, M., Reggio, M., and Camarero, R., "A Conservative Shock Fitting Method on Unstructured Grids," *Journal of Computational Physics*, Vol. 126, No. 2, Jul. 1996, pp. 421–433.

[147] Moretti, G. and Abbett, M., "A Time-Dependent Computational Method for Blunt Body Flows," *AIAA Journal*, Vol. 4, No. 12, Dec. 1966, pp. 2136–2141.

[148] Moretti, G., "The $\lambda$-scheme," *Computers & Fluids*, Vol. 7, 1979, pp. 191–205.

[149] de Neef, T. and Moretti, G., "Shock Fitting for Everybody," *Computers and Fluids*, Vol. 8, 1980, pp. 327–334.

[150] Moretti, G., "Computation of Flows with Shocks," *Annual Review of Fluid Mechanics*, Vol. 19, 1987, pp. 313–337.

[151] Moretti, G. and Bleich, G., "Three-Dimensional Flow around Blunt Bodies," *AIAA Journal*, Vol. 5, No. 9, Sep. 1967, pp. 1557–1562.

[152] Vos, J. B., Bergman, C. M., and Rizzi, A. W., "Hypersonic Flow Computations with Real Gas Effects and Centred Difference Schemes," *Computational Aeronautical Fluid Dynamics*, edited by L. Fezoui, J. Periaux, and J. C. R. Hunt, Clarendon Press, Oxford, 1994, pp. 51–69.

[153] Sod, G. A., "A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 27, No. 1, Apr. 1978, pp. 1–31.

[154] Oreskes, N., Shrader-Frechette, K., and Belitz, K., "Verification, Validation, and Confirmation of Numerical Models in the Earth Sciences," *Science*, Vol. 263, Feb. 1994, pp. 641–646.

[155] Roache, P. J., "Verification of Codes and Calculations," *AIAA Journal*, Vol. 36, No. 5, May 1998, pp. 696–702.

[156] Mehta, U. B., "Guide to Credible Computer Simulations of Fluid Flows," *Journal of Propulsion and Power*, Vol. 12, No. 5, Sep.–Oct. 1996, pp. 940–948.

[157] Marvin, J. G., "Perspective on Computational Fluid Dynamics Validation," *AIAA Journal*, Vol. 33, No. 10, Oct. 1995, pp. 1778–1787.

[158] Aeschliman, D. P. and Oberkampf, W. L., "Experimental Methodology for Computational Fluid Dynamics Code Validation," *AIAA Journal*, Vol. 36, No. 5, May 1998, pp. 733–741.

[159] Rizzi, A. and Vos, J., "Toward Establishing Credibility in Computational Fluid Dynamics Simulations," *AIAA Journal*, Vol. 36, No. 5, May 1998, pp. 668–675.

[160] Aftosmis, M., Gaitonde, D., and Tavares, T. S., "Behavior of Linear Reconstruction Techniques on Unstructured Meshes," *AIAA Journal*, Vol. 33, No. 11, Nov. 1995, pp. 2038–2049.

[161] Roache, P. J., "Perspective: A Method for Uniform Reporting of Grid Refinement Studies," *Journal of Fluids Engineering*, Vol. 116, Sep. 1994, pp. 405–413.

[162] Hillier, R., "Computation of Shock Wave Diffraction at a Ninety Degrees Convex Edge," *Shock Waves*, Vol. 1, No. 2, 1991, pp. 89–98.

[163] Colella, P. and Glaz, H. M., "Efficient Solution Algorithms for the Riemann Problem for Real Gases," *Journal of Computational Physics*, Vol. 59, No. 1, May 1985, pp. 264–289.

[164] Grossman, B. and Cinnella, P., "Flux-Split Algorithms for Flows with Nonequilibrium Chemistry and Vibrational Relaxation," *Journal of Computational Physics*, Vol. 88, No. 1, May 1990, pp. 131–168.

[165] Korte, J. J., *An Explicit, Upwind Algorithm for Solving the Parabolized Navier-Stokes Equations*, Ph.D. thesis, North Carolina State University, 1989.

[166] Pruett, C. D. and Streett, C. L., "A Spectral Collocation Method for Compressible, Nonsimilar Boundary Layers," *International Journal for Numerical Methods in Fluids*, Vol. 13, No. 6, Sep. 1991, pp. 713–737.

[167] Smith, A. L. and Mee, D. J., "Drag Measurements in a Hypervelocity Expansion Tube," *Shock Waves*, Vol. 6, No. 3, Sep. 1996, pp. 161–166.

