



THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

**ADJOINT-BASED AERODYNAMIC DESIGN OPTIMISATION
IN HYPERSONIC FLOW**

Kyle A. Damm

B.Eng. (Mechanical and Aerospace) (Hons. I) UQ

A thesis submitted for the degree of Doctor of Philosophy at
The University of Queensland in 2019

School of Mechanical and Mining Engineering
Centre for Hypersonics

Abstract

The drive for improved flexibility and reusability in satellite launch systems has brought a resurgence in the popularity of scramjet-powered access-to-space launch vehicle concepts. A challenge in scramjet-powered accelerator vehicles is achieving positive thrust margins at the high-speed end of their flight envelopes. As a consequence, scramjet-powered vehicles typically rely on highly integrated airframe-engine configurations. However, due to the tight integration, the geometries and the flow physics are complex. An automated optimisation method seems an excellent candidate approach to explore the complex design space of hypersonic vehicles. This thesis focuses on the development and application of a particular optimisation method, adjoint-based optimisation, to aid in efficient aerodynamic design in hypersonic flows.

Shape optimisation of scramjet-powered vehicles requires many design parameters to capture the geometric detail, and, since the flow physics is complex, the fidelity of Reynolds-Averaged Navier-Stokes (RANS) analyses is desirable. The combination of many design parameters and an expensive objective function evaluation drives the need for gradient evaluation methods that scale well with the number of design variables. An advantage of the adjoint method is that all shape sensitivities for an objective function are evaluated at the cost of only one flow solution and one adjoint solution. As a result of its efficiency, the adjoint method has become widely used for aircraft design, evolving to the design optimisation of full configurations. Despite the wide use of adjoint methods in aircraft optimisation, there has been very little application to hypersonic vehicle design. Several high-speed adjoint solvers have been reported in the literature, however, a majority of these works have only verified the adjoint sensitivities, few have followed on to demonstrate the method for use in design optimisation. The contribution of this work is the description and application of a discrete adjoint solver in high-speed compressible flow optimisation. What is unique in this work is a demonstration that complex-step differentiation works well to linearise a second-order spatially accurate unstructured RANS solver. In particular, the approach presented in this work utilises the $k - \omega$ turbulence model in high-speed ducted flow configurations.

As part of this work, to provide flow analysis with a rapid turn-around, an unstructured steady-state RANS solver driven by a Jacobian-Free Newton-Krylov method was developed. Turbulence is modelled using the two-equation $k - \omega$ turbulence model. The Newton method is globalised by using the pseudo-transient approach. A restarted GMRES method is used to solve the system of linear equations arising when solving for the Newton steps. Evaluation of the matrix-vector products required in the GMRES algorithm is accomplished by Fréchet derivatives using imaginary perturbations in the complex plane. This is necessary to achieve robust convergence of the types of turbulent hypersonic flows considered in this work. Equation scaling ensures that the linear solver provides an adequate solution of the linear system, especially for turbulent flows, where the flow and turbulence variables differ by several orders of magnitude. Incomplete lower-upper preconditioning with zero-fill is used

to accelerate linear system convergence. To achieve adequate residual convergence for flows with embedded shocks, limiter freezing is required to prevent early stall. The developed flow solver is verified using several methods, including the Method of Manufactured Solutions. Several validation cases from the literature are presented to establish the appropriateness of the implemented physical models for design analysis in high-speed flow.

An accompanying discrete adjoint solver was developed to provide efficient computation of the required shape sensitivities. The primary complication of linearising the flow solver routines is handled via a complex-step derivative approach. Targeted differentiation is employed to provide an efficient means of constructing the adjoint operator. The adjoint gradients are verified against a complex variable direct-differentiation method.

The flow and adjoint solvers are coupled to the open-source optimisation library, DAKOTA, to perform design optimisation in high-speed flows. Design surfaces are parameterised using Bézier curves and mesh deformation is achieved by the inverse distance weighting method. Two optimisation applications are presented here as a demonstration of the development work: (a) wave drag minimisation of an axisymmetric body; and (b) hypersonic inlet design optimisation. For the case of axisymmetric bodies, the optimal shape determined by this work compares favourably to several minimum-drag power-law bodies published in the literature. The second application is the redesign of the P2 hypersonic inlet. The chosen objective function aimed at removing the reflected cowl shock whilst obtaining the desired compression ratio. The results presented show that the optimiser has removed the reflected shock while achieving the desired compression ratio, at no cost to the inlet performance metrics.

The conclusions are that the developed discrete adjoint-based optimisation framework does work well in a hypersonic flow context and that the use of complex-step differentiation is a key enabler in the implementation. In particular, the inlet example presented in this work demonstrates the efficiency, accuracy, and applicability of discrete adjoint-based optimisation to design analysis in turbulent high-speed flow.

Declaration by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, financial support and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my higher degree by research candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the policy and procedures of The University of Queensland, the thesis be made available for research and study in accordance with the Copyright Act 1968 unless a period of embargo has been approved by the Dean of the Graduate School.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis and have sought permission from co-authors for any jointly authored works included in the thesis.

Publications during candidature

Peer-reviewed Journal Papers

K.A. Damm, R.J. Gollan, P.A. Jacobs, M.K. Smart, S. Lee, E. Kim, and C. Kim. “Discrete adjoint optimization of a hypersonic inlet”. In: *AIAA Journal* (2019)

Conference Papers

K.A. Damm, R.J. Gollan, and A. Veeraragavan. “Acceleration of combustion simulations using GPUs”. In: *Australian Combustion Symposium*. Melbourne, Australia, 2015, pp. 148–151

K.A. Damm, R.J. Gollan, and A. Veeraragavan. “On the effect of workload ordering for reacting flow simulations using GPUs”. In: *20th Australasian Fluid Mechanics Conference*. Paper Number: 748. Perth, Australia, 2016, pp. 748.1–748.4

K.A. Damm, R.J. Gollan, and P.A. Jacobs. “Verification of the least-squares procedure within an unstructured-grid flow solver”. In: *20th Australasian Fluid Mechanics Conference*. Paper Number: 703. Perth, Australia, 2016, pp. 703.1–703.4

K.A. Damm, R.J. Gollan, P.A. Jacobs, S. Lee, E. Kim, and C. Kim. “Adjoint design optimization for the P2 hypersonic inlet”. In: *Korean Society of Computational Fluid Engineering Conference*. Jeju, South Korea, 2018

Conference Posters

K.A. Damm, R.J. Gollan, P.A. Jacobs, S. Lee, E. Kim, and C. Kim. “Adjoint design optimization for the P2 hypersonic inlet”. In: *10th National Congress on Fluids Engineering Conference*. Yeosu, South Korea, 2018

Technical Reports

K. A. Damm. *Shock Fitting mode for Eilmer*. Mechanical Engineering Report 2016/15. Centre for Hypersonics, University of Queensland, 2016

Publications included in this thesis

None.

Contributions by Others to the Thesis

There are no contributions by others to this thesis.

Statement of Parts of the Thesis Submitted to Qualify for the Award of Another Degree

None.

Research Involving Human or Animal Subjects

No animal or human subjects were involved in this research.

Acknowledgements

First and foremost I would like to express my sincerest gratitude to my supervisory team. To my primary supervisor, Dr. Rowan Gollan, thank you for taking me on as your first Ph.D. student. Your patience, foresight, and your knowledge and passion for CFD have made you not only a great academic supervisor but also an exceptional personal mentor. Many thanks go to my associate supervisor, Dr. Peter Jacobs. Without your vision for Eilmer, this work surely would have not been possible. The breadth of your knowledge on all things related to computers and engineering never ceases to amaze me. Although I never quite managed to grow a beard that would make you both proud, I hope I more than made up for it with my academic efforts. Finally, thanks must also go to my associate supervisor, Prof. Michael Smart. Your passion for the development of SPARTAN has been an inspiration. To everyone else in the Centre for Hypersonics, thank you for making my time in the lab such an enjoyable and memorable experience.

To Prof. Chongam Kim, thank you for initiating the opportunity for an extended research visit to the Aerodynamic Simulation and Design Laboratory (ASDL). I am forever grateful for the knowledge and experience I gained from my time at the ASDL. And to all the members of the ASDL, thank you for welcoming me into your lab, and making my time in South Korea a most enjoyable one. My sincere thanks also go to Prof. Gisu Park and the members of the Hypersonics Lab at KAIST, for providing such a welcoming start to my research life in South Korea.

I would certainly be remiss if I did not also thank all of the support I have been given outside of my academic life. So, to all my friends in South Korea, who gave me a reason to leave the lab once in a while, 정말 고맙습니다! And an emphatic thank you to all my friends in Australia, who did the same. Finally, I wish to thank my family for the continuing support in this seemingly never-ending path. To my amazing mother, father, brother, and little nephew, Rupert. I could not have done this without you.

The true delight is in the finding out rather than in the knowing.

— Isaac Asimov

Financial support

This research was supported by an Australian Government Research Training Program (RTP) Scholarship and the Cooperative Research Centre for Space Environment Management (SERC Limited) through the Australian Governments Cooperative Research Centre Program.

Keywords

adjoint optimisation, computational fluid dynamics, hypersonic aerodynamics

Australian and New Zealand Standard Research Classification (ANZSRC)

ANZSRC code: 091501 Computational Fluid Dynamics, 50%

ANZSRC code: 010303 Optimisation, 35%

ANZSRC code: 090107 Hypersonic Propulsion and Hypersonic Aerodynamics, 15%

Fields of Research (FoR) Classification

FoR code: 0901, Aerospace Engineering, 100%

CONTENTS

Contents	xi
List of figures	xv
List of tables	xix
Nomenclature	xxi
1 Introduction	1
1.1 Motivation and research context	1
1.1.1 Scramjet-assisted access-to-space	1
1.1.2 Hypersonic vehicle design challenges	3
1.1.3 Modern CFD-based design optimisation	6
1.2 Thesis aims	7
1.3 Thesis outline	8
2 Background for Hypersonic Vehicle Design	11
2.1 Airbreathing hypersonic vehicles	11
2.2 Relevant flow phenomena	12
2.3 Existing design methodology	14
3 Review of Adjoint-Based Aerodynamic Shape Optimisation	19
3.1 Computational Fluid Dynamics	19
3.2 Numerical optimisation	20
3.3 The adjoint method	22
3.4 Construction of the adjoint operator	25

3.4.1	Analytic differentiation	25
3.4.2	Finite differences	25
3.4.3	Complex-Step derivative approximation	26
3.4.4	Algorithmic differentiation	26
3.5	Hypersonic applications	27
3.5.1	Continuous adjoint method	27
3.5.2	Discrete adjoint method	27
3.5.3	Context for contributions in this work	28
4	Flow Solver	31
4.1	Governing equations	31
4.1.1	Turbulence modelling	31
4.1.2	The RANS flow equations	32
4.2	Discretisation of the flow equations	34
4.2.1	Finite Volume Method	34
4.2.2	Gradient reconstruction	35
4.2.3	Convective flux	36
4.2.3.1	Higher-order reconstruction	37
4.2.3.2	Gradient limiting	37
4.2.3.3	Carbuncle Fix	38
4.2.4	Viscous flux	39
4.3	Temporal discretisation	40
4.3.1	A globalised Newton-Krylov method	40
4.3.2	Linear system scaling	41
4.3.3	Matrix-free approach	42
4.3.4	Preconditioning	43
4.3.5	GMRES algorithm	45
4.4	Boundary Conditions	47
4.5	Parallelisation	48
4.6	Summary	49
5	Flow Solver Verification and Validation	51
5.1	Overview of flow solver settings	51
5.2	Verification	52
5.2.1	Method of Manufactured Solutions	52
5.2.1.1	Baseline Manufactured Solutions	53
5.2.1.2	Two-dimensional flow solver	53
5.2.1.3	Three-dimensional flow solver	56

5.2.2	Shock-capturing capability	58
5.2.3	Self-similar laminar boundary layers	62
5.2.4	Summary	64
5.3	Validation	65
5.3.1	Turbulent flat plate	65
5.3.2	Turbulent flat plate with impinging shock	70
5.3.3	Sharp nose double cone	74
5.3.4	24° compression-corner	78
5.3.5	JCEAP	82
5.3.6	Summary	87
6	Adjoint Solver	89
6.1	Discrete adjoint equations	89
6.2	Constructing the discrete adjoint operator	90
6.2.1	Parallelisation	90
6.2.2	Boundary conditions	91
6.2.3	Frozen limiter	92
6.3	Numerical solution of the adjoint equations	93
6.4	Total derivative calculation	94
6.5	Verification	95
6.5.1	Verification procedure	95
6.5.2	Verification test case	96
6.5.3	Debugging the adjoint operator	97
7	Shape Optimisation Methodology	101
7.1	Surface parameterisation	101
7.1.1	A brief review of surface parameterisation methods	101
7.1.2	Bézier curves	102
7.2	Mesh generation and movement strategy	103
7.2.1	A brief review of mesh deformation methods	103
7.2.2	Inverse Distance Weighting method	104
7.3	Numerical optimiser	106
7.4	Design optimisation algorithm	106
8	Optimisation Applications	109
8.1	Selection of grid topology	109
8.2	Hypersonic minimum-drag slender bodies of revolution	110
8.2.1	Optimisation problem description	111

CONTENTS

8.2.2	Adjoint solver convergence characteristics	113
8.2.3	Results	115
8.3	Redesign of the NASA P2 hypersonic inlet	118
8.3.1	Background for hypersonic inlet design	118
8.3.2	P2 hypersonic inlet	118
8.3.3	Baseline flowfield	119
8.3.4	Optimisation problem description	123
8.3.5	Results	125
8.4	Summary	129
9	Conclusions	131
9.1	Summary and discussion	131
9.2	Contributions	134
9.3	Recommendations for future work	134
	References	137
A	Method of Manufactured Solutions Constants	159
A.1	2D Euler Equations	159
A.2	2D Navier-Stokes Equations	160
A.3	2D RANS Equations	161
A.4	3D Euler Equations	162
A.5	3D Navier-Stokes Equations	163
A.6	3D RANS Equations	164
B	Additional Flow Solver Test Cases	165
B.1	Supersonic ramp	165
B.2	Mach 10 blunted wedge	167
B.3	Mach 8 cylinder	168
C	Derivation of Transform Jacobian	169
D	Minimum-drag slender body of revolution baseline and optimised geometries	171
E	P2 inlet baseline and optimised geometries	175

LIST OF FIGURES

1.1	Scramjet-rocket launch system mission profile.	2
1.2	Schematic of a scramjet engine.	3
1.3	Integrated vehicle design for SPARTAN hypersonic accelerator.	3
1.4	Airbreathing hypersonic vehicle schematic.	4
1.5	CART3D simulation of SPARTAN at Mach 8 flight conditions.	6
2.1	Shock wave/viscous interactions on a generic hypersonic vehicle.	14
2.2	Boeing’s state of the art MDO framework for a hypersonic vehicle.	17
3.1	Computational fluid models	20
3.2	Single parameter optimisation.	21
4.1	Quadrilateral finite volume cell with geometric definitions.	35
4.2	Least-squares point cloud stencil.	35
4.3	Global residual history for simulation with frozen limiter on 3000th iteration.	38
4.4	Carbuncle fix example.	39
4.5	Effect of h selection for a supersonic ramp simulation.	43
4.6	Comparison of level of fill for a supersonic ramp simulation.	45
4.7	A scaled right-preconditioned restarted GMRES algorithm.	47
5.1	Example grid for two-dimensional MMS simulations.	54
5.2	Two-dimensional MMS results	55
5.3	Three-dimensional MMS results	57
5.4	Oblique-shock simulation schematic.	58
5.5	Oblique shock numerical solution colour map of Mach number and entropy change.	60
5.6	Pressure distribution along a streamline	61

LIST OF FIGURES

5.7	Laminar flat plate simulation schematic.	62
5.8	Laminar flat plate simulation, skin friction coefficient.	63
5.9	Laminar boundary layer profiles.	64
5.10	Turbulent flat plate simulation schematic.	66
5.11	Turbulent flat plate simulation, skin friction coefficient	67
5.12	Turbulent boundary layer profiles at $x = 0.368$ m	68
5.13	Comparison of boundary layer profiles at $x = 0.368$ m and $x = 0.45$ m	69
5.14	Shock wave turbulent boundary layer interaction simulation schematic (units in meters).	70
5.15	Shock wave turbulent boundary layer interaction flow solution.	72
5.16	Calculated distributions for shock wave turbulent boundary layer interaction.	73
5.17	Sharp nose double cone simulation schematic (units in meters).	75
5.18	Sharp nose double cone simulation flow features.	76
5.19	Calculated distributions double cone.	77
5.20	24° compression-corner simulation schematic	79
5.21	Colour map of pressure field for 24° compression corner.	80
5.22	Calculated distributions for compression-corner	81
5.23	JCEAP simulation details.	83
5.24	Surface pressure distribution for axisymmetric simulations.	85
5.25	Colour map of pressure for JCEAP 10° flap configuration at -5° angle of attack.	85
5.26	Axial force coefficients for JCEAP model	86
6.1	Residual stencils	90
6.2	Domain decomposed Jacobian	91
6.3	Schematic of flat-plate wedge geometry with Bézier points superimposed on wedge.	97
6.4	Matrix-vector product error vector (ϵ) for nominal case.	99
6.5	Matrix-vector product error vector (ϵ) for case with implemented bug.	99
7.1	Cubic Bézier curve.	103
7.2	Example of mesh deformation via IDW.	105
7.3	Design optimisation algorithm.	107
8.1	Schematic of conical shock simulation.	112
8.2	Baseline mesh for minimum-drag optimisation.	112
8.3	Influence of convergence levels on sensitivity estimates for two selected variables.	114
8.4	Objective function history.	116
8.5	Minimum-drag bodies.	117
8.6	Schematic of the P2 Hypersonic Inlet.	119
8.7	P2 inlet grid level G2.	121

8.8	Baseline P2 inlet pressure colour map and contour lines.	121
8.9	Simulation S3 Pitot pressure (P_P)	122
8.10	Simulation S3 on level G3 Pitot pressure (P_P)	122
8.11	P2 inlet objective function integral path (design surface shown as dashed line).	124
8.12	Objective function history.	124
8.13	Two numerical solutions	126
8.14	Comparison of Pitot pressure (P_P).	127
8.15	Comparison of pressure (p).	127
8.16	Comparison of baseline P2 inlet and optimised Bézier curves.	128
8.17	Colour map of an adjoint variable.	128
B.1	Supersonic ramp simulation schematic.	166
B.2	Colour map of pressure for supersonic ramp simulation.	166
B.3	Blunted wedge simulation schematic.	167
B.4	Colour map of pressure for blunted-wedge simulation.	167
B.5	Cylinder simulation	168
D.1	Minimum-drag slender body of revolution.	174
E.1	P2 inlet Bézier parameterisation (a) 11-point curve (b) 20-point curve.	177

LIST OF TABLES

1.1	Sample results for Equation 1.3 with varying Mach number.	5
3.1	Summary of published works.	30
5.1	General flow solver settings for verification and validation cases.	52
5.2	Levels of grid refinement used for two-dimensional verification.	54
5.3	Levels of grid refinement used for three-dimensional verification.	56
5.4	Levels of grid refinement for oblique shock verification.	59
5.5	Freestream inflow conditions for oblique shock verification.	59
5.6	Post-shock Mach number and shock angle for oblique shock simulation.	60
5.7	Levels of grid refinement for laminar flat plate verification.	63
5.8	Freestream inflow conditions for laminar flat plate verification.	63
5.9	Levels of grid refinement for turbulent flat plate validation case.	66
5.10	Freestream inflow conditions for turbulent flat plate validation case.	66
5.11	Levels of grid refinement for shock wave turbulent boundary layer interaction validation case.	71
5.12	Freestream inflow conditions for shock wave turbulent boundary layer interaction validation case.	71
5.13	Levels of grid refinement for sharp nose double cone validation case.	74
5.14	Freestream inflow conditions for sharp nose double cone validation case.	75
5.15	Levels of grid refinement for 24° compression-corner validation case.	78
5.16	Freestream inflow conditions for 24° compression-corner validation case.	79
5.17	Levels of grid refinement for JCEAP validation case.	84
5.18	Freestream inflow conditions for JCEAP validation case.	84

LIST OF TABLES

6.1	Sensitivities of the inviscid drag component acting on the wedge section. SB = Singleblock; 4MB = 4 blocks; 8MB = 8 blocks.	98
8.1	Levels of grid refinement and calculated drag for baseline cone simulation.	112
8.2	Freestream inflow conditions for baseline cone simulation.	113
8.3	Freestream inflow conditions for P2 inlet.	120
8.4	Testing of freestream turbulence properties for P2 inlet.	120
8.5	Levels of grid refinement used for validation.	120
8.6	Performance metrics at respective throat locations.	129
A.1	Constants for 2D supersonic Euler manufactured solution.	159
A.2	$f_s(\cdot)$ functions for 2D Euler and 2D subsonic Navier-Stokes manufactured solution.	159
A.3	Constants for 2D subsonic Navier-Stokes manufactured solution.	160
A.4	$f_s(\cdot)$ functions for 2D Euler and 2D subsonic Navier-Stokes manufactured solution.	160
A.5	Constants for 2D $k - \omega$ Reynolds-Averaged Navier-Stokes manufactured solution.	161
A.6	$f_s(\cdot)$ functions for 2D RANS manufactured solution.	161
A.7	Constants for 3D supersonic Euler manufactured solution.	162
A.8	$f_s(\cdot)$ functions for 3D Euler manufactured solution.	162
A.9	Constants for 3D subsonic Navier-Stokes manufactured solution.	163
A.10	$f_s(\cdot)$ functions for 3D subsonic Navier-Stokes manufactured solution.	163
A.11	Constants for 3D subsonic Navier-Stokes manufactured solution.	164
A.12	$f_s(\cdot)$ functions for 3D subsonic Navier-Stokes manufactured solution.	164
B.1	Freestream inflow conditions for supersonic ramp simulation.	165
B.2	Freestream inflow conditions for blunted wedge simulation.	167
B.3	Freestream inflow conditions for cylinder simulation.	168
D.1	20-point Bézier coordinates for minimum-drag slender body of revolution (in meters).	172
D.2	40-point Bézier coordinates for minimum-drag slender body of revolution (in meters).	173
E.1	11-point Bézier coordinates for P2 inlet (in meters).	175
E.2	20-point Bézier coordinates for P2 inlet (in meters).	176

NOMENCLATURE

The nomenclature used in this thesis is outlined here. Some symbols which appear in this list have more than one meaning. The specific meaning of the symbol should be clear in the document from the context.