[168] Taylor, G. I. and Maccoll, J. W., "The Air Pressure on a Cone Moving at High Speed," *Proceedings of the Royal Society of London Series A: Mathematical and Physical Sciences*, Vol. 139, 1932, pp. 278–311.

[169] Gnoffo, P. A., Weilmuenster, K. J., Braun, R. D., and Cruz, C. I., "Effects of Sonic Line Transition on Aerothermodynamics of the Mars Pathfinder Probe," AIAA Paper 95-1825, 1995.

[170] Merzkirch, W., *Flow Visualization*, Academic Press, Orlando, 2nd ed., 1987.

[171] Ambrosio, A. and Wortman, A., "Stagnation Point Shock Detachment Distance for Flow around Spheres and Cylinders," *ARS Journal*, Vol. 32, 1962, pp. 281.

[172] Billig, F. S., "Shock-Wave Shapes around Spherical and Cylindrical-Nosed Bodies," *Journal of Spacecraft*, Vol. 4, No. 6, Jun. 1967, pp. 822–823.

[173] Anderson, J. D., *Modern Compressible Flow with Historical Perspective*, McGraw-Hill, New York, 2nd ed., 1990.

[174] Hornung, H. G., "Non-Equilibrium Dissociating Nitrogen Flow Over Spheres and Circular Cylinders," *Journal of Fluid Mechanics*, Vol. 53, 1972, pp. 149–176.

[175] Hornung, H. G., "Non-Equilibrium Ideal-Gas Dissociation After a Curved Shock Wave," *Journal of Fluid Mechanics*, Vol. 74, 1976, pp. 143–159.

[176] Stalker, R. J., "Development of a Hypervelocity Wind Tunnel," *The Aeronautical Journal*, Vol. 76, No. 738, Jun. 1972, pp. 374–384.

[177] Macrossan, M. N., "Hypervelocity Flow of Dissociating Nitrogen Downstream of a Blunt Nose," *Journal of Fluid Mechanics*, Vol. 217, 1990, pp. 167–202.

[178] Inouye, Y., Watanabe, S., Fujii, K., Ohtake, K., Takaki, R., Takizawa, M., Ito, T., Shirouzu, M., Kai, T., and Teraoka, K., "Quick Report of HYFLEX Onboard Measurements," 20th International Symposium on Space Technology and Science, Gifu, Japan, 1996.

[179] Hornung, H. G., "28th Lanchester Memorial Lecture – Experimental Real-Gas Hypersonics," *The Aeronautical Journal*, Vol. 92, No. 920, 1988, pp. 379–389.

[180] Fay, J. A., *Molecular Thermodynamics*, Addison-Wesley, Reading, 1965.

[181] Hansen, C. F. and Heims, S. P., "A Review of the Thermodynamic, Transport, and Chemical Reaction Rate Properties of High-Temperature Air," NACA TN-4359, 1958.

[182] Hartung, L. C. and Throckmorton, D. A., "Computer Graphic Visualization of Orbiter Lower Surface Boundary-Layer Transition," *Journal of Spacecraft and Rockets*, Vol. 24, No. 2, 1987, pp. 109–114; also AIAA Paper 84–0228, Jan. 1984.

[183] Yamamoto, Y. and Yoshioka, M., "HYFLEX Computational Fluid Dynamics Analysis," 19th International Symposium on Space Technology and Science, Yokohama, Japan, 1994.

[184] Ma, L., Ingham, D. B., and Wen, X., "A Finite Volume Method for Fluid Flow in Polar Cylindrical Grids," *International Journal for Numerical Methods in Fluids*, Vol. 28, 1998, pp. 663–677.

[185] Richardson, L. F., "The Deferred Approach to the Limit," *Transactions of the Royal Society of London, Series A*, Vol. 226, 1927, pp. 299–361.

[186] Tuttle, S. L. and Shimoda, T., "A Report on the T4 HYFLEX Experiment," Research Report 9/97, Department of Mechanical Engineering, The University of Queensland, 1997.

[187] Mee, D. J., "Uncertainty Analysis of Conditions in the Test Section of the T4 Shock Tunnel," Tech. Rep. 4/93, Department of Mechanical Engineering, The University of Queensland, 1993.

[188] Hedin, A. E., "Extension of the MSIS Thermospheric Model into the Middle and Lower Atmosphere," *Journal of Geophysical Research*, Vol. 96, No. A2, Feb. 1991, pp. 1159–1172.