Acronyms and Initialisms

2D	= Two-dimensional
3D	= Three-dimensional
AD	= Algorithmic Differentiation
AUSMDV	= Advection Upwind Splitting Method (flux-Difference-splitting and flux-Vector-splitting)
B.C.	= Boundary Condition
BFILU	= Block-Fill Incomplete Lower Upper
CFD	= Computational Fluid Dynamics
DNS	= Direct Numerical Simulation
FDS	= Flux Difference Splitting
FVS	= Flux Vector Splitting
IDW	= Inverse Distance Weighting
ILU	= Incomplete Lower Upper
LES	= Large-Eddy Simulation

NOMENCLATURE

MDO	= Multidisciplinary Design Optimisation
MHD	= Magnetohydrodynamics
NACA	= National Advisory Committee for Aeronautics
NASA	= National Aeronautics and Space Administration
PDE	= Partial Differential Equation
RANS	= Reynolds-Averaged Navier-Stokes

Roman Symbols - Upper Case

A	= area
C_D	= coefficient of drag
C_f	= coefficient of friction
C_{lim}	= stress-limiter coefficient
\mathbf{D}	= vector of design variables = linear system scaling matrix
E	= total energy
F	= flux through an interface
F_c	= convective flux through an interface
F_D	= drag force
F_T	= thrust force
F_v	= viscous flux through an interface
I	= turbulence intensity
\mathbf{I}	= identity matrix
I_{sp}	= specific impulse
J	= objective function = Jacobian matrix

K	= Venkatakrisshnan limiter factor
L	= Lagrangian function
	= reference length
M	= Mach number
\mathbf{P}	= precondition matrix
$P_{t,\infty}$	= freestream stagnation pressure
P_p	= Pitot pressure
Pr	= Prandtl number
\mathbf{Q}	= vector of primitive flow state variables
\mathbf{R}	= residual vector for control volumes
R	= gas constant
Re	= Reynolds number
\bar{S}_{ij}	= zero-trace Favre-averaged strain-rate tensor
\hat{S}_{ij}	= Galilean-invariant Favre-averaged strain-rate tensor
S_{ij}	= mean strain-rate tensor
S	= least-squares error term
S_{turb}	= turbulence model source terms
T	= temperature
T_{ref}	= reference temperature for Sutherland semi-empirical formulae
\mathbf{U}	= vector of cell-averaged conserved flow state variables
V	= volume
Roman Symbols - Lower Case	
a	= speed of sound
d	= dimension

NOMENCLATURE

e	= internal energy
f	= discretised solution
f_β	= vortex-stretching function
f_{ref}	= analytical manufactured solution
g	= acceleration due to Earth's gravity (9.81 m/s ²)
h	= perturbation parameter
	= static enthalpy
i	= $\sqrt{-1}$
k	= thermal conductivity coefficient
	= turbulent kinetic energy
\hat{n}	= unit normal vector
p	= pressure
	= cell-center/vertex coordinates
\mathbf{q}	= vector of ghost cell primitive flow state variables
r	= refinement level
\bar{t}_{ij}	= viscous stress tensor
\mathbf{v}	= directional vector
t	= time
u, v, w	= Cartesian components of velocity
w	= least-squares weighting
x, y, z	= Cartesian coordinates

Greek Symbols

α	= closure coefficient in the $k - \omega$ model
β, β^*, β_0	= closure coefficients in the $k - \omega$ model

χ_ω	= vortex-stretching parameter
δ_{ij}	= Kronecker delta
ε	= L_2 error norm
η	= efficiency
γ	= ratio of specific heats
λ	= adjoint variable
μ	= dynamic viscosity coefficient
ω	= specific dissipation rate
Ω_{ij}, Ω_{jk}	= rotation tensors
ρ	= density
$\sigma, \sigma^*, \sigma_{d_0}$	= closure coefficients in the $k - \omega$ model
σ_d	= cross-diffusion term
τ	= shear stress
τ_{ij}	= specific Reynolds stress tensor
ϕ	= general solution variable
	= limiting factor
	= IDW weighting function

Superscripts

'	= fluctuations from the mean value
-	= Time-averaged value
~	= Favre-averaged value
T	= transpose

Subscripts

0	= total flow condition
---	------------------------

NOMENCLATURE

∞	= freestream conditions
lam	= laminar
turb	= turbulent
<i>ad</i>	= adiabatic
<i>e</i>	= boundary layer edge properties
<i>f</i>	= interface mid-point location
<i>k</i>	= verification mesh level ($k = 1, 2, 3, \text{etc.}$, fine to coarse)
<i>KE</i>	= kinetic energy
<i>T</i>	= turbulent term
<i>w</i>	= wall properties

CHAPTER 1

INTRODUCTION

A hypersonic vehicle travels through the atmosphere at Mach numbers well above the sonic threshold. To reduce overall drag and achieve positive thrust margins at hypersonic velocities, the high-speed air-breathing engines that propel the aircraft must be made an integral part of the vehicle airframe. The highly coupled airframe aerodynamics and engine performance characteristics leads to a complex nonlinear design space. This makes intuition-based design challenging. A computer-based design methodology utilising gradient-based optimisation seems a more suitable and efficient approach to explore the complex design space of hypersonic vehicles. Gradient-based optimisation requires: (a) an objective function evaluation at each design iteration; and (b) the objective function sensitivities for each design parameter. A typical objective function evaluation, e.g. calculating lift/drag, is computationally expensive for a hypersonic vehicle, as a consequence of the complex flow physics that occur at hypersonic velocities. Furthermore, an airframe-engine integrated vehicle requires many design parameters to sufficiently capture the geometric detail. Black-box style gradient-based optimisation, e.g. using finite differences, does not scale well for this type of design problem. The development of new design methods that incorporate efficient gradient-based optimisation techniques are thus required for hypersonic vehicle design. The work in this thesis focuses on the investigation of one particular optimisation approach, discrete adjoint optimisation, in the context of high-speed flow design.

1.1 Motivation and research context

1.1.1 Scramjet-assisted access-to-space

The motivation for the design method presented in this thesis is the development of scramjet-assisted access-to-space launch systems. The number of small satellite launches per year has been steadily increasing over the last decade [8]. As new and innovative satellite-based solutions are engineered, this

number is predicted to increase further. Australia, in particular, has many industries that could benefit from the application of small satellites. For example, remote sensing has applications in mining, agricultural, defence and environmental industries [9]. Despite the increased demand, small satellites still rely on conventional large-scale rocket technology to achieve orbit. The current business model is structured on launching several small satellites with a single rocket-powered launch system. This places fundamental restrictions on when a satellite is launched and, more importantly, to which orbit it can be inserted. For some satellite applications, these restrictions are considered prohibitive, and a tailored launch system that provides flexible orbit insertion is desired. In addition to flexible orbit insertion, a dedicated small satellite launch system should possess a fast turn-around, to remain competitive with current vendors.

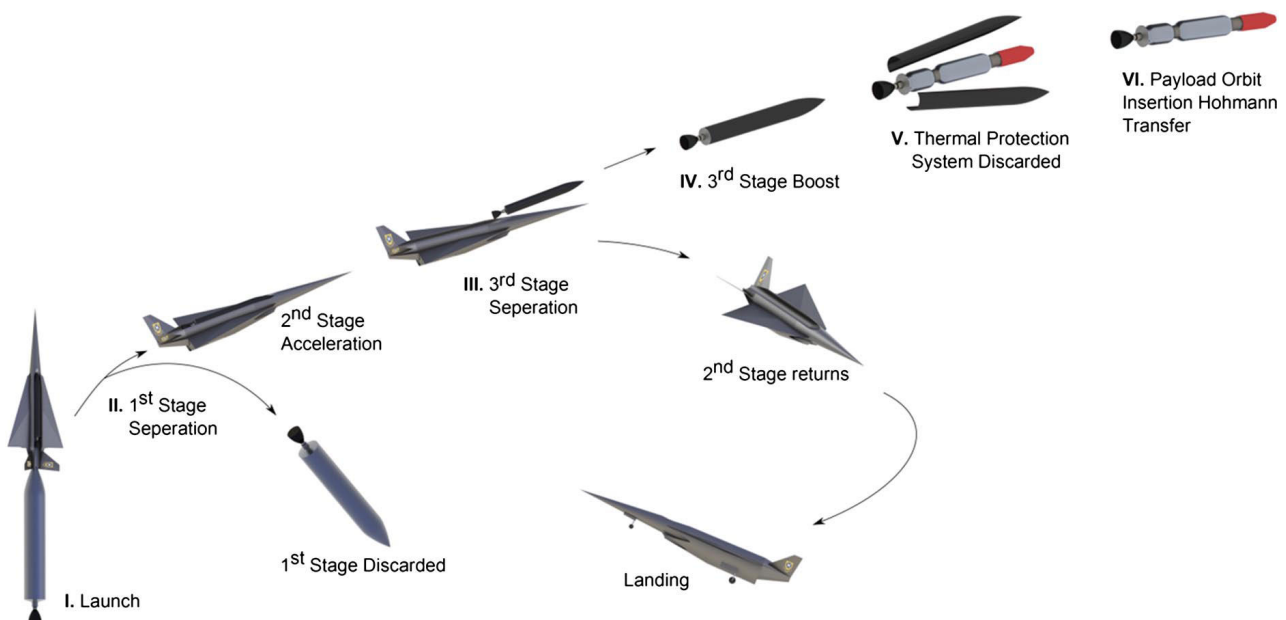


Figure 1.1: Scramjet-rocket launch system mission profile. Reproduced from Preller et al. [10].

At the University of Queensland’s Centre for Hypersonics, a three-stage-to-orbit launch system tailored for small satellites is being developed. A schematic of the launch system is illustrated in Figure 1.1. There are a number of innovative technologies incorporated into the design [11]. Of particular interest to this current work is the second stage vehicle: SPARTAN. SPARTAN is an airbreathing hypersonic accelerator that uses a system of scramjet engines to propel the third stage rocket-booster and payload from Mach 5 to Mach 10.

The scramjet or supersonic combustion ramjet [12], illustrated in Figure 1.2, is a type of high-speed engine that has no moving parts and relies on the turning of the ingested air by the vehicle forebody and engine inlet to provide the required compression. Unlike a traditional ramjet, scramjet engines do not decelerate the captured air to subsonic velocities in the combustor. Instead, the air-fuel combustion process takes place in a flow that remains supersonic in the combustor. This results in less

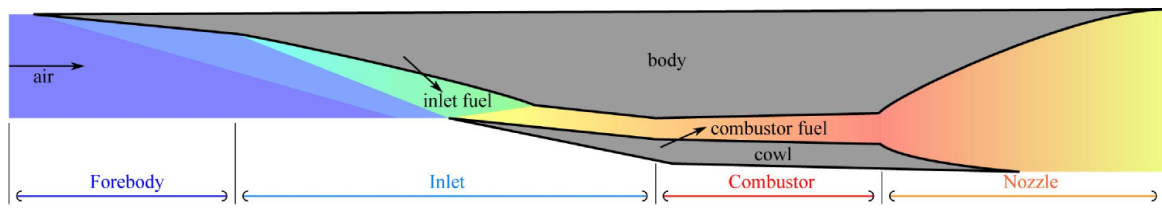


Figure 1.2: Schematic of a scramjet engine. Image by V. Wheatley (22nd October, 2013).

severe total pressure losses for flight at high Mach number. A scramjet-powered accelerator has several advantages over traditional rocket-powered propulsion systems. Foremost is the superior specific impulse offered by scramjet engines, since they do not need to carry any on-board oxidiser. In addition to the superior specific impulse, scramjet-powered accelerators have the advantage of operating like traditional aircraft. Thus offering better manoeuvrability, reusability and fast turn-around between launches [13]. These favourable traits make a scramjet-powered accelerator an attractive second stage vehicle for a small satellite launch system.

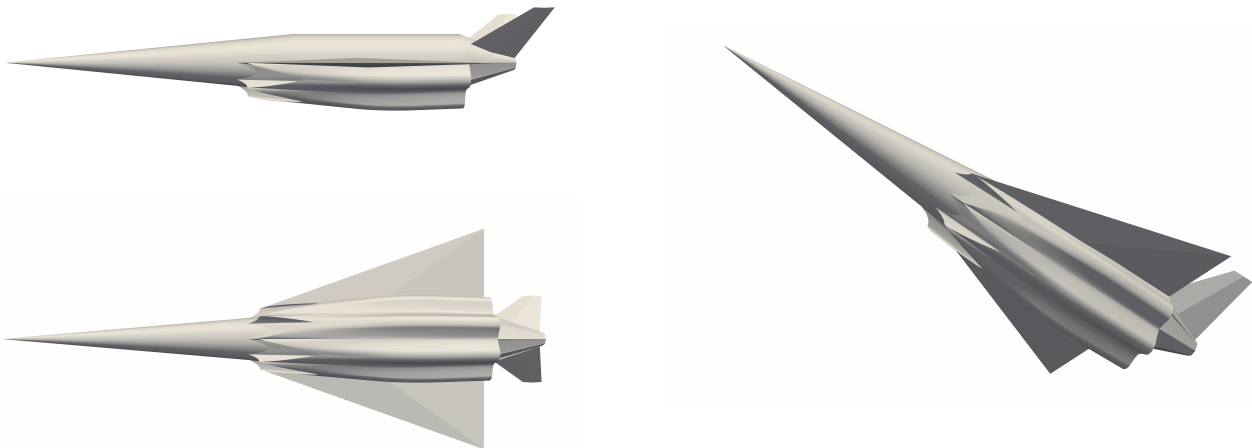


Figure 1.3: Integrated vehicle design for SPARTAN hypersonic accelerator. Image by A. Ward (2nd September 2019)

1.1.2 Hypersonic vehicle design challenges

Careful consideration of airframe-engine integration is critical in scramjet-powered hypersonic vehicle design. In an airframe-engine integrated vehicle, the engines are mounted directly onto the airframe, unlike conventional aircraft which typically have their engines fixed to the underside of the wings, away from the fuselage. An example of airframe-engine integration is illustrated in Figure 1.3, which shows the integrated design of the SPARTAN vehicle. An important effect of the airframe-engine integration is that the entire underside of the vehicle forebody acts as the initial compression system for the engine intake. Similarly, the underside of the aft body acts as part of the engine nozzle.

The integrated design is a direct consequence of operating an airbreathing engine at flight Mach numbers well into the hypersonic regime. A neat back-of-the-envelope calculation, first shown to me by P. Jacobs (personal correspondence, 2019), nicely illustrates the reasoning behind this design choice. Consider the schematic of an airbreathing hypersonic vehicle indicating the relative engine area and vehicle area in Figure 1.4. The drag force acting on the vehicle can be approximated by

$$F_D = \frac{1}{2} \rho_\infty u_\infty^2 C_D A_{\text{vehicle}}, \quad (1.1)$$

where ρ_∞ is the air density, A_{vehicle} is the vehicle reference area, and C_D is the coefficient of drag for the vehicle. The engine specific impulse I_{sp} , which provides a measure of the engine efficiency, is written as

$$I_{sp} = \frac{F_T}{\dot{m}_{\text{fuel}} g}, \quad (1.2)$$

where F_T is the engine thrust, \dot{m}_{fuel} is the fuel flow rate, and g is the acceleration of gravity. For stoichiometric combustion, $\dot{m}_{\text{air}} = 17\dot{m}_{\text{fuel}}$, where $\dot{m}_{\text{air}} = \rho_\infty u_\infty A_{\text{engine}}$, is the captured air mass flow rate. Finally by introducing the non-dimensional Mach number, $M = u_\infty/a_\infty$, where a is the speed of sound of air; and then combining and rearranging Equations 1.1 and 1.2, a relation for the airframe-engine area ratio with respect to Mach number can be determined as

$$A_{\text{ratio}} = \frac{A_{\text{engine}}}{A_{\text{vehicle}}} = \left[\frac{17C_D a_\infty}{2gI_{sp}} \right] M. \quad (1.3)$$

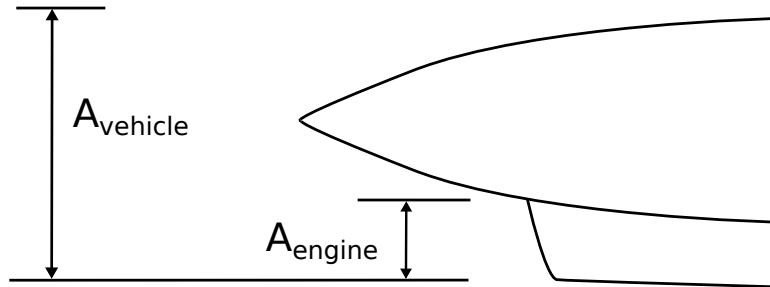


Figure 1.4: Airbreathing hypersonic vehicle schematic.

Sample calculations for various Mach number are provided in Table 1.1. It is observed that the engine-to-airframe area ratio has a strong dependence on the Mach number. For a vehicle that operates up to Mach 10, the engines will make up approximately 20% of the vehicle frontal area. The problem is compounded when also considering that the airbreathing engines will require some form of compression system to compress the freestream air to the desired combustor inlet conditions. If such engines were installed under the vehicle's wings, away from the main airframe, the vehicle mass would be prohibitively large. By integrating the engines on the underside of the airframe, the vehicle forebody can be used as a compression surface for the air-intakes. This significantly reduces the

structural mass of the vehicle. Airframe-engine integration is now widely recognised as an enabling technology for airbreathing hypersonic flight.

Table 1.1: Sample results for Equation 1.3 with varying Mach number.

M	3	5	10	25
A_{ratio}	0.057	0.095	0.19	0.475

The airframe-engine integration adds complexity to the design of hypersonic vehicles. In an airframe-engine integrated vehicle, the vehicle aerodynamics and propulsion system are strongly coupled. This strong coupling ultimately means that conventional techniques that design each component separately and integrate the parts in the final stage of the design process cannot be employed. Instead, airbreathing hypersonic vehicle design requires a technique called Multidisciplinary Design Optimisation (MDO) to generate realisable designs [14]. In MDO, the aerodynamic and propulsive systems are considered concurrently, and so the integrated nature of the hypersonic vehicle is accounted for inherently in the design methodology. The challenge with applying MDO is the large computational expenses incurred from incorporating several physics-based solvers into a numerical optimisation loop. In an effort to make MDO more practical, it is common to apply computationally inexpensive, low-fidelity methods to model the complex physics. For example, the use of panel methods based on shock-expansion theory to evaluate vehicle aerodynamic forces have been applied in several recent MDO of hypersonic vehicles [10, 15–18]. There is some concern that the reduced fidelity modelling does not adequately capture the complex flow phenomena that form around a hypersonic vehicle. Figure 1.5 shows an inviscid simulation of SPARTAN flying at Mach 8. It is observed that a complex structure of shocks has formed around the vehicle. As a result of the leading edge shocks impinging on downstream vehicle surfaces, complex shock-shock and shock wave/boundary layer interactions occur. These flow phenomena are not resolved by using low-fidelity methods and require solutions of the Navier-Stokes equations. In a state of the art MDO of a hypersonic vehicle, this concern has been addressed by incorporating solutions of the Reynolds-Averaged Navier-Stokes (RANS) equations for the underside vehicle flow-path [19]. Due to computational expense, however, the number of design variables used to represent the vehicle geometry for the design optimisation was restricted to fewer than fifteen. This is considered a coarse parameterisation when considering that to parameterise a hypersonic inlet alone requires 50+ design variables to capture the geometric detail [20]. The design challenge for hypersonic vehicles is developing methods for incorporating high-fidelity physics-based solvers into the MDO of geometrically-complex vehicles requiring many-parameters.

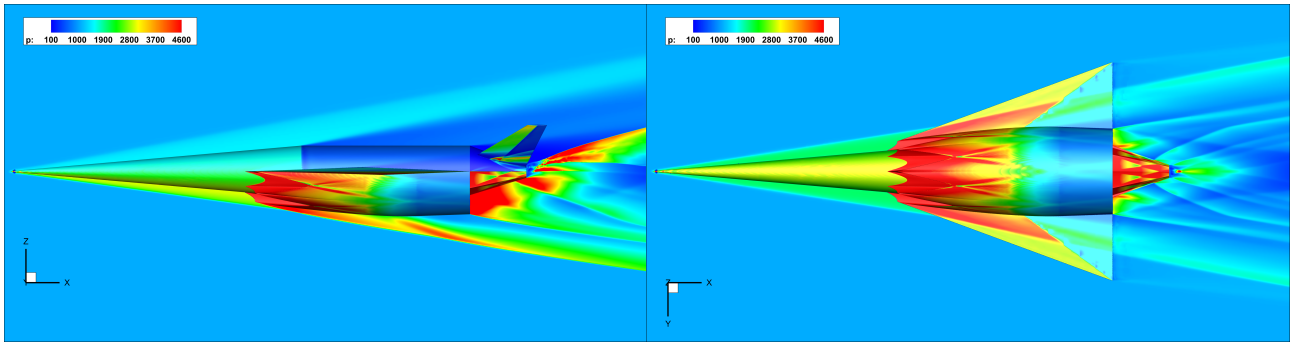


Figure 1.5: CART3D simulation of SPARTAN at Mach 8 flight conditions. Image provided by A. Ward (5th August 2019).

1.1.3 Modern CFD-based design optimisation

A numerical optimisation algorithm is required to drive the MDO methodology outlined above. In this work, the focus is restricted to gradient-based optimisation techniques, given the computational expense associated with evaluating the objective function via a converged CFD solution. Alternate global methods, such as those based on evolutionary systems, are better suited to optimisation problems where the objective function is computationally inexpensive. The most straightforward approach to computing the design sensitivities used in a gradient-based optimisation algorithm is via finite differences. In the finite difference approach, an additional CFD solution is required for each design variable per design iteration. This approach does not scale well for many-parameter design optimisation. Consider the CFD solution in Figure 1.5. This inviscid result took approximately 5 hours to converge on 24 compute cores. If this vehicle was parameterised by 10 design variables, as done in the work by Jazra et al. [16, 17] and Preller et al. [10, 18], then one design iteration would take approximately 50 hours using finite differences. This is too prohibitive for vehicle design. A modern approach to CFD-based optimisation is the adjoint method [21]. The adjoint method enables the computation of all shape sensitivities for a given problem at the cost of only one flow solution and one adjoint solution [22]. As a result of its efficiency, the adjoint method has become widely used for aircraft design, evolving to the design optimisation of full wing-body configurations [23]. The adjoint method has also been considered an enabling technology for MDO of complex configurations [24], recent examples include aerostructural design optimisation [25] and sonic boom mitigation [26]. MDO of hypersonic vehicles could potentially benefit significantly from the efficiency of an adjoint-based design optimisation approach. However, before adjoint-based MDO can be achieved, the applicability of the adjoint method for high-speed design analysis within the individual disciplines should be investigated. In the interim before adjoint-based MDO, there could be significant pay-offs from the application of adjoint optimisation techniques within the disciplines [27, 28]. This current work aims to investigate the applicability of the adjoint method to the particular discipline of aerodynamic shape optimisation in hypersonic flow.

1.2 Thesis aims

The overall aim of this thesis is to

Investigate how high-fidelity CFD-based optimisation using the adjoint method can be applied to aerodynamic shape design in hypersonic flow.

This will be achieved through the following objectives:

1. *Implement a Newton-Krylov accelerated, unstructured grid, compressible flow solver*

The University of Queensland's in-house compressible flow code, Eilmer, operates on structured grid topologies using an explicit time-accurate method to march in time to the steady-state. To enable geometric flexibility and improve efficiency in reaching steady-state solutions, a Newton-Krylov accelerated steady-state solver that operates on unstructured grids is implemented. The Newton-Krylov approach applied to high-speed flows has some coverage in the literature, however, has not been covered extensively for second-order spatially accurate methods with turbulence modelling. The approach is selected over other competing methods due to its applicability to complex geometries, owing to no inherent preferential relaxation directions.

2. *Implement a discrete adjoint solver based on the developed flow solver*

In the context of CFD-based optimisation, the adjoint method is the most efficient. The primary complication of implementing the discrete adjoint method is linearising the flow solver routines. For this work, the second-order spatially accurate flow routines for the steady-state solver outlined above are linearised with no additional simplifications via a complex variable approach.

3. *Develop an optimisation package for aerodynamic shape design*

The flow solver and adjoint solver are two pieces of the overall optimisation package. These are coupled with a surface parameterisation method and a mesh deformation procedure in order to compute shape sensitivities via the adjoint method. The gradient evaluation routines in Eilmer coupled to the open-source optimisation library, DAKOTA, to construct a complete optimisation package.

4. *Apply the developed optimisation package to aerodynamic shape design in hypersonic flow*

The developed optimisation software is applied to several design cases to assess the applicability of the discrete adjoint method to hypersonic vehicle design.

1.3 Thesis outline

This thesis is organised into nine chapters which are outlined below. The appendices included at the end of this document contain further technical information.

Chapter 2 - Background for Hypersonic Vehicle Design

This chapter provides a background to hypersonic vehicle design. It is divided into three sections. Firstly, an introduction to airbreathing hypersonic vehicles is presented. Following this, the relevant flow phenomena that encompass an airbreathing hypersonic vehicle as it travels through the atmosphere is examined. The chapter then concludes with a critical review of current design methodologies.

Chapter 3 - Review for Adjoint-Based Aerodynamic Shape Optimisation

This chapter reviews the literature for adjoint-based aerodynamic shape optimisation. The chapter begins by covering the broad topics of computational fluid dynamics and optimisation. Following this, an introduction to the adjoint method is presented. A key concern with implementing an adjoint solver is the method by which the flow solver is linearised, a discussion of this is presented here. The chapter concludes with a review of adjoint-based optimisation in hypersonic flow.

Chapter 4 - Flow Solver

This chapter provides details of the flow solver implemented for this work. The topics covered include a detailed description of the Newton-Krylov implementation, in addition to details pertaining to the inviscid and viscous flux calculations. Where appropriate, example results from several baseline problems are presented to aid in the discussion.

Chapter 5 - Flow Solver Verification and Validation

This chapter presents the results of a verification and validation study for the flow solver. Several methods of verification are employed to examine the accuracy of the implemented numerical methods. An extensive set of validation cases have been taken from the literature to establish the appropriateness of the implemented physical models for design analysis in high-speed flow.

Chapter 6 - Adjoint Solver

This chapter details the adjoint solver implemented for this work. Special attention is given to describing the method employed to construct the flow solver linearisation. The chapter concludes by

presenting verification test results for the adjoint solver, in addition to presenting a debugging technique used throughout the adjoint solver development.

Chapter 7 - Shape Optimisation Methodology

This chapter provides a description of the shape optimisation package developed for this work. The parameterisation technique and grid movement strategy employed for this work are presented. Details on connecting the flow solver and adjoint solver to the open-source optimisation package DAKOTA are also provided.

Chapter 8 - Optimisation Applications

Results from two optimisation applications are presented in this chapter. The first application is the minimisation of wave drag on an axisymmetric body. This application presents a simple test case for verifying the overall accuracy of the developed shape optimisation package. The second application presents the re-design of a pre-existing hypersonic inlet.

Chapter 9 - Conclusions

The body of this thesis concludes by summarising the most significant findings from Chapters 4 through 8. Recommendations for future studies are provided.

BACKGROUND FOR HYPERSONIC VEHICLE DESIGN

The design of any aircraft begins with the mission definition. The details of the mission dictate the preliminary design choices and the environment in which the aircraft must operate. This information provides an insight into the general geometric layout of the aircraft and the relevant flow physics that should be included in the design analysis. These two characteristics of an aircraft have a direct impact on the selection of an appropriate design approach. In this chapter, a background for the design of airbreathing hypersonic accelerators is presented. The discussion will cover the geometric complexities of hypersonic vehicles, in addition to the relevant flow phenomena that encompass a hypersonic vehicle as it travels through the atmosphere. The chapter concludes by providing a critical analysis of the current design practices for airbreathing hypersonic vehicles. The purpose of this chapter is to reiterate and expand on what the design challenges are for hypersonic vehicles, with regards to external aerodynamics and, to some extent, the integration with an internal propulsion flowpath.

2.1 Airbreathing hypersonic vehicles

The typical mission definition for an airbreathing hypersonic accelerator is to assist in delivering a payload into orbit. The accelerator can be used as a single stage to orbit vehicle or as a reusable stage in a multistage launch system. Early research on the National Aerospace Plane (NASP) [29], however, has shown that a multistage system is significantly more practical. Depending on the exact staging number, a combination of propulsive systems may be required, such as a turbine-based combined cycle (TBCC) or rocket-based combined cycle (RBCC) engine. Regardless of the propulsive system used for low Mach number flight, e.g. $M \leq 5$, scramjet engines are currently the preferred engine for high Mach number flight [12]. In Chapter 1, an example demonstrated that for moderate hypersonic

Mach number, airbreathing engines account for approximately 20% of the vehicle frontal surface area. The airframe-engine integrated design was proposed to accommodate the large engine intakes required at high Mach number flight, however, it serves several other purposes as well. Airframe-engine integration was first proposed in the late 1960's as a successor to NASA's Hypersonic Research Engine (HRE). The HRE was a "pod" style axisymmetric scramjet engine [30] that provided excellent internal performance but could not produce positive installed thrust due to several limitations [31]. Firstly, in agreement with the theoretical analysis presented in Chapter 1, the HRE scaled poorly for high Mach number operation. Secondly, the HRE required a large exit to inlet area ratio to sufficiently expand the exhaust gases, this resulted in excessive cowl drag. Thirdly, inefficient use of the cross-sectional area within the shock layer meant that the inlet had an insufficient combustor inlet capture area. Lastly, it was found that vehicle-body/pod-engine interference increased drag and heating values. To reduce these effects, the airframe-engine integrated vehicle was designed. In an airframe-engine integrated vehicle, the aerodynamics and propulsion systems are tightly coupled. This coupling allows vehicle components to have multiple purposes. For example, the forebody of the vehicle not only houses the fuel tanks, vehicle systems and makes up part of the thermal protection system, but also acts as part of the compression system for the combustor inlet. A consequence of an integrated design is that the vehicle aerodynamics and propulsion system are tightly coupled. For instance, the propulsion system provides not only thrust but also portions of vehicle lift and pitching moment [15, 32]. The strong coupling has a direct impact on the design methodology for hypersonic aircraft, as will be discussed in Section 2.3. The airframe-engine integrated design is now widely considered as a crucial element to the success of hypersonic vehicle design.

2.2 Relevant flow phenomena

This section will review the relevant flow phenomena that encompass a hypersonic accelerator traveling through the atmosphere. The altitude at which an airbreathing accelerator operates is limited by the requirement of robust combustion: recent optimised trajectories have an upper bound of approximately 30-35 km [33]. The air is sufficiently dense at this altitude such that the Knudsen number, defined as the ratio of the molecular mean free path length to a representative physical length scale, is small enough to restrict our attention to the continuum flow regime.

The hypersonic regime spans a wide range of Mach number and is characterised by a broad range of physical phenomena. A typical range of Mach number that an accelerator might operate between is Mach 5 to Mach 10. This range is at the lower end of the hypersonic spectrum, and consequently, many of the high-temperature gas-dynamics do not apply to the design of accelerator vehicles. For example, at the flight Mach numbers considered here, the effects of thermochemical non-equilibrium and radiative heat transfer have been shown to have a negligible effect on the vehicle aerodynamics and propulsive system performance [34–37].

The most significant phenomena experienced by a vehicle in hypersonic flight is the system of strong shock waves that form on the frontal surfaces of the vehicle. A discontinuous¹ increase in entropy occurs across a shock wave which manifests itself as a drag force on the vehicle, this phenomenon is sometimes referred to as wave drag. Another adverse effect of the system of shock waves forming on the frontal surfaces is that they can impinge on the vehicle surface downstream, such as the wings or engine intakes. Several locations of shock-shock interactions on a generic hypersonic vehicle are indicated in the schematic presented as Figure 2.1. The effect of shock-shock interactions is to generate locally high pressures and heating loads.

In addition to wave drag, hypersonic vehicles must overcome severe skin-friction drag. The shear stress along the vehicle surface is proportional to M_e^2 [37], the Mach number at the boundary layer edge, as shown in Equation 2.1. In turn, M_e is proportional to the flight Mach number, and, as a consequence, the skin-friction increases significantly as flight Mach number increases.

$$\tau_w = \frac{1}{2} \gamma p_e M_e^2 C_f. \quad (2.1)$$

The shear-stress at the wall is also a function of the skin friction coefficient (C_f). This coefficient depends on what state the boundary is in: laminar, turbulent or in transition. In a laminar boundary layer, the viscous forces of the fluid dominate over the inertial forces, and, as a result, smooth constant flow persists. Alternatively, turbulent boundary layers are characterised by chaotic eddies and vortices [37]. Turbulent boundary layers transfer energy more readily to the surface due to the chaotic motion, thus turbulent boundary layers exhibit higher friction and heat transfer rates. To add further complexity, the shock waves that impinge on the vehicle surface interact with the boundary layer. The interaction of a shock wave with a turbulent boundary layer is a complex phenomenon, and much is still unknown for interactions occurring on heated vehicle surfaces. The primary effect of these interactions is to produce severe local pressure and heating loads, in addition to creating large regions of separated flow. Figure 2.1 indicates regions on a generic hypersonic vehicle where shock wave boundary layer interactions are likely to occur.

The complex flow phenomena that encompass a hypersonic vehicle requires sophisticated multi-physics simulation software to capture the appropriate detail. This provides certain challenges to the design of hypersonic vehicles, as will be discussed in Section 2.3.

¹When compared to the vehicle length scale.

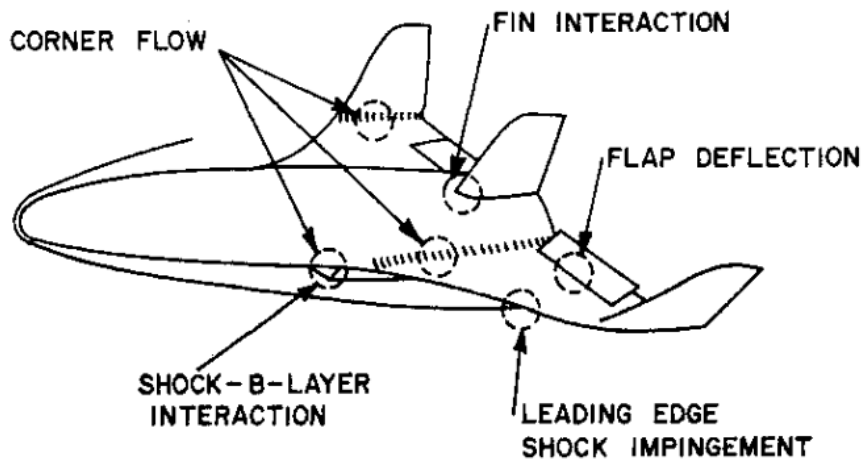


Figure 2.1: Shock wave/viscous interactions on a generic hypersonic vehicle. Reproduced from Korkegi [38].

2.3 Existing design methodology

Traditional aircraft design methods employ a human-in-the-loop approach. For example, several of the NACA 4 digit airfoils were designed by engineers relying on wind tunnel test results and engineering-based intuition to aid in manually shaping the geometry toward an optimum [39, 40]. Modern applications of human-in-the-loop design have replaced wind tunnel test results with CFD solutions to improve efficiency and reduce costs [39]. For a highly coupled system, often the separate disciplines have competing effects on the design requirements. This makes intuition-based design by an engineer difficult [15, 28]. For such highly nonlinear design problems, numerical optimisation techniques are required to assist in finding an optimum solution efficiently. The simplest application of numerical optimisation is to perform sequential optimisation of each discipline. However, in a highly coupled system, this approach does not often converge to the true optimum [25]. This is particularly problematic for hypersonic vehicles, where the optimum or near optimum performance of the overall system is required to achieve a sufficient design margin [14, 19]. To account for the coupling of the systems, all disciplines should be included in the optimisation concurrently. This branch of optimisation is called Multidisciplinary Design Optimisation (MDO) [41]. Today, MDO is considered essential to achieving hypersonic vehicle designs that achieve adequate performance at a reasonable operational cost [14].

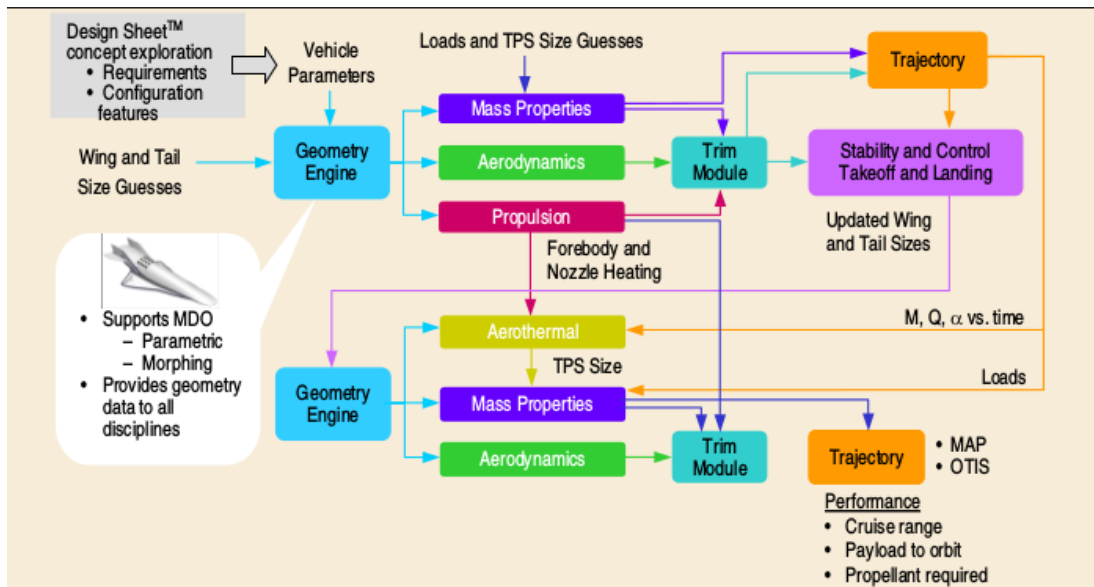
Several hypersonic vehicle MDO studies have been published over the last decade. The incorporation of several high-fidelity physics-based solvers into an optimisation routine is currently too computationally expensive. To reduce the computational costs, MDO studies have typically applied low-fidelity models that provide a means of fast analysis at the cost of accuracy. An example of the type of reduced modelling applied in MDO is the use of panel methods that rely on analytic solutions

to the shock-expansion equations and boundary layer correction techniques to evaluate aerodynamic forces [10, 15, 16, 42]. Figure 2.2(a) presents a flow diagram showing the interactions between a partial set of analysis codes utilised in Boeing's hypersonic vehicle MDO. This illustrates the state of the art for the MDO of hypersonic vehicles. The analysis codes vary in fidelity depending on the discipline. The methodology, however, is novel in its inclusion of high-fidelity RANS solutions to model the airbreathing engine inlet flow path [19]. The optimisation algorithm employed in the MDO routine couples a Design of Experiments (DOE) technique with surrogate modelling and a Sequential Quadratic Programming (SQP) method. The software has been applied to the design of a Boeing reusable launch vehicle concept called FASST [43], shown in Figure 2.2(b). In this example optimisation, the FASST geometry was parameterised using 50 design parameters. However, only 12 design variables were free for the MDO to modify [19].

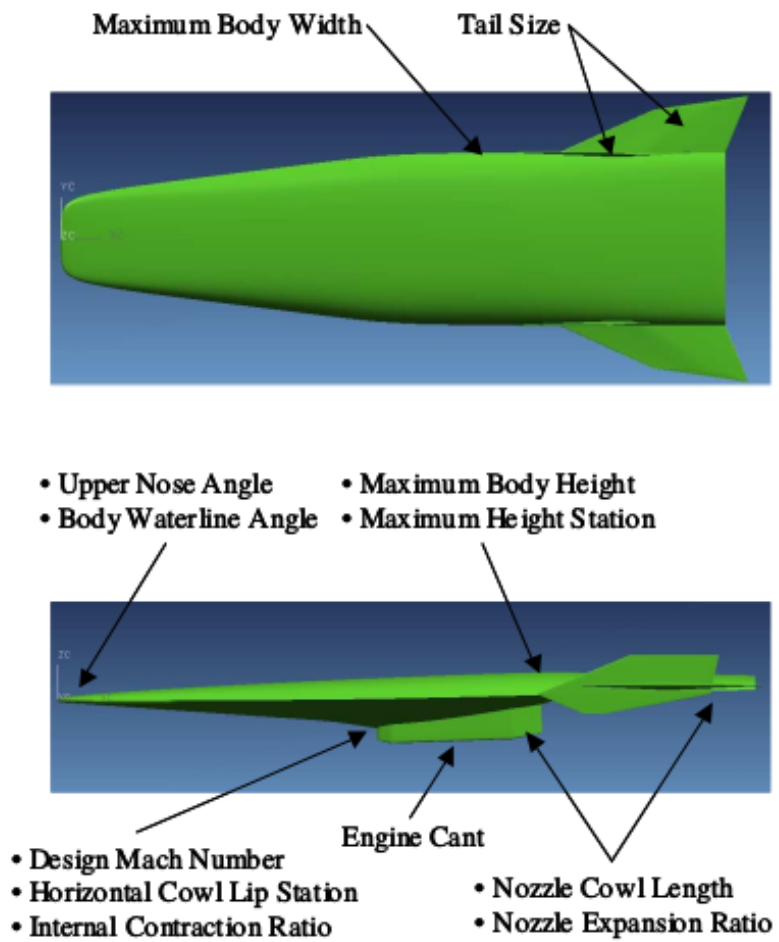
Other hypersonic vehicle MDO applications published in the literature have employed low-fidelity physics modelling coupled with a Simplex method [10, 15, 16, 42]. In these cited studies, the vehicle geometries were also parameterised using on the order of 10 design parameters. The Simplex method is a local direct search method that only requires objective function evaluations [44]. It is often chosen over gradient-based methods since it typically only requires one or two objective function evaluations per design iteration. This is considered more economical than a gradient-based method that employs numerical gradients computed via finite differences, which requires an objective function evaluation for each design variable per design iteration. A disadvantage of the Simplex method is that it can take a large number of design iterations to converge. In the context of hypersonic missile MDO, the simplex method required 200-300 design iterations to find the optimal aircraft parameterised using five design parameters [15].

In Chapter 1, the challenges associated with hypersonic vehicle design were discussed. It was concluded that the primary challenge is developing methods for incorporating high-fidelity physics-based solvers into the MDO of geometrically-complex vehicles requiring many-parameters. Boeing's MDO software has begun to address the challenge of incorporating high-fidelity physics-based solvers into the design loop. However, to reduce computational costs, it is more common to employ low-fidelity modelling. The challenge of many-parameter MDO has yet to be addressed, with all cited MDO studies having used only a handful of design parameters to capture the geometric detail. It is thought that hypersonic MDO could potentially benefit from incorporating high-fidelity modelling techniques and increased design parameterisation flexibility. No published results say whether hypersonic vehicle MDO should include a more flexible parameterisation. However, the results from an aerodynamic design optimisation study by Ueno et al. [45] provide an insight into many-parameter vehicle optimisation. In this work, a hypersonic airliner was parameterised using several Bézier curves and surfaces. In total, 127 design parameters were used in the optimisation. In the present author's observation, a comparison of the optimised geometry and the baseline vehicle showed that whilst the overall geometric topology remained similar, local geometric topologies, such as the lifting surfaces,

had altered substantially from the baseline. The flow for this optimisation was modelled using the Euler equations. It would be computationally taxing to perform MDO of such a vehicle using solutions of the Reynolds-Averaged Navier-Stokes equations coupled with either a finite difference or Simplex method. A modern gradient-based optimisation technique, called the adjoint method, has shown promise in enabling solutions to MDO problems in several engineering fields that were once considered intractable [24–26]. It is believed that adjoint-based optimisation could enable high-fidelity MDO of hypersonic vehicles. A review of adjoint-based optimisation will be presented in Chapter 3.