[189] Hedin, A. E., Fleming, E. L., Manson, A. H., Schmidlin, F. J., Avery, S. K., Clark, R. R., Franke, S. J., Fraser, G. J., Tsuda, T., Vial, F., and Vincent, R. A., "Empirical Wind Model for the Upper, Middle, and Lower Atmosphere," *Journal of Atmospheric and Terrestrial Physics*, Vol. 58, No. 13, 1996, pp. 1421–1447.

[190] Benser, E. T., "Air Data Measurement for Hypersonic Vehicles," SAE Technical Paper Series, Paper No. 912143, Sep. 1991.

[191] Gracey, W. H., "Measurement of Aircraft Speed and Altitude," NASA RP-1046, 1980.

[192] Whitmore, S. A., Davis, R. J., and Fife, J. M., "In-Flight Demonstration of a Real-Time Flush Airdata Sensing (RT-FADS) System," NASA TM-104314, Oct. 1995.

[193] Whitmore, S. A. and Moes, T. R., "Measurement Uncertainty and Feasibility Study of a Flush Airdata System for a Hypersonic Flight Experiment," NASA TM-4627, Jun. 1994.

[194] Takaki, R. and Takizawa, M., "ADS Measurement of HYFLEX (Hypersonic Flight Experiment)," AIAA Paper 97-0193, 1997.

[195] Pruett, C., Wolf, H., Heck, M., and Siemers, P., "An Innovative Air Data System for the Space Shuttle Orbiter: Data Analysis Techniques," AIAA Paper 81-2455, Nov. 1981.

[196] Wolf, H., Henry, M. W., and Siemers, P. M., "Shuttle Entry Air Data System (SEADS): Optimization of Preflight Algorithms Based on Flight Results," AIAA Paper 88-2053, May 1988.

[197] Ishimoto, S., Takizawa, M., Suzuki, H., and Morito, T., "Flight Control Design of Hypersonic Flight Experiment (HYFLEX) Vehicle," 20th International Symposium on Space Technology and Science, Paper 96-f-07, May 1996.

[198] Deshpande, S. M., Kumar, R. R., Seywald, H., and Siemers, P. M., "Air Data System Optimization using a Genetic Algorithm," AIAA Paper 92-4466-CP, 1992.

[199] Johnston, R. L., *Numerical Methods*, chap. 2, John Wiley & Sons, New York, 1982, pp. 76–86.

[200] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, 2nd ed., 1992.

[201] Atchison, T. D., "Analysis of a Flush Airdata System Using Neural Networks," Honours thesis, The Univeristy of Queensland, 1997.

[202] Pap, R., Saeks, R., Lewis, C., Carlton, A., Kocher, M., and Kersting, T., "The LoFLYTE Program," AIAA Paper 98-1519, 1998.

[203] Rohloff, T. J. and Catton, I., "Development of a Neural Network Flush Airdata Sensing System," *Proceedings of the ASME Fluids Engineering Division*, Vol. 242, 1996, pp. 39–43.

[204] Rohloff, T. J., Whitmore, S. A., and Catton, I., "Air Data Sensing from Surface Pressure Measurements Using a Neural Network Method," *AIAA Journal*, Vol. 36, No. 11, Nov. 1998, pp. 2094–2101.

[205] Gupta, R. N., Lee, K. P., Moss, J. N., and Sutton, K., "Viscous-Shock-Layer Solutions with Coupled Radiation and Ablation Injection for Earth Entry," AIAA Paper 90-1697, 1990.

[206] Gupta, R. N., Lee, K. P., Moss, J. N., and Sutton, K., "A Viscous-Shock-Layer Analysis of the Martian Aerothermal Environment," AIAA Paper 91-1345, Jun. 1991.

[207] Rastogi, P. K., editor, *Holographic Interferometry*, Vol. 68 of *Springer Series in Optical Sciences*, Springer-Verlag, Berlin, 1994.

[208] Rock, S. G., Candler, G. V., and Hornung, H. G., "Analysis of Thermochemical Nonequilibrium Models for Carbon Dioxide Flows," *AIAA Journal*, Vol. 31, No. 12, 1993, pp. 2255–2262.

[209] Kernighan, B. W. and Ritchie, D. M., *The C Programming Language*, Prentice Hall, Englewood Cliffs, 2nd ed., 1988.

[210] Anonymous, *Tecplot User's Manual, Version 7*, Amtec Engineering, Inc., Bellevue, Aug. 1996.

[211] Anonymous, *Tecplot Reference Manual, Version 7*, Amtec Engineering, Inc., Bellevue, 1996.

[212] Wendt, J. F., editor, *Computational Fluid Dynamics: An Introduction*, Springer, Berlin, 2nd ed., 1996.