(a) Flow diagram of MDO methodology. Reproduced from Bowcutt. [14]



(b) FASST hypersonic vehicle designed using MDO. Reproduced from Bowcutt et al. [19]

Figure 2.2: Boeing’s state of the art MDO framework for a hypersonic vehicle.

REVIEW OF ADJOINT-BASED AERODYNAMIC SHAPE OPTIMISATION

In this chapter, a literature review for adjoint-based aerodynamic shape optimisation is presented. The chapter begins with a brief review of computational fluid dynamics and numerical optimisation, in the context of the current work. Following this, an introduction to the adjoint method is presented. A key concern with implementing the adjoint method is the technique by which the flow solver is linearised, and so, a review of several approaches is presented here. The chapter concludes with a summary of previously published works on adjoint-based optimisation in the hypersonic flow context.

3.1 Computational Fluid Dynamics

Computational Fluid Dynamics (CFD) refers to the analysis of fluid dynamic systems by means of solving sets of physical equations that describe fluid motion. Figure 3.1 illustrates the various levels of fidelity available for modelling a fluid. For irrotational inviscid flows, where the viscous effects are negligible, the potential methods suffice to represent fluid motion. For inviscid rotational flows, the Euler equations are employed. For some applications where viscous effects are non-negligible, it is appropriate to employ either the potential flow or Euler equations in conjunction with a boundary layer correction technique. However, such an approach is not appropriate when modelling complex interactions between the viscous and inviscid components of a flow, such as shock wave/boundary layer interactions. To simulate complex viscous interactions requires solutions of the full Navier-Stokes equations. The Navier-Stokes equations contain all the necessary modelling information to completely resolve both laminar and turbulent flows. This approach is referred to as Direct Numerical Simulation (DNS). To perform DNS requires the fluid domain to be finely discretised, such that the computational elements are small enough to resolve the entire spectrum of turbulent eddies. This typi-

cally demands a mesh consisting of roughly $Re^{9/4}$ elements to appropriately simulate a turbulent flow using DNS. Even for a modest Reynolds number (Re) of 1 million, this would require on the order of 10^{13} computational cells. To reduce the computational costs, it is possible to resolve only the largest eddies whilst applying a sub-grid model to approximate eddies under some specified threshold. This method is referred to as Large Eddy Simulation (LES). LES requires significantly less mesh points than DNS. However, the computational expense is still prohibitive for complex geometries. For some industrial-level engineering problems, it is appropriate to model all the turbulent eddy structures. In this approach, a time-averaging filter is applied to the Navier-Stokes equations: the set of resulting equations are the Reynolds-Averaged Navier-Stokes (RANS) equations. The application of RANS modelling is appropriate for problems that require only steady-state solutions of the flow field. Solutions of the RANS equations are far more economical than DNS and LES, with three-dimensional turbulent solutions achievable on engineering workstations within 24 to 48 hours. As a consequence, RANS is typically employed instead of DNS and LES methods for aircraft design [39].

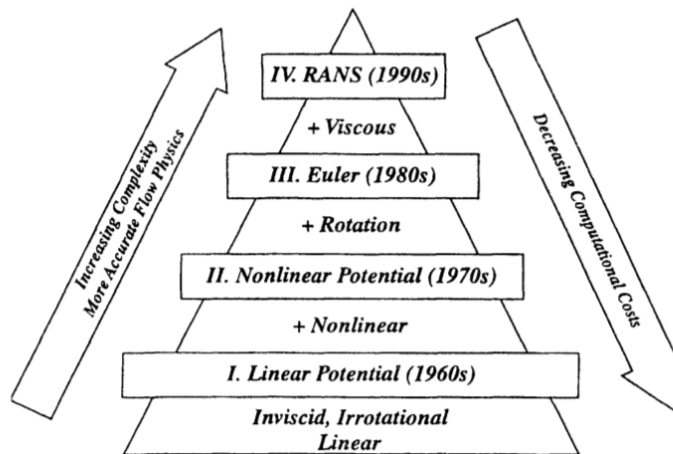


Figure 3.1: Computational fluid models. Reproduced from Jameson et al. [27]

3.2 Numerical optimisation

This section begins with a definition of shape optimisation. Thévinin et al. [22] identified an unfortunate trend in the literature that misrepresents parametric studies as optimisation, and so, it is important to remove any ambiguous terminology. In this current work, optimisation is taken to mean the pursuit of a vector of design parameters which is optimal in that it minimises a certain objective function

while satisfying given constraints. This can be formulated mathematically as,

$$\begin{aligned}
 &\text{minimise} && J(x_j) \\
 &\text{w.r.t} && x_j && j = 1, 2, \dots, N_x \\
 &\text{subject to} && g_m(x_j) \geq 0, && m = 1, 2, \dots, N_g
 \end{aligned} \tag{3.1}$$

where J is the objective function, x_j is one of the N_x design variables, and g_m is one of the N_g nonlinear constraint equations. Aerodynamic shape optimisation simply refers to the specific optimisation of some aerodynamic characteristic, such as minimising drag or maximising lift, by altering the geometry in some way. The general optimisation problem can be explained graphically by considering the schematic representation of a single parameter optimisation problem illustrated in Figure 3.2. The objective is: given a starting point, either point A or D in this example, find the minimum point B. The algorithms developed to search for the optimum point are divided into two categories, local methods and global methods.

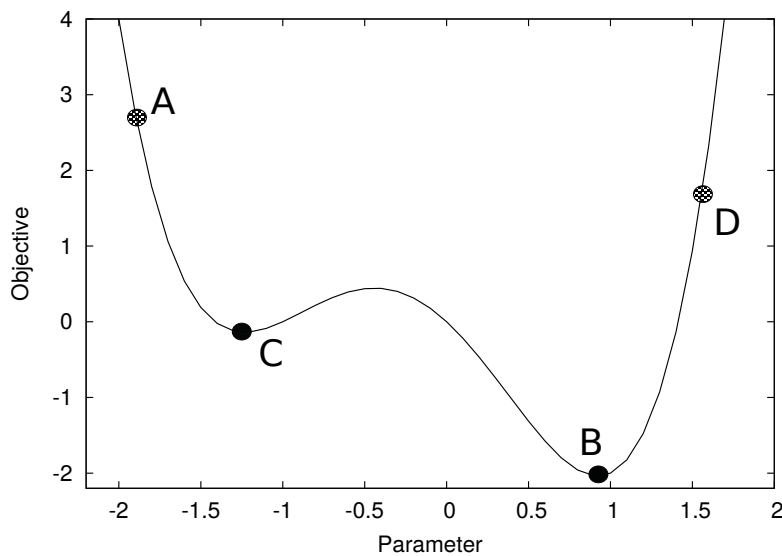


Figure 3.2: Single parameter optimisation.

Global methods search the parameter space using only objective function evaluations. They are given the name global, since the algorithms in this category will attempt to find the global minimum given any starting condition. In practice, referring to Figure 3.2, a global method would find the true optimum, point B, given either of the starting points. Popular global methods include genetic algorithms and bio-inspired probabilistic algorithms. The primary limitation of these methods is the large computational costs. The number of objective function evaluations rapidly increases with respect to the number of design parameters. Consequently, the time required to find the global optimum can be prohibitive for computationally expensive objective function evaluations.

Local methods use both objective function evaluations and an approximation of the objective

function gradient to search the parameter space. By doing so, the algorithm is accelerated to finding the nearest minimum value. If the design space is convex, then local search methods will converge on the global optimum. However, for complex design spaces, it is hard to prove convexity, and as such, it is only certain that the converged solution is a local optimum [22]. This can be explained graphically, again by referring to Figure 3.2. For a local search method, the algorithm will find point B given starting point D, or point C given starting point A. The simplest gradient-based method is Steepest Descent, where the gradient information is used to ensure that the algorithm is always moving “down-hill”. More sophisticated gradient-based methods such as Newton-type methods require second-order gradients (Hessian matrix) to be evaluated, but they exhibit faster convergence.

There have been several comparative studies published that examine the relative performance of global and local search methods for aerodynamic shape optimisation. Most of these works provide qualitative results and do not provide a definitive answer on which method is more superior. However, there appears to be a consensus that local search methods are more efficient at finding an optimum, albeit at the expense of potentially converging on a local minimum [46, 47]. Zingg et al. [48] provided a more quantitative comparison by comparing the number of flow solutions required to obtain a specified degree of convergence for the single point, multipoint and multiobjective design optimisation of subsonic and transonic aerofoils. The published results showed that a genetic algorithm was 5 to 200 times more expensive than a gradient-based local search method. Importantly, the final converged solutions for the two alternate search methods were observed to be similar, suggesting that the local search method and the global search method had found the same minimum point. Hybrid methods which combine a local search method with an initial population of baseline designs have been suggested as a solution to the local minimum deficiency of local search methods [49].

3.3 The adjoint method

The adjoint method has gained popularity in recent years as an efficient means of computing sensitivities for use in gradient-based optimisation [21, 50]. A distinguishing feature of the method is that the cost of computing the complete sensitivity vector is independent of the number of design variables [28]. To compute the adjoint sensitivities requires only one flow solution and one adjoint solution per design iteration. Given that an adjoint solution costs roughly between one to ten times that of a flow solution [51], the adjoint method is significantly more efficient at computing sensitivities than a finite difference approach for complex geometries requiring many design parameters. The primary disadvantage of the adjoint method is the non-trivial code development required to implement an adjoint solver. Unlike the finite difference approach which treats the flow solver as a black-box, implementation of the adjoint method requires direct access to the flow solver source code.

The adjoint method can be applied in either the discrete form or continuous form. The discrete adjoint method is derived directly from the discrete state equations, whilst the continuous adjoint

method first derives the system of adjoint PDEs before discretising them [22]. There is no compelling evidence that proves one formulation is better than the other, and both methods should, as the mesh is refined, provide the same level of accuracy [52, 53]. There are, however, two key advantages of deriving the adjoint system directly from the discrete state equations. Foremost is that there exists algorithmic methods to linearise discretised equations. This ultimately means that the adjoint system construction procedure may be automated. Techniques for automating the linearisation are discussed in Section 3.4. This is not true for the continuous approach, which requires re-derivation of the adjoint system for every new modification to the underlying set of equations. Secondly, the discrete adjoint method can be easily verified against a direct differentiation of the flow code, since the differentiation is of the discretised system. The continuous method does not have any theoretical equivalent, making verification difficult. For these reasons, the discrete adjoint method is the preferred method for this current work.

The derivation of the discrete adjoint method is presented here. The sensitivity of an objective function, $J(\mathbf{Q}(\mathbf{D}), \mathbf{X}(\mathbf{D}))$, with respect to the design variables, \mathbf{D} , is given by the chain rule to be

$$\frac{dJ}{d\mathbf{D}} = \frac{\partial J}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \mathbf{D}} + \frac{\partial J}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial \mathbf{D}}, \quad (3.2)$$

where \mathbf{Q} is the vector of flow state variables, \mathbf{X} are the grid points, and \mathbf{R} is the residual vector. The evaluation of Equation 3.2 incurs significant computational cost as a direct result of requiring the evaluation of the sensitivity of the flow state variables with respect to the design variables, $\frac{\partial \mathbf{Q}}{\partial \mathbf{D}}$. The simplest approach to computing this term is via finite differences. In this approach, each design variable is perturbed sequentially and an independent flow solution is converged. This approach does not scale favourably for problems with many design variables. By applying the adjoint method, this costly term can be eliminated from Equation 3.2. In this work, the Lagrange multiplier approach is preferred [51]. To begin the derivation, the objective function, $J(\mathbf{Q}(\mathbf{D}), \mathbf{X}(\mathbf{D}))$, is reformulated as a Lagrangian function

$$L(\mathbf{Q}(\mathbf{D}), \mathbf{X}(\mathbf{D}), \lambda) = J(\mathbf{Q}(\mathbf{D}), \mathbf{X}(\mathbf{D})) + \lambda^T \mathbf{R}(\mathbf{Q}(\mathbf{D}), \mathbf{X}(\mathbf{D})), \quad (3.3)$$

under the assumption that the flow is at steady state, i.e.

$$\mathbf{R}(\mathbf{Q}(\mathbf{D}), \mathbf{X}(\mathbf{D})) = 0. \quad (3.4)$$

The new term λ , is used to denote the adjoint variables. By applying the chain rule, the sensitivity of this Lagrangian function with respect to the design variables is given as the following

$$\frac{dL}{d\mathbf{D}} = \frac{\partial J}{\partial \mathbf{X}} \frac{d\mathbf{X}}{d\mathbf{D}} + \left(\frac{\partial J}{\partial \mathbf{Q}} + \lambda^T \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \right) \frac{d\mathbf{Q}}{d\mathbf{D}} + \lambda^T \frac{\partial \mathbf{R}}{\partial \mathbf{X}} \frac{d\mathbf{X}}{d\mathbf{D}}. \quad (3.5)$$

The sensitivity of the flow state variables with respect to the design variables, $\frac{d\mathbf{Q}}{d\mathbf{D}}$, is computationally expensive to compute, and the goal of the adjoint method is to remove it from the sensitivity calculation. Given that the residual of the Navier-Stokes equations at steady-state is zero (since they are a set of conservation equations), the adjoint variables may take any arbitrary value. The adjoint variables are chosen to eliminate the $\frac{d\mathbf{Q}}{d\mathbf{D}}$ term in Eq. 3.5 by setting

$$\left(\frac{\partial J}{\partial \mathbf{Q}} + \lambda^T \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \right) = 0, \quad (3.6)$$

this is more commonly written as the adjoint equation

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \right)^T \lambda = - \left(\frac{\partial J}{\partial \mathbf{Q}} \right)^T. \quad (3.7)$$

The objective function sensitivity is then given by

$$\frac{dL}{d\mathbf{D}} = \frac{\partial J}{\partial \mathbf{D}} + \lambda^T \frac{\partial \mathbf{R}}{\partial \mathbf{D}}, \quad (3.8)$$

where,

$$\frac{\partial \mathbf{R}}{\partial \mathbf{D}} = \frac{\partial \mathbf{R}}{\partial \mathbf{X}} \frac{d\mathbf{X}}{d\mathbf{D}}, \quad (3.9)$$

and,

$$\frac{\partial J}{\partial \mathbf{D}} = \frac{\partial J}{\partial \mathbf{X}} \frac{d\mathbf{X}}{d\mathbf{D}}. \quad (3.10)$$

Eq. 3.8 is independent of $\frac{d\mathbf{Q}}{d\mathbf{D}}$. As a consequence, it is significantly less computationally expensive to evaluate than Eq. 3.5. The calculation of the residual sensitivity $\frac{\partial \mathbf{R}}{\partial \mathbf{D}}$ in Equation 3.8 term can be computationally expensive for certain mesh deformation algorithms¹, since it implicitly requires several deformed meshes equal to the number of design variables. This term can be removed in a similar manner to the flow sensitivity $\frac{\partial \mathbf{Q}}{\partial \mathbf{D}}$ by applying a mesh adjoint, in addition to the flow adjoint [54]. A mesh adjoint was not required in this work, as will be discussed in Chapter 7. The interested reader is directed to the referenced work by Nielsen et al. [54] for more details on implementing the mesh adjoint. The procedure to evaluate the objective function sensitivity via the discrete adjoint method is then:

1. Converge a flow solution.
2. Construct the adjoint operator and other partial derivatives.
3. Converge an adjoint solution.
4. Evaluate the objective function sensitivities via Equation 3.8.

¹Refer to Chapter 7 for a review of mesh deformation methods.

The difficulty in applying the discrete adjoint method is in constructing the adjoint operator, $(\frac{\partial \mathbf{R}}{\partial \mathbf{Q}})^T$. Techniques for constructing this operator are discussed in the next section.

3.4 Construction of the adjoint operator

Several methods exist for constructing the adjoint operator. This section will briefly describe these methods and discuss the advantages and disadvantages of each method. Readers familiar with CFD methods might recognise the adjoint operator as the transpose of the flow Jacobian matrix.

3.4.1 Analytic differentiation

Analytic differentiation is the most widely referenced method for constructing the adjoint operator. In this approach, the underlying numerical methods of the flow solver are differentiated by hand. Hand differentiation of a spatially second-order flow solver requires a significant number of hours and is prone to human error. However, there is evidence that shows adjoint solvers developed from hand differentiated routines typically provide the most efficient means of evaluating an adjoint solution [55]. The application of analytic differentiation to high-speed flow solvers or flow solvers which apply turbulence modelling can also be problematic. The numerical routines in these types of flow solvers can be non-differentiable [56]. Typically, simplifications such as assuming a constant turbulent eddy viscosity is used in these circumstances.

3.4.2 Finite differences

The simplest method for constructing the adjoint operator is via finite differences. A first-order accurate finite difference approximation for one entry of the adjoint operator is given as

$$\frac{\partial \mathbf{R}_i}{\partial \mathbf{Q}_j} = \frac{\mathbf{R}_i(\mathbf{Q}_j + h) - \mathbf{R}_i(\mathbf{Q}_j)}{h}, \quad (3.11)$$

shown for the i^{th} residual vector entry and j^{th} primitive flow variable. This method requires very little modifications to the underlying flow solver routines, making the implementation of this method attractive. However, compared to analytic differentiation, the finite difference approach is computationally expensive, since a flow residual evaluation is required for each flow state variable in every cell. Targeted differentiation is possible to help reduce the computational cost [56–58]. Another deficiency of the finite difference approach is the difficult selection of the perturbation size h . This parameter should be selected such that it is small enough to minimise the truncation error, yet large enough to minimise the finite-precision subtractive error in the numerator.

3.4.3 Complex-Step derivative approximation

The complex-step derivative approximation is similar to the finite difference approach, except now the perturbation is along the imaginary axis in the complex plane [59]. The second-order accurate gradient formula is given as

$$\frac{\partial \mathbf{R}_i}{\partial \mathbf{Q}_j} = \frac{\text{Im}[\mathbf{R}_i(\mathbf{Q}_j + ih)]}{h}. \quad (3.12)$$

It is noted that only one flow residual evaluation is required to obtain a second order accurate approximation. However, since the evaluation now requires complex arithmetic, this typically incurs on the order of twice the computational time, so the cost is roughly the same a second-order accurate finite difference approximation. The primary advantage of the complex-step derivative approximation over the finite difference approach is that there is no subtraction operator. This relaxes the conditions on which the perturbation size (h) must be selected. In practice, this is set to be a very small number (for example $1e^{-50}$) to satisfy truncation error. Similar to finite differences, targeted differentiation can be used to improve the efficiency of this method.

3.4.4 Algorithmic differentiation

Algorithmic differentiation (AD), sometimes referred to as automatic differentiation, is a method of generating source code for evaluating the sensitivities of numerical routines written in a higher-level programming language. The premise is that, no matter how complex the underlying source code is, it will always be constructed of elementary operations such as “+”, “-”, “x”. An AD code essentially applies the chain-rule to differentiate the sequence of elementary operations. A new source code is then built that evaluates the desired sensitivities. AD can be applied in either the forward mode or the reverse mode. The forward mode was shown to be equivalent to the complex-step derivative approximation by Martins [60], however, the complex-step derivative approach has been noted to be simpler to implement and maintain [61, 62]. The reverse mode executes the code forward and then backward, and evaluates the derivatives to a single objective function with respect to N design variables. It is the reverse mode of AD that can be used to automate the application of the discrete adjoint method. In small applications, it is sometimes possible to use AD as a “black-box”, feeding in a nonlinear code and obtaining a corresponding linear perturbation (forward mode AD) or adjoint (reverse mode AD) code. However, typically for large codes applying AD in a targeted or selective manner is necessary to achieve a code that is both computationally efficient and memory efficient [52]. An alternate form of AD is operator overloading. In this approach a new structure that contains both the value of a variable and its derivative are created. All of the existing operators in the program are then overloaded for the new type by redefining the operator instructions. In the overloaded program, the operator processes the value part in the same manner as the original implementation, however, the operator now calculates the derivative using the definition of the derivative of the operator. Operator

overloading is sometimes favoured since it does not require a third-party AD package to implement.

3.5 Hypersonic applications

As a result of its efficiency, the adjoint method has become widely used for aircraft design, evolving to the design optimisation of full wing-body configurations [23, 26]. This section will provide a brief review of adjoint studies that have focused on the design of hypersonic aircraft. First, it is interesting to consider the conclusions made by Giles et al. [63–65] on the topic of what happens to the discrete adjoint system when the underlying nonlinear flow solution has an embedded shock. By considering a fundamental one-dimensional problem with an embedded shock, the authors presented numerical results that indicate it is necessary to smear the shock over a few mesh points to get a convergent adjoint system. A similar result was also observed by Bueno-Orovio et al. [66]. In spite of these results, several high-speed discrete adjoint solvers have been implemented in practical two- and three-dimensional flow codes, as discussed below. One explanation for this apparent contradiction is that the numerical methods employed in high-speed flow solvers are most commonly shock-capturing. Consequently, the captured shocks will be smeared across a few mesh points under these circumstances.

3.5.1 Continuous adjoint method

Several studies have applied the continuous adjoint method to the design of hypersonic aircraft. Loehner et al. [67] performed unconstrained, single-point optimisation of the nozzle on a generic hypersonic vehicle geometry at Mach 8 flight conditions using the continuous adjoint approach. The flow was modelled using the three-dimensional Euler equations. Several more recent works have extended the continuous adjoint framework for hypersonic flows. The work by Copeland et al. [68, 69] implemented the continuous adjoint approach to aerothermodynamic shape design for hypersonic vehicles in thermochemical non-equilibrium flows. The objective functions derived in this work were force-based and the flow was modelled as inviscid. Adjoint gradient verification against a finite difference approach yielded reasonable accuracy. Kline et al. [70–72] has extended the continuous adjoint method to hypersonic inlet design. Over a series of papers, the authors optimised both two- and three-dimensional inlets using the RANS equations with the one-equation Spalart-Allmaras turbulence model [73].

3.5.2 Discrete adjoint method

Perhaps the first hypersonic application of the discrete adjoint method was by Baysal et al. [74]. In this work, a simplified two-dimensional scramjet-afterbody configuration was optimised for maximum

axial thrust. The flow was modelled using the Euler equations. The authors compared optimisation results from both an adjoint and a finite difference implementation. The adjoint approach was shown to require far fewer objective function evaluations, as expected. The adjoint operator was constructed by hand-differentiating the underlying flow solver routines.

Nemec et al. [75] applied the discrete adjoint method to their three-dimensional Euler solver which operated on Cartesian grids. The adjoint operator was also constructed by hand-differentiation, neglecting the non-differentiable limiter function. They performed optimisation of a re-entry capsule at Mach 10 flight conditions, while the objective function was to achieve a lift-to-drag ratio of 0.4. They verified their implementation against central-difference gradients, and reasonable agreement was observed. It was noted that to stabilise the adjoint solver, the second-order reconstruction had to be reduced to first order in some regions of the flow field.

Eyi et al. [76, 77] performed aerothermodynamic optimisation of hypersonic blunt bodies using a discrete adjoint approach. The flow was modelled with the two-dimensional Reynolds-Averaged Navier-Stokes equations coupled with the algebraic Baldwin-Lomax turbulence model and included finite-rate chemistry. The adjoint operator was constructed by hand-differentiation. The objective was to minimise pressure drag on the blunt-body. This work has most recently been extended to include thermal non-equilibrium [78].

As discussed in Section 3.4, it is possible to automate the construction of the adjoint operator. Nielsen et al [56] implemented a discrete adjoint solver using the complex-step approximation. The implementation was verified for two-dimensional, spatially first-order accurate, laminar, hypersonic flows with finite-rate chemistry. Another application of an automated adjoint operator is the work by Marta [79], which presents the ADjoint method (AD refers to the algorithmic differentiation within the adjoint approach). In this work, the fluid was modelled using the Euler equations coupled with Maxwell's equations for application in magnetohydrodynamic flow control. The adjoint gradients were verified against finite differences with good agreement, also, several optimisation examples were presented. The ADjoint approach has also been used for the optimisation of an oblique wing in hypersonic flow by Mader et al. [80]. Again, the flow was modelled using the Euler equations for this work. The adjoint gradients were also verified against finite differences with good agreement.

3.5.3 Context for contributions in this work

Table 3.1 summarises the reviewed work on adjoint-based aerodynamic shape optimisation in hypersonic flow. The novel contribution in this thesis is the implementation of an adjoint solver for high-speed flow that operates on the RANS equations coupled with a two-equation turbulence model. The two-equation turbulence model used in this work is the $k - \omega$ turbulence model. The complex-step derivative approach has been chosen to construct the adjoint operator for the spatially second-order unstructured grid flow solver routines. The use of the complex-step derivative approach provides an

extensible framework that is intended to provide automatic consistency between the flow solver and the adjoint solver. It is superior to the finite difference approach since it does not incur any finite-precision subtractive error, and it is more appropriate than AD for emerging languages, such as the D programming language [81] used in this work, which may not have established AD libraries available.

Table 3.1: Summary of published works on adjoint-based aerodynamic shape optimisation in hypersonic flow.

Researchers	Method	Adjoint operator	Analysis fidelity	Turbulence	Chem.	Grid	Configuration
Baysal et al. [74]	Discrete	Analytic	Euler	-	-	2D Structured	Scramjet-afterbody
Loehner et al. [67]	Continuous	Analytic	Euler	-	-	3D Unstructured	Scramjet-afterbody
Nemec et al. [75]	Discrete	Analytic	Euler	-	-	3D Cartesian	Re-entry capsule
Nielsen et al. [56]	Discrete	Complex	Navier-Stokes	-	Yes	3D Unstructured	Cylinder
Marta [79]	Discrete	AD	MHD (Euler)	-	-	3D Structured	Generic vehicle
Mader et al. [80]	Discrete	AD	Euler	-	-	3D Structured	Oblique wing
Eyi et al. [76–78]	Discrete	Analytic	RANS	Baldwin-Lomax	Yes	2D Structured	Blunt body
Kline et al. [70–72]	Continuous	Analytic	RANS	Spalart-Allmaras	-	3D Unstructured	Hypersonic inlet
Copeland et al. [68, 69]	Continuous	Analytic	Euler	-	Yes	3D Unstructured	Blunted cone & Shuttle wing

CHAPTER 4

FLOW SOLVER

Historically, Eilmer has operated on structured grid topologies using an explicit time-accurate method [82]. To improve efficiency in reaching steady-state solutions and to provide geometrical flexibility, a Newton-Krylov accelerated steady-state solver that operates on unstructured grid topologies has been implemented. This chapter provides the details of the numerical methods employed within Eilmer's unstructured steady-state solver. The core routines of the flow solver are written in the D programming language [81]. More details on the implementation of a compressible-flow code in the D programming language are provided by Jacobs et al. [83].

4.1 Governing equations

4.1.1 Turbulence modelling

The set of equations that govern the behaviour of compressible turbulent flow are the Navier-Stokes equations. It was discussed in Chapter 3 how it is currently too computationally prohibitive for engineering design work to partially or fully resolve the turbulent eddy structures employing LES or DNS methods. For this reason, time-averaging (or Reynolds-averaging as it is referred to) is employed to describe the flow field in terms of mean flow quantities. In this approach, all turbulent eddy structures are modelled. When the governing equations are Reynolds-averaged, a quantity $\phi(t)$ can be considered as the sum of a mean value and a fluctuating part as

$$\phi(t) = \bar{\phi} + \phi', \quad \text{where} \quad \bar{\phi} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \phi \, dt \quad \text{and} \quad \bar{\phi}' = 0, \quad (4.1)$$

where $\bar{\phi}$ is the time- or Reynolds-averaged term, and ϕ' is the fluctuating part. Due to the compressibility of hypersonic flows, a further density-averaging is necessary. This density-weighting is referred to as Favre-averaging.

$$\phi(t) = \tilde{\phi} + \phi'', \quad \text{where} \quad \tilde{\phi} = \frac{1}{\bar{\rho}} \lim_{T \rightarrow \infty} \frac{1}{T} \int_t^{t+T} \rho \phi \, dt = \frac{\overline{\rho \phi}}{\bar{\rho}} \quad \text{and} \quad \overline{\rho \phi''} = 0. \quad (4.2)$$

As in Reynolds-averaging, $\tilde{\phi}$ is the Favre-averaged term, and ϕ'' is the Favre fluctuating part (note that this is different from the Reynolds-averaged fluctuating term). When substituted into the full Navier-Stokes equations, the Reynolds- and Favre-averaged equations produce additional terms within the governing laminar transport equations [84]. These terms, sometimes called the Reynolds stresses and turbulent heat-flux vector [85], cannot be solved analytically and must be modelled by a turbulence model [84]. There exists a vast quantity of turbulence models in the literature, for instance, Roy et al. [86] reviewed 18 turbulence models for their suitability in predicting turbulent hypersonic flows. Despite this, no turbulence model has been shown to provide sufficiently accurate results over a wide range of flow regimes. Furthermore, even when simulating a particular experiment, the ranking of turbulence models based on accuracy is dependent on the chosen metric. As an example, Coratekin et al. [87] found that for hypersonic flow over a compression ramp, the Spalart-Allmaras turbulence model [73] predicted the correct extent of separation, whereas only the $k - \omega$ model [84] predicted sufficiently accurate surface heat transfer rates. Considering this, Wilcox's 2006 $k - \omega$ turbulence model [84] is selected for this current work, primarily due to its proven performance in Eilmer's structured grid explicit solver [88–90].

4.1.2 The RANS flow equations

The differential form of the Reynolds-averaged, Favre-averaged Navier-Stokes equations (referred to herein as the RANS equations) coupled with the $k - \omega$ turbulence model are given as follows:

Mass Conservation:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i} (\bar{\rho} \tilde{u}_i) = 0 \quad (4.3)$$

Momentum Conservation:

$$\frac{\partial}{\partial t} (\bar{\rho} \tilde{u}_i) + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{u}_j \tilde{u}_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} [\bar{t}_{ji} + \bar{\rho} \tau_{ji}] \quad (4.4)$$

Energy Conservation:

$$\begin{aligned} & \frac{\partial}{\partial t} \left[\bar{\rho} \left(\tilde{e} + \frac{\tilde{u}_i \tilde{u}_i}{2} + k \right) \right] + \frac{\partial}{\partial x_j} \left[\bar{\rho} \tilde{u}_j \left(\tilde{h} + \frac{\tilde{u}_i \tilde{u}_i}{2} + k \right) \right] = \\ & \frac{\partial}{\partial x_j} \left[\left(\frac{\mu}{P_{rL}} + \frac{\mu_T}{P_{rT}} \right) \frac{\partial \tilde{h}}{\partial x_j} + \left(\mu + \sigma^* \frac{\bar{\rho} k}{\omega} \right) \frac{\partial k}{\partial x_j} \right] + \frac{\partial}{\partial x_j} [\tilde{u}_i (\bar{\tau}_{ij} + \bar{\rho} \tau_{ij})] \end{aligned} \quad (4.5)$$

Molecular and Reynolds-Stress Tensors:

$$\bar{\tau}_{ij} = 2\mu \bar{S}_{ij} \quad \bar{\rho} \tau_{ij} = 2\mu_T \bar{S}_{ij} - \frac{2}{3} \bar{\rho} k \delta_{ij} \quad \bar{S}_{ij} = S_{ij} - \frac{1}{3} \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \quad (4.6)$$

Eddy Viscosity:

$$\mu_T = \frac{\bar{\rho} k}{\tilde{\omega}} \quad \tilde{\omega} = \max \left\{ \omega, C_{\text{lim}} \sqrt{\frac{2\bar{S}_{ij} \bar{S}_{ij}}{\beta^*}} \right\} \quad C_{\text{lim}} = \frac{7}{8} \quad (4.7)$$

Turbulence Kinetic Energy:

$$\frac{\partial}{\partial t} (\bar{\rho} k) + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{u}_j k) = \bar{\rho} \tau_{ij} \frac{\partial \tilde{u}_i}{\partial x_j} - \beta^* \bar{\rho} k \omega + \frac{\partial}{\partial x_j} \left[\left(\mu + \sigma^* \frac{\bar{\rho} k}{\omega} \right) \frac{\partial k}{\partial x_j} \right] \quad (4.8)$$

Specific Dissipation Rate:

$$\begin{aligned} \frac{\partial}{\partial t} (\bar{\rho} \omega) + \frac{\partial}{\partial x_j} (\bar{\rho} \tilde{u}_j \omega) &= \alpha \frac{\omega}{k} \bar{\rho} \tau_{ij} \frac{\partial \tilde{u}_i}{\partial x_j} - \beta \bar{\rho} \omega^2 + \sigma_d \frac{\bar{\rho}}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \\ &+ \frac{\partial}{\partial x_j} \left[\left(\mu + \sigma^* \frac{\bar{\rho} k}{\omega} \right) \frac{\partial \omega}{\partial x_j} \right] \end{aligned} \quad (4.9)$$

Closure Coefficients:

$$\begin{aligned} \alpha &= \frac{12}{25} & \beta &= \beta_0 f_\beta & \beta^* &= \frac{9}{100} & \sigma &= \frac{1}{2} & \sigma^* &= \frac{3}{5} & \sigma_{d0} &= \frac{1}{8} \\ \beta_0 &= 0.0708 & P_{rT} &= \frac{8}{9} & \sigma_d &= \begin{cases} 0, & \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \leq 0 \\ \sigma_{d0}, & \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} > 0 \end{cases} \\ f_\beta &= \frac{1 + 85\chi_\omega}{1 + 100\chi_\omega} & \chi_\omega &= \left| \frac{\Omega_{ij} \Omega_{jk} \hat{S}_{ki}}{(\beta^* \omega)^3} \right| & \hat{S}_{ki} &= S_{ki} - \frac{1}{2} \frac{\partial \tilde{u}_m}{\partial x_m} \delta_{ki} \end{aligned} \quad (4.10)$$

For this work, the system is closed using the ideal gas equation of state

$$p = \bar{\rho} R \tilde{T}. \quad (4.11)$$

The viscosity coefficient μ is assumed a function of temperature only, and is evaluated by Sutherland's

semi-empirical formula

$$\mu = \mu_{\text{ref}} \left(\frac{T}{T_{\text{ref}}} \right) \frac{T_{\text{ref}} + S}{T + S}, \quad (4.12)$$

where $\mu_{\text{ref}} = 1.716 \times 10^{-5} \text{ kg}\cdot\text{m}^{-1}\cdot\text{s}^{-1}$, $T_{\text{ref}} = 273.15 \text{ K}$, and $S = 111.0 \text{ K}$. Similarly, the thermal conductivity coefficient k is assumed a function of temperature only, and is evaluated by Sutherland's semi-empirical formula

$$k = k_{\text{ref}} \left(\frac{T}{T_{\text{ref}}} \right) \frac{T_{\text{ref}} + S}{T + S}, \quad (4.13)$$

where $k_{\text{ref}} = 2.41 \times 10^{-2} \text{ W}\cdot\text{m}^{-1}\cdot\text{K}^{-1}$, $T_{\text{ref}} = 273.0 \text{ K}$, and $S = 194.0 \text{ K}$.

4.2 Discretisation of the flow equations

4.2.1 Finite Volume Method

The governing equations outlined above are discretised using a cell-centered finite-volume approach. The finite volumes (referred to herein as cells) supported in Eilmer are triangles or quadrilaterals in two-dimensions; and hexahedrals, tetrahedrals, prisms, or pyramids in three-dimensions. A quadrilateral cell is shown in Figure 4.1 with geometric definitions. The finite volume method operates directly on the integral form of the governing equations

$$\frac{\partial}{\partial t} \int_V U dV = - \oint_S (\overline{F}_c - \overline{F}_v) \cdot \hat{n} dA + \int_V S_{\text{turb}} dV, \quad (4.14)$$

where U is the vector of conserved variables,

$$U = \begin{bmatrix} \bar{\rho} \\ \bar{\rho} \tilde{u}_x \\ \bar{\rho} \tilde{u}_y \\ \bar{\rho} \tilde{u}_z \\ \bar{\rho} E \\ \bar{\rho} k \\ \bar{\rho} \omega \end{bmatrix}, \quad (4.15)$$

t is time, V is volume, A is an interfacial area, F_c is the convective flux vector, F_v is the viscous flux vector, and S_{turb} is the turbulence model source term. The residual function for a cell, discretised by the finite volume method, can then be written as

$$R(U) = \frac{dU}{dt} = - \frac{1}{V} \sum_{\text{faces}} (\overline{F}_c - \overline{F}_v) \cdot \hat{n} dA + S_{\text{turb}}, \quad (4.16)$$

where U and S_{turb} now represent cell-average values, and \sum_{faces} denotes integration over the cell interfaces.

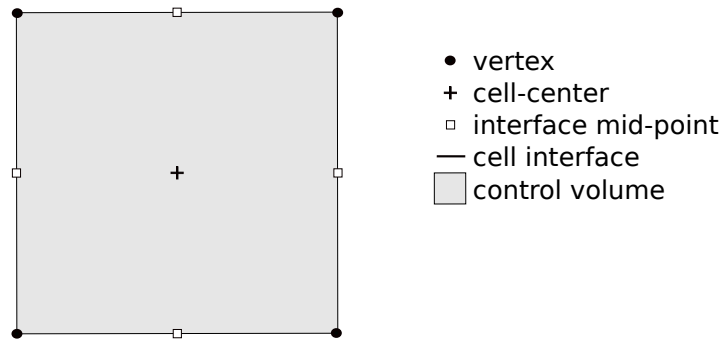


Figure 4.1: Quadrilateral finite volume cell with geometric definitions.

4.2.2 Gradient reconstruction

When applying the finite volume method, gradient reconstruction is required in both the convective flux update (outlined in Section 4.2.3) and the viscous flux update (outlined in Section 4.2.4). Two popular methods for approximating the gradients are Gauss' divergence theorem and the least-squares procedure. The least-squares procedure is chosen for the current work as suggested by several authors who have reported superior performance on unstructured grids [91–94].

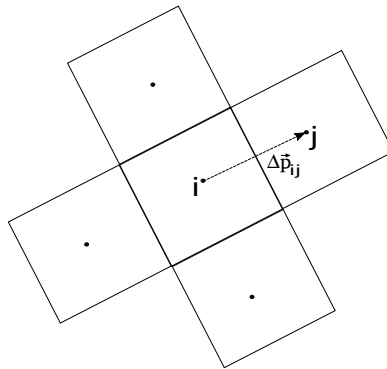


Figure 4.2: Least-squares point cloud stencil.

Consider the cloud of points, j , centred around the i^{th} cell in Figure 4.2. For a primitive flow variable Q_i , the least-squares error term (S) can be defined as

$$S = \sum_{j=1}^N w_j^2 (\nabla \vec{Q}_i \cdot \Delta \vec{p}_{ij} - \Delta Q_j)^2, \quad (4.17)$$

where w_j is the weight applied to the j^{th} cell. To minimise the error over the cloud, the error term is

differentiated with respect to the unknown gradients and set to zero,

$$\frac{\partial \mathcal{S}}{\partial \nabla Q_i} = 0. \quad (4.18)$$

The linear system of equations which form (shown for two dimensions only) can then be solved to estimate the gradients

$$\nabla Q_i = [M]^{-1} \vec{b}, \quad (4.19)$$

where,

$$[M] = \begin{bmatrix} \sum_{j=1}^N w_j^2 \Delta x_{ij} \Delta x_{ij} & \sum_{j=1}^N w_j^2 \Delta x_{ij} \Delta y_{ij} \\ \sum_{j=1}^N w_j^2 \Delta x_{ij} \Delta y_{ij} & \sum_{j=1}^N w_j^2 \Delta y_{ij} \Delta y_{ij} \end{bmatrix}, \quad (4.20)$$

and,

$$\vec{b} = \begin{bmatrix} \sum_{j=1}^N w_j^2 \Delta Q \Delta x_{ij} \\ \sum_{j=1}^N w_j^2 \Delta Q \Delta y_{ij} \end{bmatrix}. \quad (4.21)$$

If the weights are dependent only on the point positions, the least-squares estimates for the gradients can be reduced to a summation of the ΔQ values, with coefficients determined from the equations above. This is the form coded within the flow solver to improve efficiency.

4.2.3 Convective flux

The convective flux through a cell interface is computed by nominally first-order upwinding schemes,

$$F = F(Q_L, Q_R), \quad (4.22)$$

where $F(\cdot)$ is the flux operator, Q_L and Q_R are the cell-centered flow state from the left and right cell respectively. Upwind schemes can be categorised into either flux difference splitting (FDS) or flux vector splitting (FVS) schemes. FDS schemes use an approximate solution to the local Riemann problem, whereas FVS schemes split the flux vector into upstream and downstream travelling components. Several flux calculators are available in Eilmer. The popular FDS scheme by Roe [95, 96], the FVS scheme by Hanel et al. [97, 98], and the hybrid AUSMDV scheme by Liou and Wada [99] have been used in this work. Higher-order accuracy is achieved by reconstructing the cell-centered flow state up to a cell interface before the flux calculation, as outlined in Section 4.2.3.1.

4.2.3.1 Higher-order reconstruction

To achieve second-order spatial accuracy, the flow states at the cell interfaces are calculated by interpolating the cell-centered flow state using a piece-wise-linear scheme,

$$Q_{i+1/2} = Q_i + \phi \cdot \nabla Q_i \cdot \frac{1}{2} \Delta \vec{p}_{ij}, \quad (4.23)$$

where ϕ is some limiting factor (discussed in Section 4.2.3.2), and Q_i is a primitive variable (u , v , w , $\bar{\rho}$, p , k , ω). The gradients of the primitive flow variables (∇Q_i) are approximated using the least-squares procedure outlined in Section 4.2.2. The reconstructed flow states, either side of the interface, are then used for Q_L and Q_R in Equation 4.22. A compact, nearest neighbour stencil is chosen as the least-squares stencil, illustrated in Figure 4.2. Our reconstruction procedure computes a gradient for all internal cells only. For multi-block simulations, this gradient is communicated to the ghost-cells at block connection boundaries such that consistent inviscid fluxes are computed at shared interfaces of these connections. This increases the complexity of the code as a result of the tightly coupled communication between neighbouring blocks. However, failing to do this results in small inconsistencies in flux estimates across boundary interfaces. The compact nature of the stencil is memory efficient and requires only one layer of ghost-cells.

4.2.3.2 Gradient limiting

When reconstructing a flow state with Equation 6.4 in regions near strong shocks, a limiting factor ϕ is required to ensure stability. The limiter acts to prevent new local minima or maxima during the reconstruction process by limiting the computed gradient, and so ϕ is taken to be some positive value less than 1. For unstructured grids, the multidimensional limiters by Barth [100] and Venkatakrishnan [101] are frequently employed. The Venkatakrishnan limiter is preferred for steady-state solvers due to its superior convergence properties [85], and, for this reason, the Venkatakrishnan limiter is used in this work. Details of the version implemented in Eilmer can be found in the textbook by Blazek [85]. The multidimensional limiter procedure (MLP) [102, 103] has also been identified as a promising technique to improve the discontinuity capturing capability and robustness of the unstructured grid solvers near strong shocks [104]. Future work may incorporate it.

A known problem for supersonic and hypersonic flow-codes is that the limiters necessary to ensure stability can oscillate causing a stalling of the convergence. In the literature, this is referred to as limiter ringing [104, 105]. The level of convergence achieved before stalling is problem-dependent. Since the adjoint method is predicated on the assumption that the residual vector has been reduced to machine precision, performing an adjoint solve on a stalled flow field residual can cause the adjoint solver to either diverge or converge on an inaccurate solution. Furthermore, the routines in the flow code that evaluate the limiter values have been observed to be sensitive to complex variable per-

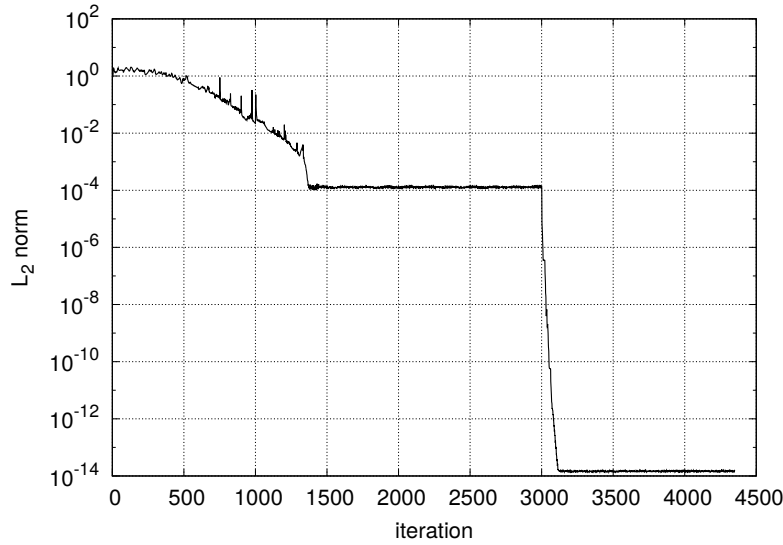


Figure 4.3: Global residual history for simulation with frozen limiter on 3000th iteration.

turbations [105]. This has also been observed in this current work. One approach to overcome this limitation is to freeze the limiter after stalled convergence [101]. In this approach, once convergence has stalled, the limiter value for each cell is frozen, i.e. the limiter value is no longer updated during residual evaluations, noting that the limiter is still applied. Thompson [105] applied this approach in the context of high-speed adjoint optimisation. Limiter freezing is applied in the current work on problems that exhibit stalled convergence. Figure 4.3 presents an example of freezing the limiter. This example is of Mach 10 air flowing over a blunted wedge ¹. It is observed that the stalled residual can drop several orders of magnitude upon freezing the limiter on step 3000.

4.2.3.3 Carbuncle Fix

The AUSMDV and Roe-Pike flux schemes are both known to suffer from the Carbuncle phenomenon [106]. The carbuncle phenomenon manifests itself as a region of non-physical re-circulation behind a strong shock wave. One of several suggested fixes in the literature is to apply a highly dissipative flux scheme in the shock region whilst employing a low dissipation flux scheme elsewhere. This was suggested by Wada et al [98], where the Roe flux scheme was paired with the HLLE scheme [107, 108] and the AUMSDV flux scheme was paired with with scheme by Hanel et al. [97, 98]. To switch between the flux schemes, a simple shock detector based on a measure of the relative change in normal velocity at interfaces is used. Specifically, we indicate a strong compression at cell-interface $i + 1/2$ when

$$\frac{u_{n,i+1} - u_{n,i}}{\min(a_{i+1}, a_i)} < \text{tol}, \quad (4.24)$$

¹More details for this simulation are provided in Appendix B.2.

where u is the gas velocity, a is the gas speed of sound, and tol is a compression tolerance, typically set at -0.05 . This shock detector is favoured over a pressure-based detector, since the latter can be triggered by compression waves that are not shock waves, such as in boundary layers [98]. An example of the employed fix is shown in Figure 4.4. Here a carbuncle arises when simulating Mach 8 inviscid air flow over a cylinder² using the AUSMDV flux scheme. By applying the Hanel flux scheme in regions where Equation 4.24 is triggered, the carbuncle is completely avoided. It is noted that we only observe the carbuncle phenomenon with the unstructured grid solver. We have not observed the carbuncle phenomenon using the structured grid solver within Eilmer.

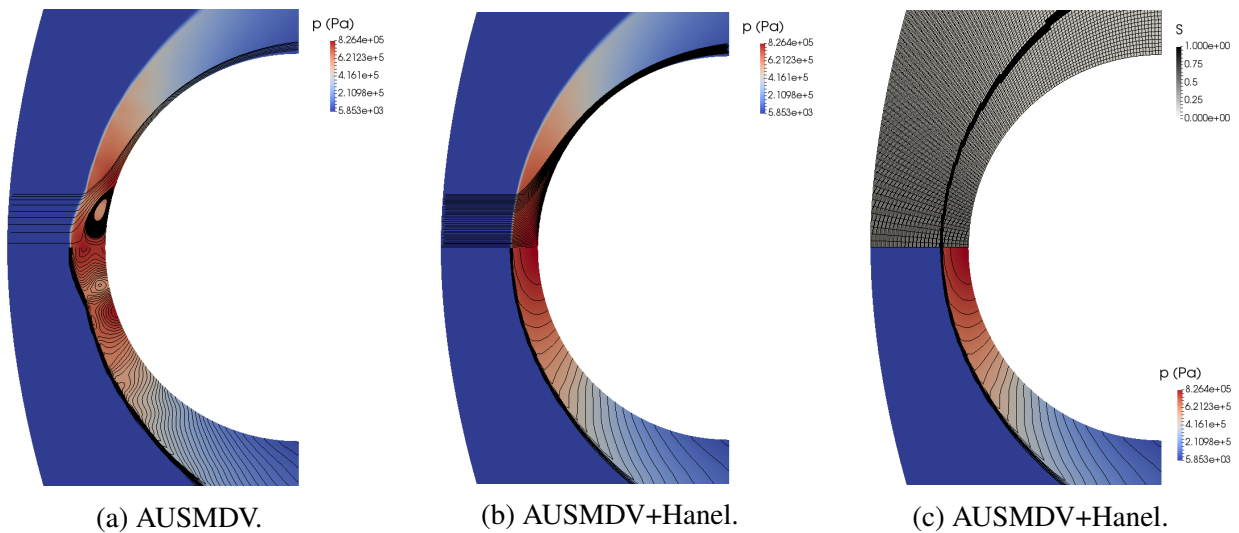


Figure 4.4: Carbuncle fix example. For figure (c) $S = 1$ denotes cells in which the Hanel flux calculator is employed; $S = 0$ denotes cells in which the AUSMDV flux calculator is employed.

4.2.4 Viscous flux

To calculate the viscous flux, the flow variables and their gradients are required at the cell interface mid-points. The interface flow state is simply the average of the left and right cell-centered values. Although it is possible to also average the flow gradients in a likewise manner, this is known to produce an oscillatory checker-board pattern [109, 110]. Several studies [91, 92, 104, 111] have shown that the augmented-face face-tangent method by Haselbacher et al. [109, 110] is a preferred method of averaging the gradients. The spatial gradients at the cell interfaces, computed using the augmented-face face-tangent method, are given as

$$(\nabla Q)_f = \overline{\nabla Q}_f - \left[\overline{\nabla Q}_f \cdot \hat{\Delta p}_{ij} - \frac{Q_j - Q_i}{|\hat{\Delta p}_{ij}|} \right] \left(\frac{\hat{n}_f}{\hat{n}_f \cdot \hat{\Delta p}_{ij}} \right), \quad (4.25)$$

²More details for this simulation are provided in Appendix B.3.

with reference to Figure 4.2 and where $\overline{\nabla Q_f}$ is the average of the left and right cell-centered gradients evaluated using the weighted least-squares approach outlined in Section 4.2.2. An inverse distance weighting was applied in this work. The stencil of points used in the least-squares method here are the cell-center and the surrounding interface mid-points.

4.3 Temporal discretisation

4.3.1 A globalised Newton-Krylov method

A Newton-Krylov method is used to accelerate the flow solution to the steady state, i.e. $\mathbf{R}(\mathbf{U}) = 0$. A Newton iteration can be derived by applying a multivariate Taylor expansion about a current point \mathbf{U}^n :

$$\mathbf{R}(\mathbf{U}^{n+1}) = \mathbf{R}(\mathbf{U}^n) + \mathbf{R}'(\mathbf{U}^n)(\mathbf{U}^{n+1} - \mathbf{U}^n) + \text{higher order terms.} \quad (4.26)$$

Then by setting the right-hand side to zero and neglecting the higher-order terms, the Newton iteration is obtained as

$$\mathbf{R}'(\mathbf{U}^n)\Delta\mathbf{U}^n = -\mathbf{R}(\mathbf{U}^n), \quad \mathbf{U}^{n+1} = \mathbf{U}^n + \Delta\mathbf{U}^n, \quad n = 0, 1, \dots \quad (4.27)$$

Where $\mathbf{R}' = \frac{\partial \mathbf{R}}{\partial \mathbf{U}}$ is the flow Jacobian. A disadvantage of Newton's method is the lack of convergence robustness. In practice, globalisation strategies that lead from an initial iterate into the region of convergence of Newton's method around the desired root are required [112]. Standard globalisation strategies such as line search or trust-region methods often stagnate at local minima. This can be particularly problematic when the solution has complex features such as shocks not present in the initial condition [113]. The pseudo-transient continuation is more favourable for these types of problems since it does not require a reduction in the global residual at each step [113]. In practical terms, Knoll et al. [112] describe it as being able to "climb hills." Pseudo-transient continuation can be viewed as time-marching to the steady-state solution and is applicable when the residual is time-independent and the eigenvalues of its Jacobian have negative real parts [114]. When choosing a time-marching method to globalise Newton's method, the implicit-Euler time-marching scheme is attractive, because it is unconditionally stable, and its time linearisation becomes Newton's method in the limit of infinite step sizes [115]. An implicit-Euler scheme can be derived by first rewriting Equation 4.14 in semi-discrete form using the discrete residual given as Equation 4.16.

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{R}. \quad (4.28)$$

Using implicit-Euler time-integration, Equation 4.28 can be written in fully discrete form as

$$\frac{\Delta \mathbf{U}^n}{\Delta t} = \mathbf{R}^{n+1}, \quad \text{where } \Delta \mathbf{U}^n = \mathbf{U}^{n+1} - \mathbf{U}^n, \quad (4.29)$$

and Δt is the discretised time increment. Since we do not know \mathbf{R}^{n+1} , Equation 4.29 is linearised in time as

$$\frac{\Delta \mathbf{U}^n}{\Delta t} = \mathbf{R}^n + \frac{\partial \mathbf{R}^n}{\partial \mathbf{U}} \Delta \mathbf{U}^n, \quad (4.30)$$

this may then be rearranged to recover the implicit-Euler time marching iterate,

$$[\mathbf{A}]^n \Delta \mathbf{U}^n = \left\{ \frac{1}{\Delta t} \mathbf{I} - \frac{\partial \mathbf{R}^n}{\partial \mathbf{U}} \right\} \Delta \mathbf{U}^n = \mathbf{R}^n. \quad (4.31)$$

Note that as $1/\Delta t$ approaches 0, Newton's method (Equation 4.27) is recovered. In practice it is unnecessary to solve the linear system at each step exactly, instead an inexact-Newton's method is applied [112, 115, 116], where an update to the linear system is sought that satisfies

$$\|\mathbf{R}^n + [\mathbf{A}]^n \Delta \mathbf{U}^n\| \leq \eta \|\mathbf{R}^n\|, \quad (4.32)$$

where $\eta \in [0, 1)$ is called the forcing parameter. Although several methods are available for computing the forcing parameter [117], it is typical to use a constant forcing parameter [118]. The time step at each iteration is updated corresponding to a reduction of the residual norm over a step [85, 112],

$$\Delta t_n = \Delta t_{n-1} \left(\frac{\|\mathbf{R}(\mathbf{U}^{n-1})\|}{\|\mathbf{R}(\mathbf{U}^n)\|} \right)^a, \quad (4.33)$$

where a is some positive number, typically chosen as 1. This formulation allows the time-step to grow and reduce depending on the convergence history.

The Krylov term in the name Newton-Krylov comes from the use of a Krylov subspace linear solver for solving the linear system (Equation 4.31) arising at each Newton iteration. Several Krylov subspace methods exist [119, 120]. The Generalised Minimal RESidual (GMRES) algorithm [121] has been a favourable selection for CFD solvers [112, 122–125]. A primary factor in the popularity of GMRES is that the algorithm only requires a matrix-vector product, which may be approximated by a Fréchet derivative [126], this will be discussed further in Section 4.3.3. This suggests that the method can be matrix- or Jacobian-free. However, it will be shown in Section 4.3.4 that some approximation of the Jacobian is still required as a preconditioner to the linear system. The GMRES algorithm employed in the flow solver will be presented in Section 4.3.5.

4.3.2 Linear system scaling

The linear system of equations presented as Equation 4.31 are poorly scaled. For example, for turbulent flows, the turbulence variables can have residual components that are several orders of magnitude larger than the flow state variables. As a consequence of solving the linear system inexactly, the poor

scaling can lead to sub-optimal convergence or even divergence since the system can be dominated by the largest residual component [118]. To improve the convergence characteristics of Equation 4.31, residual vector (row) and solution vector (column) scaling is recommended [122]. The row and vector scaling outlined by Brown et al. [127] is applied to Equation 4.31 in this work. The scaled system is given as

$$[\tilde{\mathbf{A}}]\Delta\tilde{\mathbf{U}} = \tilde{\mathbf{R}}, \quad \text{where } [\tilde{\mathbf{A}}] = \mathbf{D}^{-1}[\mathbf{A}]\mathbf{D}, \quad \Delta\tilde{\mathbf{U}} = \mathbf{D}^{-1}\Delta\mathbf{U}, \quad \tilde{\mathbf{R}} = \mathbf{D}^{-1}\mathbf{R}. \quad (4.34)$$

The scaling matrix \mathbf{D} , which is a diagonal matrix, is made up of typical values for $\Delta\mathbf{U}$ [127, 128]. For the current work, this is chosen to be the absolute value of the maximum time rate of change for each conserved variable at the current time step. To avoid scaling by values close to zero, when the time rates of change are very small, we limit the absolute value of the entries in \mathbf{D} to be 1.

4.3.3 Matrix-free approach

Explicit formation of the flow Jacobian, $\frac{\partial \mathbf{R}^n}{\partial \mathbf{U}}$, in Equation 4.31 requires linearisation of the spatially second-order flow routines. This requires excessive storage and is computationally expensive to compute [129]. An advantage of GMRES over other linear solvers is that it requires only matrix-vector products, which may be approximated by a directional difference or Fréchet derivative [126]. The Fréchet derivative is given as

$$\mathbf{J}\mathbf{v} \approx \frac{[\mathbf{R}(U + h\mathbf{v}) - \mathbf{R}(U)]}{h}, \quad (4.35)$$

where \mathbf{v} is a directional vector, h is the perturbation parameter, \mathbf{R} is the residual vector, and \mathbf{J} is the matrix operator, in this case the flow Jacobian. A difficulty of applying Equation 4.35 directly is the choice of the perturbation parameter h . Considering truncation error, h should be chosen sufficiently small. However, if h is too small, then the round-off error from subtracting two large, almost equal numbers can degrade the accuracy of the approximation [130]. Erroneous matrix-vector products resulting from poor selection of h can lead to GMRES converging on the incorrect linear system solution at each step as discussed by Zingg et al. [131]. To relieve the difficulty of selecting an appropriate perturbation size, and to improve the accuracy of the evaluated derivatives, a complex-step approximation to the Fréchet derivative may be applied [132]. The complex-step approximation to the Fréchet derivative is written as

$$\mathbf{J}\mathbf{v} \approx \frac{\text{Im}[\mathbf{R}(\mathbf{U} + h\mathbf{v}i)]}{h}. \quad (4.36)$$

In the complex-variable approach, no explicit differencing is involved, and so the perturbation parameter may be chosen sufficiently small to solely control the truncation error. A comparison of the real and complex variable Fréchet derivative for several choices of h is shown in Figure 4.5. Here

the convergence history for an inviscid simulation of Mach 2 air flowing over a ramp is presented ³, this problem was originally published by Marques et al. [133]. From Figure 4.5, it is observed that the level of convergence achievable is strongly dependent on the selection of h for the real-valued Fréchet derivative. The complex-valued Fréchet derivative does not show the same dependence. For turbulent simulations, it was found throughout this work that the complex-valued Fréchet derivative was necessary to achieve suitable convergence.

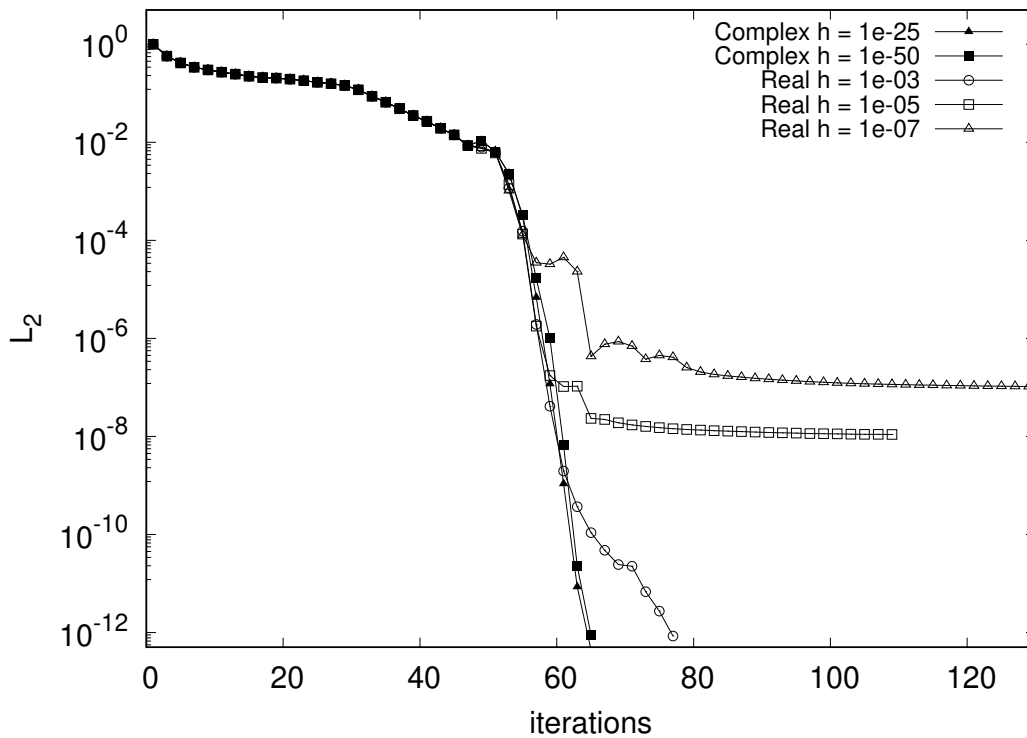


Figure 4.5: Effect of h selection for a supersonic ramp simulation.

4.3.4 Preconditioning

The Krylov subspace methods tend to suffer from slow convergence for problems which arise from fluid dynamics [119]. In practice, methods such as GMRES require some form of preconditioning to achieve fast, robust convergence. Preconditioning transforms the system by either pre- or post-multiplying by a matrix which approximates \mathbf{A} . The former is referred to as left-preconditioning, and is applied as given

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{x} = \mathbf{P}^{-1}\mathbf{b}. \quad (4.37)$$

³More details for this simulation are provided in Appendix B.1.

In left-preconditioning, the residual of the system is changed. This can be avoided by applying post-multiplying or right-preconditioning,

$$\mathbf{A}\mathbf{P}^{-1}\mathbf{P}\mathbf{x} = \mathbf{b}. \quad (4.38)$$

Right-preconditioning is used in this work. The precondition matrix \mathbf{P}^{-1} should resemble \mathbf{A}^{-1} as much as possible. In practice, \mathbf{P} is typically chosen to be a reduced-order approximation of \mathbf{A} which is then decomposed by some approximate factorisation. The factorisation of \mathbf{P} enables fast inversion when performing the preconditioning. Selection of a suitable factorisation is challenging. In a review of factorisation methods for CFD applications, Saad et al. [134] found that no single method was superior to the rest. For hypersonic flows, a zero-fill Incomplete Lower-Upper (ILU) factorisation of an approximate first-order flow Jacobian has shown good results for inviscid, laminar and turbulent flows [135–137]. An ILU factorisation of a matrix is a sparse approximation of the LU factorisation; zero-fill denotes that the sparsity pattern of the ILU factorisation is that of the original matrix. Higher levels of fill are possible, typically denoted as ILU(p), where p is the level of fill. As the level of fill increases, the sparsity pattern becomes less like the original matrix and more like the exact factorisation. However, the added accuracy of the factorisation is balanced by the additional memory and computational expense. Several authors [122–125] have found higher fill levels, ranging from 2 through to 4, to provide accelerated convergence for the simulation of subsonic airfoils. The effect of increasing the level of fill for a supersonic test case was briefly studied as part of this current work. Figure 4.6 shows the convergence history for the inviscid supersonic ramp test case first introduced in Section 4.3.3. It is observed that the additional level of fill does not provide significant performance improvement over the ILU0 preconditioner. As a consequence, for this work, the ILU0 preconditioner is used.

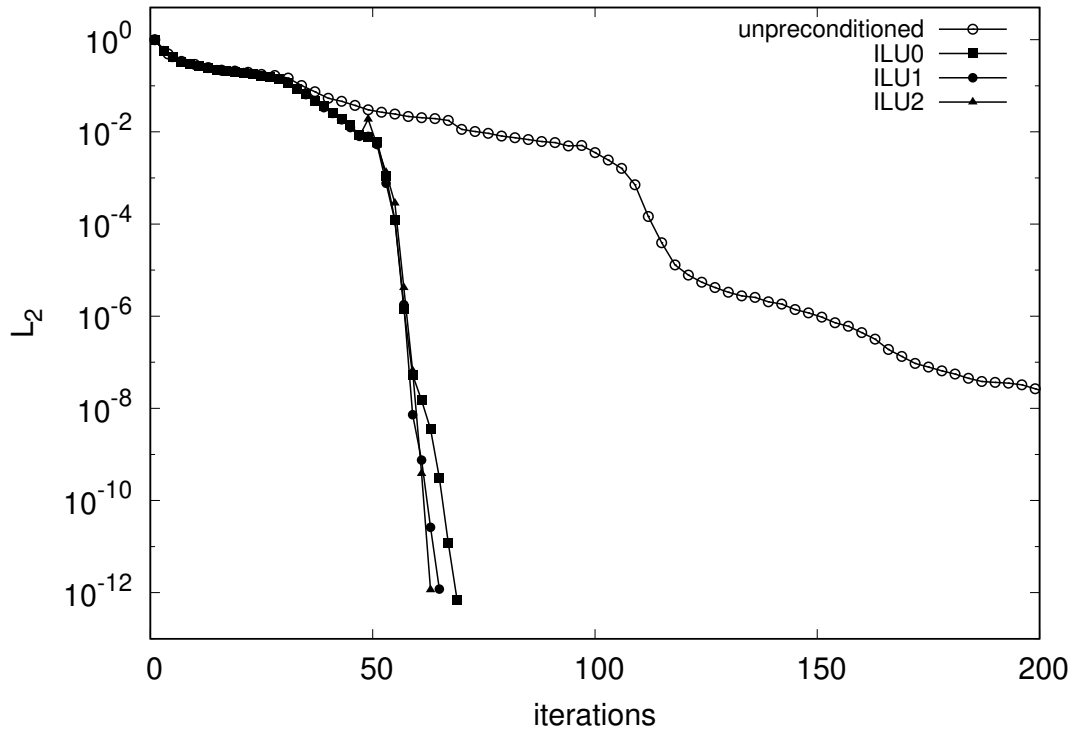


Figure 4.6: Comparison of level of fill for a supersonic ramp simulation.

The ILU preconditioner can be implemented in either a scalar or block flavour. In the block approach, the LU factorisation is applied per block, where the block structure is taken to be the natural block structure of the flow Jacobian. This is referred to as block-fill ILU or BFILU. Orkwis [138] found that BFILU was able to converge problems that the scalar approach stalled on. Pueyo [123] found that, by treating the zero entries within a block as if they were non-zeroes, the BFILU can be constructed via a scalar algorithm. This reduces the complexity of the code and is the approach applied in the current work. For domain decomposed problems, an additive-Schwarz [139] approach with no overlapping is employed.

4.3.5 GMRES algorithm

The scaled right-preconditioned restarted GMRES algorithm implemented in Eilmer is presented in Figure 4.7. In this work, we have chosen to precondition first and scale second. Note that, in the calculation of $\tilde{J}\tilde{v}_j$, the Jacobian matrix ($J = \frac{\partial \mathbf{R}}{\partial \mathbf{U}}$) is never explicitly formed. The product is calculated by first forming $z = \mathbf{D}^{-1}\tilde{v}_j$, then solving the linear system $\mathbf{P}w = z$, evaluating Jw via the complex Fréchet derivative, and then finally multiplying the result of the matrix-vector product by \mathbf{D} . In practice, the routines used to construct the adjoint operator⁴, $(\frac{\partial \mathbf{R}}{\partial \mathbf{Q}})^T$, are used to build the approximate low-order flow Jacobian used as the precondition matrix. This requires the application of a transform

⁴Details are provided in Section 6.2

from the primitive to conservative flow state variables, details of this transform matrix are provided in Appendix C. To remove unnecessary computational expense, the ILU decomposition of the transposed approximate low-order flow Jacobian is constructed, and so, it is \mathbf{P}^T that is actually stored in memory using compressed row storage format (CRS) [119]. Consequently, the preconditioning stage requires the solution of $(\mathbf{P}^T)^T w = z$ via a CRS-based factorisation transpose solve. The algorithm for performing this, which is implemented in Eilmer, is outlined by Dongarra [140]. The remainder of the GMRES algorithm resembles the standard method as outlined by Saad [119, 121].

1. Start:

- Choose x_0
- Compute

$$r_0 = b - Ax_0.$$

$$\tilde{r}_0 = Dr_0.$$

$$\tilde{v}_1 = \frac{\tilde{r}_0}{\|\tilde{r}_0\|_2}.$$

2. Iterate:

- For $j = 1, \dots, m$ do

$$\tilde{J}\tilde{v}_j = D(J(P^{-1}(D^{-1}\tilde{v}_j))).$$

$$\tilde{h}_{i,j} = (\tilde{J}\tilde{v}_j), \quad i = 1, 2, \dots, j.$$

$$\tilde{v}_{j+1} = \tilde{J}\tilde{v}_j - \sum_{i=1}^j \tilde{h}_{i,j}\tilde{v}_i.$$

$$\tilde{h}_{j+1,j} = \|\tilde{v}_{j+1}\|_2, \quad \text{and}$$

$$\tilde{v}_{j+1} = \tilde{v}_{j+1}/\tilde{h}_{j+1,j}.$$

3. Solve the least-squares problem:

- Define \tilde{H}_m to be the $(m+1) \times m$ (Hessenberg) matrix whose nonzero entries are the coefficients \tilde{h}_{ij} , $1 \leq i \leq j+1$, $1 \leq j \leq m$ and $\tilde{V}_m \equiv [\tilde{v}_1, \dots, \tilde{v}_m]$.
- Find the vector \tilde{y}_m which minimises $\|\tilde{\beta}e_1 - \tilde{H}_m\tilde{y}\|_2$ over all vectors \tilde{y} in \mathbb{R}^m .

4. Form approximate solution:

- Compute $x_m = x_0 + P^{-1}D^{-1}\tilde{z}^m$, where $\tilde{z}^m = \tilde{V}_m\tilde{y}_m$.

5. Restart:

- Compute $r_m = b - Ax_m$, if satisfied then stop,
- else $x_0 \leftarrow x_m$ Go-to 1.

Figure 4.7: A scaled right-preconditioned restarted GMRES algorithm.

4.4 Boundary Conditions

Each boundary of a computational domain in this work is modelled as either an inflow, outflow, slip-wall, or no-slip wall boundary condition. The boundary conditions are handled by attaching a ring of ghost cells, one cell deep, around the outer edges of the computational domain. The inviscid-component of an applied boundary condition is implemented by filling the ghost-cell data and then

applying the normal reconstruction and flux calculation without further discrimination of the boundary cells. For viscous simulations, the boundary interface flow state is also specified as appropriate for the particular boundary condition. The boundary values data are used in the derivative calculations that subsequently feed into the viscous fluxes.

The inflow boundary condition employed in the flow solver is a supersonic inflow boundary condition. For this inflow, the ghost cell is filled in with the user-specified freestream flow state. The outflow boundary condition is an extrapolate outflow boundary condition. For this outflow, the ghost-cell is filled in with a first-order extrapolation of the interior flow state. The interface flow state for the inflow and outflow boundary conditions are calculated similar to the interior interfaces, i.e. by averaging the flow state on either side of the interface. The slip- and no-slip wall boundary conditions each fill the ghost cell with a copy of the internal flow state, the normal velocity is then reflected with respect to the boundary surface to model the wall effect. The interface flow state for the slip wall is calculated by the same averaging as the inflow and outflow boundary conditions. For the no-slip wall boundary condition, a zero velocity is applied. Additionally, either a fixed temperature is set in the isothermal case, or a copy of the interior flow state temperature for adiabatic walls. There is also a user-defined boundary condition that allows the user to specify the ghost-cell data and boundary interface data via user-written functions. This particular boundary condition is used in the verification test cases presented in Chapter 5.

4.5 Parallelisation

The computational domain may be decomposed into several partitions of cells, referred to as blocks, for parallel processing of the calculations. For this work, domain decomposition of the computational grid is achieved via METIS[141]. The decomposition of the domain generates extra interior boundaries along block-block connections. To accommodate the transferral of data between blocks separated by these interior boundaries, each block has a buffer region, one cell deep, along block-block boundaries. The buffer region contains ghost cells, which are used to hold a copy of the flow information including the gradients required for the flowstate reconstruction and viscous flux calculation, from adjacent blocks. For a boundary common to two blocks, the ghost cells in the buffer region of each block overlap the active cells of the adjacent block. The only interaction that occurs between blocks is the exchange of boundary data. To achieve second-order spatial accuracy on domain decomposed simulations, data is exchanged three times: (1) the flow state is exchanged prior to the reconstruction phase; (2) the convective gradients are exchanged prior to the inviscid flux calculation; and (3) the spatial gradients are exchanged prior to the viscous flux calculation.

Although it is possible to use an MPI library for parallelism from D, as a first pass, we have chosen to use the shared-memory parallelism offered directly by the D compiler. We employ block-level parallelism, whereby each block may have its partition of cells operated on simultaneously.

Although the use of a thread-pool to implement the parallel updates for the blocks means we need to run parallel jobs on a multi-core computer, typical engineering workstations have enough compute resources to allow for significant acceleration as compared to a single core calculation. For example, the simulations presented in Chapters 5 and 8 were executed on a 20-core, 128GB RAM system.

4.6 Summary

To summarise, the flow solver developed for this work extends the open-source, compressible flow CFD code, Eilmer [83]. The flow solver is written in the D programming language [81] and uses a finite-volume method to solve the Euler, Navier-Stokes or Reynolds-Averaged Navier-Stokes equations on unstructured body-fitted grids using A Jacobian-Free Newton-Krylov method for acceleration to steady-state [142]. The convective fluxes are computed using the AUSMDV scheme by Liou and Wada [99]. Second-order spatial accuracy for convective fluxes is achieved by reconstructing the primitive flow state variables $(u, v, \rho, p, k, \omega)$ within the finite-volumes using an unweighted least-squares method with a nearest face neighbour stencil [143]. To ensure stability when reconstructing the flow state in finite-volumes near strong shocks, the limiter by Venkatakrishnan is employed [144]. The spatial gradients required at the cell interface mid-points for computing the viscous fluxes are computed using a cell-centered weighted least-squares method and the face-tangent augmented face-gradient method [109]. The Reynolds stresses in the RANS equations are modelled using Wilcox's 2006 $k - \omega$ turbulence model [84]. The system is closed using the ideal gas equation of state, and the viscosity and thermal conductivity coefficients are evaluated using the Sutherland semi-empirical formulae. A preconditioned, restarted GMRES method is used to solve the system of linear equations arising when solving for the Newton steps [119]. Robust preconditioning is achieved by an ILU[0] decomposition of an approximate flow Jacobian constructed by only considering a nearest-face-neighbour stencil.

FLOW SOLVER VERIFICATION AND VALIDATION

Having discussed the formulation of the flow solver in Chapter 4, this chapter will document the verification and validation of the flow solver, with the goal to show that it is suitable for high-speed flow analysis. Boehm [145] and Blottner [146] define verification as “solving the equations right” and validation as “solving the right equations” [147]. In this work, several methods of verification are employed to examine the accuracy of the implemented numerical methods. The Method of Manufactured solutions is used to quantitatively verify the order of accuracy of the core routines of the flow solver. The shock-capturing and boundary layer resolving capabilities of the flow solver are qualitatively verified by comparison to analytical and numerical solutions, respectively. An extensive set of validation cases has been taken from the literature to establish the appropriateness of the implemented physical models for design analysis in high-speed flow. The selected validation cases cover the spectrum of relevant flow phenomena outlined in Chapter 2, in particular, high-speed flows with strong inviscid/viscous interactions. These include several fundamental configurations which are well established in the literature: such as flat plates, cones, and compression corners. In addition to these fundamental cases, a representative hypersonic vehicle geometry is also presented.

5.1 Overview of flow solver settings

The following flow solver settings have been used in the verification and validation studies presented in this chapter unless otherwise stated.

Table 5.1: General flow solver settings for verification and validation cases.

Solver Type	Newton-Krylov
Preconditioner	ILU0
Grid Type	Unstructured
Grid Generation	Pointwise™
Grid Partitioning	METIS™
Flux Calculator	AUSMDV
Reconstruction	Second-order
Limiter	Venkatakrishnan ($K = 0.3$)

5.2 Verification

5.2.1 Method of Manufactured Solutions

The Method of Manufactured Solutions is a code verification technique that can be used to assess the order of accuracy of a computational fluid dynamics code. First proposed by Roache and Steinberg [148], the method allows one to choose a purely manufactured analytic solution which is fed through the governing partial differential equations to obtain the source terms which would, in turn, generate the manufactured solution. The solver is then employed to simulate the derived source terms with exact Dirichlet boundary conditions applied to all boundaries by filling the ghost cells with values from the analytic solution. As grid refinement increases, the solution generated by the flow solver should approach the analytic solution in all cells. The manufactured solutions method is a more rigorous approach than simply using exact solutions because the manufactured solution can be constructed to exercise all terms in the governing equations. In theory, on an infinitely refined mesh, the flow solver solution should approach the analytic solution. In practice, one calculates an observed order of convergence of the discretisation error. This is then compared to the actual order of convergence of the numerical model. To determine the observed order of accuracy, error terms from successive grid refinements are compared. Obkorkampf and Roy's textbook [149] presents the following equation for estimating the order of accuracy,

$$p = \frac{\ln\left(\frac{\varepsilon_{k+1}}{\varepsilon_k}\right)}{\ln(r)}, \quad (5.1)$$

where ε_{k+1} is the error at the coarse level and ε_k is the error at the fine level, and,

$$r = \left(\frac{N_k}{N_{k+1}}\right)^{1/d}. \quad (5.2)$$

Here the refinement factor, r , is defined as the ratio of the number of cells in the fine mesh (N_k) and the coarse mesh (N_{k+1}) raised to the power of $1/d$, where d is the dimension (i.e. 2 for two-dimensions).

Any of the error norms may be used to determine the order of accuracy. Here we have chosen to use the L_2 norm, defined as,

$$\varepsilon = \|f - f_{\text{ref}}\|_2 = \left(\frac{1}{N} \sum_{n=1}^N |f_n - f_{\text{ref},n}|^2 \right)^{1/2}. \quad (5.3)$$

Here f is any flow field value of interest, for example, density (ρ). f_n is the numerical value at cell n and $f_{\text{ref},n}$ is the manufactured value evaluated at cell centroid coordinates. The Method of Manufactured Solutions and its importance to the verification of CFD codes is widely known and is a strongly recommended method for code verification by the AIAA Committee on Standards [150]. Further details on the methodology can be found in the comprehensive textbook by Oberkampf and Roy [149].

5.2.1.1 Baseline Manufactured Solutions

The Manufactured Solutions employed for this work all take the following form,

$$\begin{aligned} \phi(x, y, z) = & \phi_0 + \phi_x f_{s_x} \left(\frac{a_{\phi_x} \pi x}{L} \right) + \phi_y f_{s_y} \left(\frac{a_{\phi_y} \pi y}{L} \right) + \phi_z f_{s_z} \left(\frac{a_{\phi_z} \pi z}{L} \right) \\ & + \phi_{xy} f_{s_{xy}} \left(\frac{a_{\phi_{xy}} \pi xy}{L^2} \right) + \phi_{yz} f_{s_{yz}} \left(\frac{a_{\phi_{yz}} \pi yz}{L^2} \right) + \phi_{zx} f_{s_{zx}} \left(\frac{a_{\phi_{zx}} \pi zx}{L^2} \right), \end{aligned} \quad (5.4)$$

where $\phi = [\rho, u, v, w, p, k, \omega]^T$ represents any of the primitive variables and the $f_s(\cdot)$ functions represent sine or cosine functions [151]. The generation of the source terms is accomplished by substituting the analytic solution into the respective governing equations. Since the task is non-trivial to do by hand, this is typically done using a computer algebra system, such as SymPy [152], used in this work. The specific constants and forms of f_s found in Equation 5.4, for each particular manufactured solution, can be found in Appendix A.

5.2.1.2 Two-dimensional flow solver

The two-dimensional flow solver has been verified on uniform quadrilateral cell grids. The levels of grid refinement used in the verification are presented in Table 5.2, and an example of the mesh is presented in Figure 5.1. The meshes for the MMS verification have been generated using Eilmer's native grid generator and applied manual grid partitioning. We have used a 16 block arrangement (4 blocks in the x-direction, and 4 blocks in the y-direction) for the simulations presented in this work.

The chosen analytical solutions for the Euler and Navier-Stokes equations were first presented by Roy et al. [153]. They take the form of Equation 5.4, where the ϕ_z , ϕ_{yz} , ϕ_{zx} terms are all 0; the remaining constants used are presented in Table A.1 and A.3, and the f_s functions are presented in Tables A.2 and A.4, for the Euler and Navier-Stokes equations respectively. For the Euler test, an

Table 5.2: Levels of grid refinement used for two-dimensional verification.

Grid	Dimensions	Cell width, Δx (m)
G1	8×8	0.125
G2	16×16	0.0625
G3	32×32	0.03125
G4	64×64	0.015625
G5	128×128	7.8125×10^{-3}

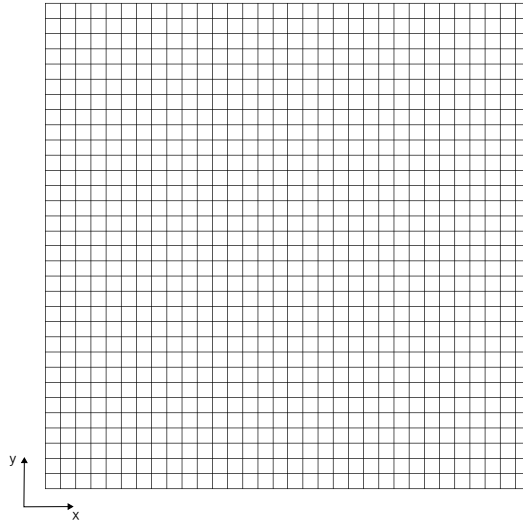


Figure 5.1: Example grid for two-dimensional MMS simulations.

inviscid supersonic flow is simulated on the uniform grid. For the Navier-Stokes test, a subsonic flow of a viscous gas is simulated on the same uniform grid. The gas is modelled as a calorically perfect gas with $\gamma = 1.4$, $R = 287.0 \text{ J}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$, $\mu = 10.0 \text{ Pa}\cdot\text{s}$, and $Pr = 1.0$. The results from the two-dimensional Euler and Navier-Stokes tests are presented in Figures 5.2(a), 5.2(b), 5.2(c), and 5.2(d).

The chosen analytic solution for the RANS equations was first presented by Roy et al [154]. It takes the form of Equation 5.4, where the ϕ_z , ϕ_{yz} , ϕ_{zx} terms are all 0: the remaining constants used are presented in Tables A.5, and the f_s functions are presented in Table A.6. For the RANS test, a subsonic flow of a viscous gas is simulated on the uniform grid. The gas is modelled as a calorically perfect gas with $\gamma = 1.4$, $R = 287.0 \text{ J}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$, $\mu = 10.0 \text{ Pa}\cdot\text{s}$, $Pr = 1.0$, and $Pr_T = 0.89$. The results from the two-dimensional RANS tests are presented in Figures 5.2(e), 5.2(f).

As expected, the error is seen to decrease with successive grid refinements for all two-dimensional test cases. Furthermore, all sets of results observe second-order convergence, consistent with the second-order accurate numerical methods implemented.

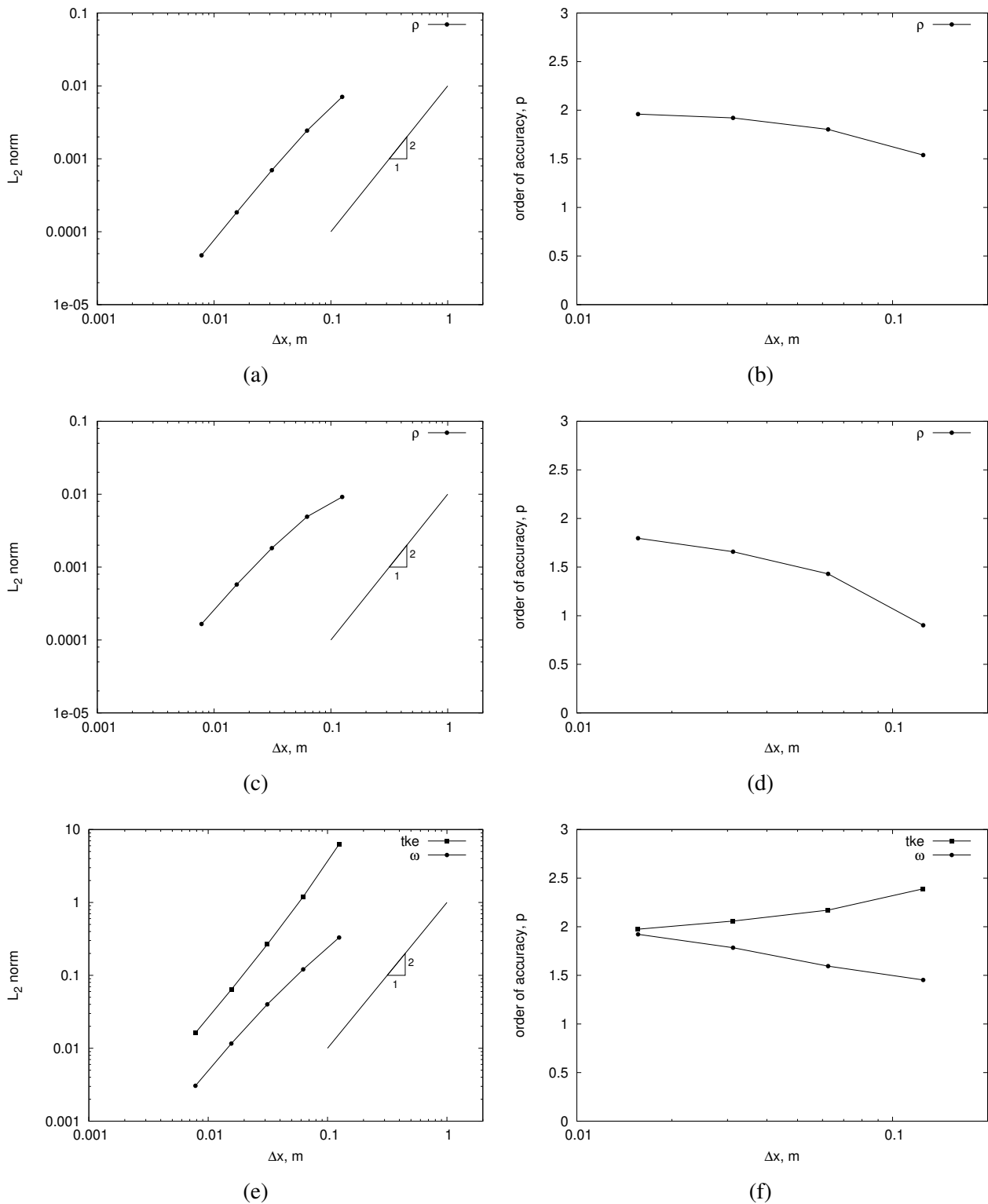


Figure 5.2: Two-dimensional MMS results for the Euler, Navier-Stokes, and RANS equations: (a) Euler flow solution L_2 error norm for density (ρ), (b) Order of accuracy for the Euler equations, (c) Navier-Stokes flow solution L_2 error norm for density (ρ), (d) order of accuracy for the Navier-Stokes equations, (e) RANS flow solution L_2 error norm for turbulent kinetic energy (k) and turbulence frequency (ω), (f) order of accuracy for the RANS equations.

5.2.1.3 Three-dimensional flow solver

The three-dimensional flow solver has been verified on uniform hexahedral cells. The levels of grid refinement used in the verification are presented in Table 5.3. We have used a 24 block arrangement (4 blocks in the x-direction, 3 blocks in the y-direction, and 2-blocks in the z-direction) for the simulations presented in this work.

Table 5.3: Levels of grid refinement used for three-dimensional verification.

Grid	Dimensions	Cell width, Δx (m)
1	$8 \times 8 \times 8$	0.125
2	$16 \times 16 \times 16$	0.0625
3	$32 \times 32 \times 32$	0.03125
4	$64 \times 64 \times 64$	0.015625
5	$128 \times 128 \times 128$	7.8125×10^{-3}

The chosen analytic solution for the Euler equations was first presented by Roy et al [155]. It takes the form of Equation 5.4, the constants used are presented in Tables A.7 and the f_s functions are presented in Table A.8. For this test, an inviscid supersonic flow is simulated on the uniform grid. The gas is modelled as a calorically perfect gas with $\gamma = 1.4$, $R = 287.0 \text{ J}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$, $\mu = 10.0 \text{ Pa}\cdot\text{s}$, and $Pr = 1.0$. The results from the three-dimensional Euler tests are presented in Figures 5.3(a), 5.3(b).

The chosen analytic solution for the Navier-Stokes and RANS equations was first presented by Veluri et al [156]. It takes the form of Equation 5.4, the constants used are presented in Tables A.9 and A.11, and the f_s functions are presented in Tables A.10 and A.12, for the Navier-Stokes and RANS equations respectively. For both test cases, a subsonic flow of a viscous gas is simulated on the same uniform grid. The gas is modelled as a calorically perfect gas with $\gamma = 1.4$, $R = 287.0 \text{ J}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$, $\mu = 10.0 \text{ Pa}\cdot\text{s}$, $Pr = 1.0$, and, for the RANS test, $Pr_T = 0.89$. The results from the three-dimensional Navier-Stokes and RANS tests are presented in Figures 5.3(c), 5.3(d), 5.3(e), and 5.3(f).

Similar results to the two-dimensional flow solver verification are observed. As expected, the error is seen to decrease with successive grid refinements for all three-dimensional test cases. Furthermore, all sets of results observe second-order convergence, consistent with the second-order accurate numerical methods implemented.

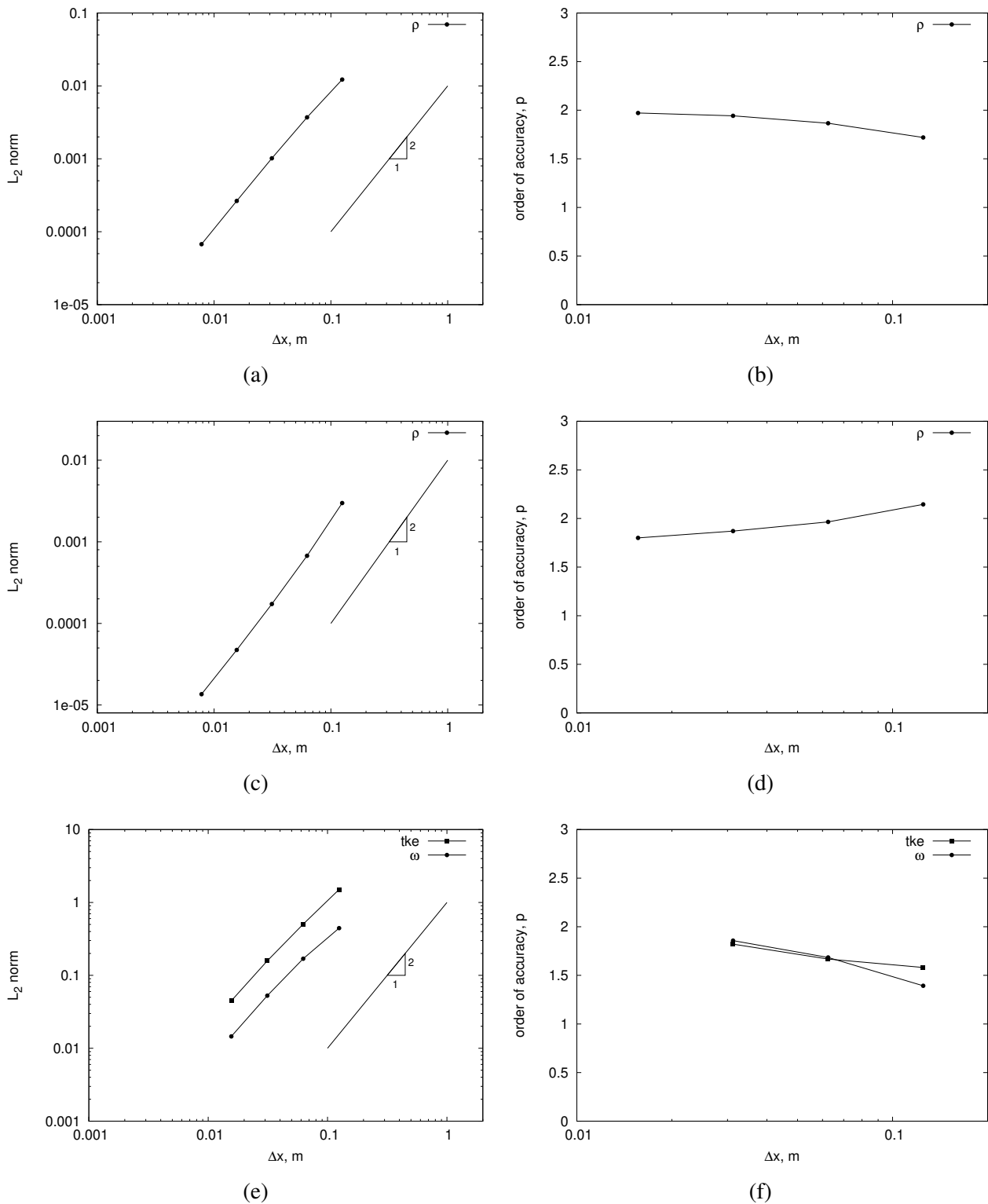


Figure 5.3: Three-dimensional MMS results for Euler, Navier-Stokes, and RANS equations: (a) Euler flow solution L_2 error norm for density (ρ), (b) order of accuracy for the Euler equations, (c) Navier-Stokes flow solution L_2 error norm for density (ρ), (d) order of accuracy for the Navier-Stokes equations, (e) RANS flow solution L_2 error norm for turbulent kinetic energy (k) and turbulence frequency (ω), (f) order of accuracy for the RANS equations.

5.2.2 Shock-capturing capability

The manufactured solutions presented in Section 5.2.1 were smoothly varying, and hence did not verify the flow solver's capability to capture embedded shocks. Whilst it is possible to develop a manufactured solution with an embedded shock [157], it is much more practical to test the shock-capturing capability of the flow solver by comparison to an exact solution with an embedded shock. There exists an analytic solution for a steady-state oblique shock wave resulting from a supersonic inviscid gas impinging on a wedge. The flow solver solution should approach this analytic solution on a family of refined grids.

Numerical setup

A schematic of the test case is shown in Figure 5.4. The problem has been simulated using several grids with increasing refinement, the dimensions are shown in Table 8.2.1. Grid G2 is also shown in Figure 5.4. The wedge angle, θ , is 15° . The solution is independent of the scale of the wedge, for the results presented here, the wedge was 1 m in length. The inflow Mach number is $M_1 = 3$, and the gas flowing over the wedge is air with a specific heat ratio of $\gamma = 1.4$. The solution is independent of the remaining inflow properties, however, for completeness, the values used in this work are provided in Table 8.2. From oblique shock theory, the solution for this particular Mach number and wedge angle is as follows: $\beta = 32.24$, $M_2 = 2.255$, and $p_2/p_1 = 2.822$. Further details on this problem can be found in the publication by Ghia et al. [150].

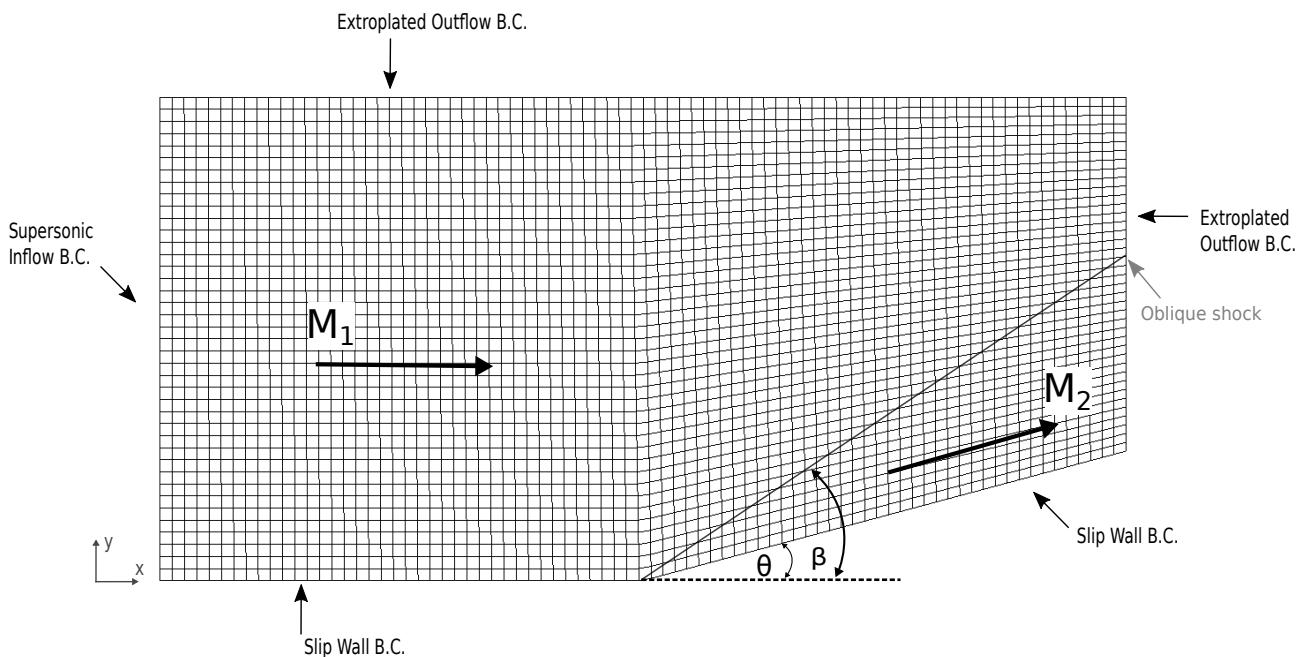


Figure 5.4: Oblique-shock simulation schematic.

Table 5.4: Levels of grid refinement for oblique shock verification.

Grid	Dimensions
G1	80 × 40
G2	160 × 80
G3	640 × 320

Table 5.5: Freestream inflow conditions for oblique shock verification.

Mach number	3.0
Velocity, $\text{m}\cdot\text{s}^{-1}$	1997.2
Pressure, Pa	100.0
Temperature, K	1103.0

Results

The flow solution for this problem on grid G3 is presented as Figure 5.5. It is observed that the oblique shock is correctly turning the flow by 15° , as shown by the superimposed streamline. The entropy change is also observed to abruptly increase across the shock and then remain constant in the post-shock flow, as expected. A small increase in the entropy is noticed near the slip-wall boundary that defines the wedge surface. This increase in entropy is from numerical error and was observed to decrease in magnitude and extent as the mesh was refined, consistent with the results published by Ghia et al. [150]. Table 5.6 presents the numerically computed shock angle, β , approximated as the angle between the x-axis and the point at which the oblique shock intersects the outflow boundary; and the post-shock Mach number M_2 , computed as an average of cell-centered values along a streamline. The analytically derived shock angle and post-shock Mach number, in addition to a measure of the relative error between the numerical and analytical solutions, defined as $\varepsilon = \frac{|f_{exact} - f_{CFD}|}{f_{exact}}$, are also presented in Table 5.6. As expected, for a given grid level, the second-order spatially accurate simulations better approximate the solution than the first-order spatially accurate simulations. Furthermore, the error for each respective spatial order reduces monotonically as the grid is refined, consistent with the expected behaviour. Figure 5.6 presents plots of the pressure distribution along the streamline illustrated in Figure 5.5, for each grid. The numerically computed pressure in the post-shock flow is observed to agree well with the analytic solution. Some slight oscillations are noticed for the spatially second-order accurate simulations, a characteristic of the Venkatakrishnan limiter. These oscillations could be damped by modifying the K parameter, however, at a cost to the spatial order of accuracy. Similar to the previous results tabulated in Table 5.6, as the grid is refined, the simulated shock approaches the analytic solution monotonically.

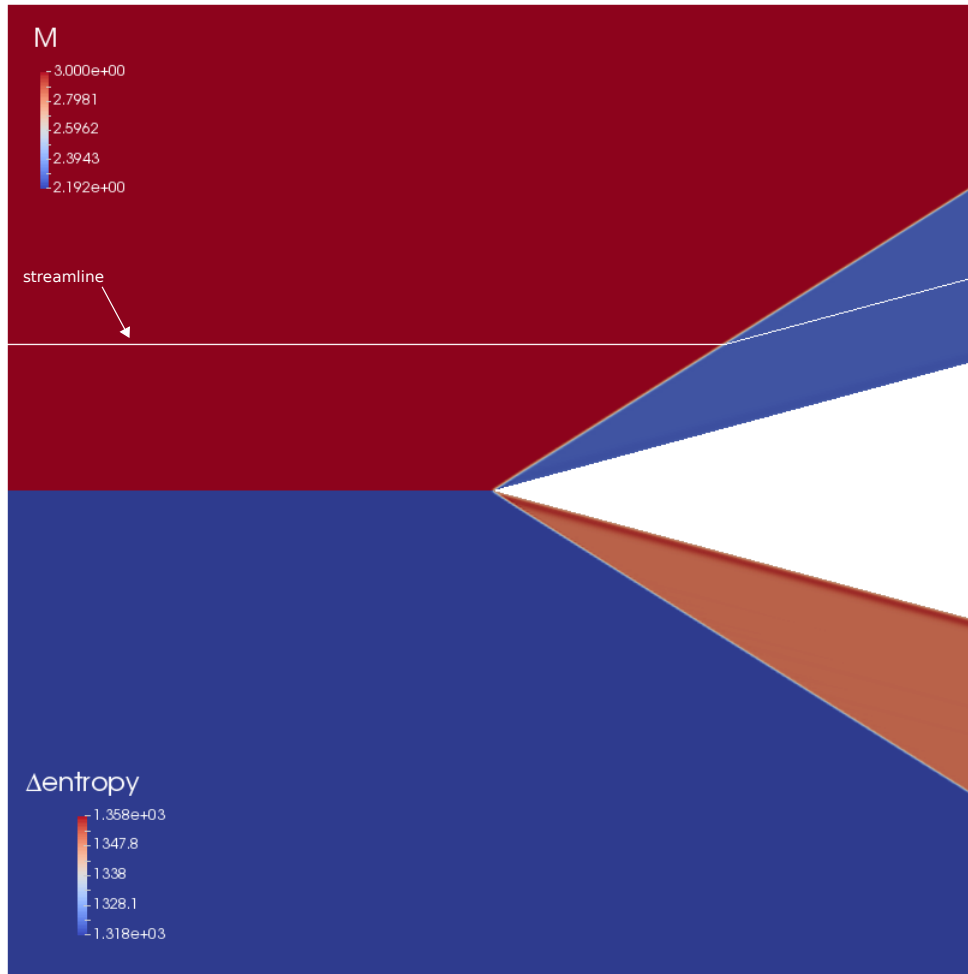
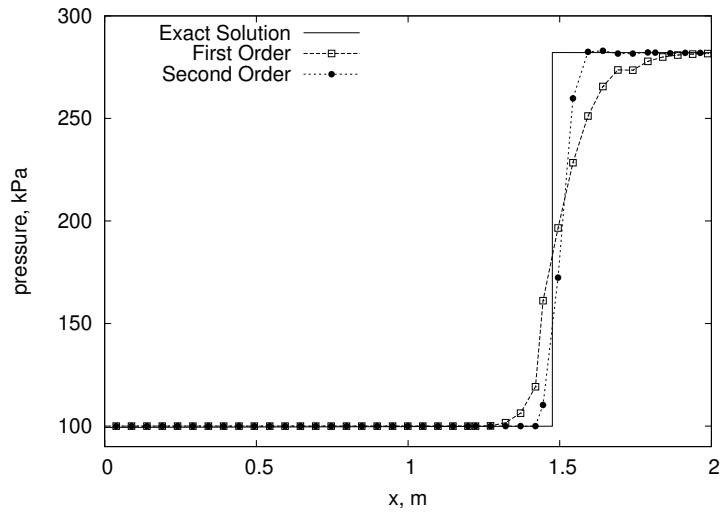


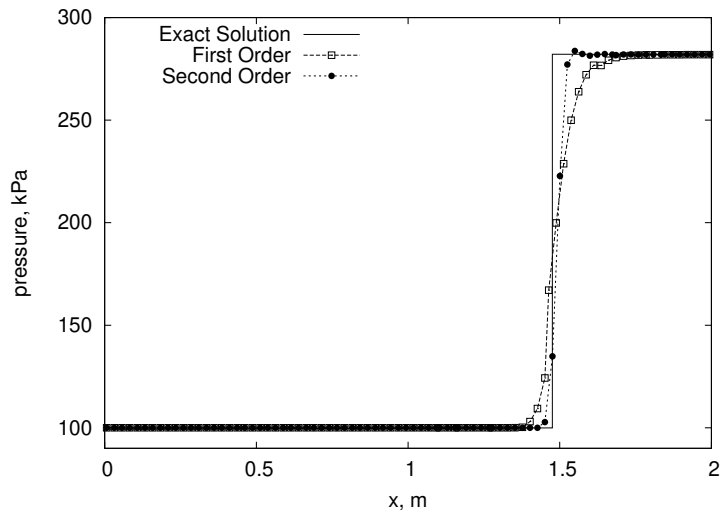
Figure 5.5: Oblique shock numerical solution colour map of Mach number and entropy change ($\text{J}\cdot\text{kg}^{-1}\text{K}^{-1}$) for a second-order computation on grid G3

Table 5.6: Post-shock Mach number and shock angle for oblique shock simulation.

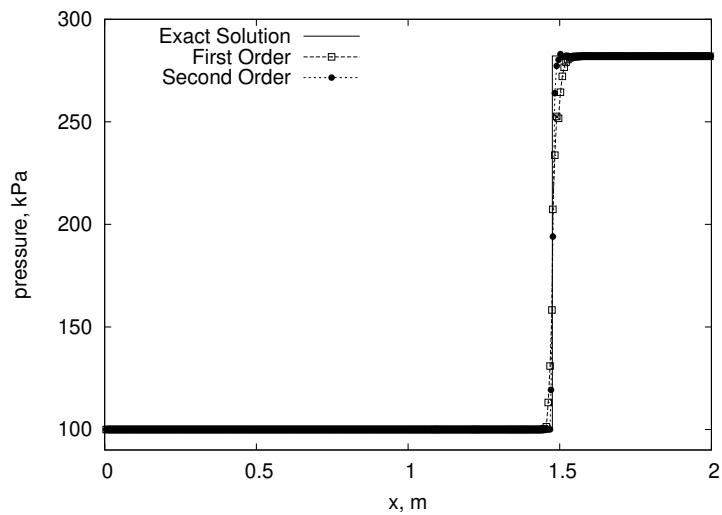
Grid	First Order		Second Order	
	M_2 (% ϵ)	β (% ϵ)	M_2 (% ϵ)	β (% ϵ)
G1	2.233 (0.976)	33.421 (3.66)	2.244 (0.487)	31.928 (0.968)
G2	2.246 (0.399)	32.899 (2.04)	2.252 (0.133)	32.152 (0.273)
G3	2.252 (0.133)	32.412 (0.533)	2.253 (0.089)	32.225 (0.047)



(a)



(b)



(c)

Figure 5.6: Pressure distribution along a streamline on grid (a) G1, (b) G2 and (c) G3.

5.2.3 Self-similar laminar boundary layers

The manufactured solutions presented in Section 5.2.1 did not exercise the flow solver routines for calculating wall bounded flows. Although it is possible to use MMS to verify the implementation of a flow solver's boundary conditions [156, 158, 159], it is more direct to use an analytical solution for comparison, similar to the approach employed in Section 5.2.2 for embedded shocks. Under certain conditions, compressible laminar boundary layers exhibit a behaviour called self-similarity. That is, the flow field profile of the boundary layer is independent of location along the surface. The self-similar solutions approach has been used to verify incompressible laminar boundary layers by Ghia et al. [150]. A similar approach is applied here for compressible laminar boundary layers. The method transforms the compressible laminar boundary layer equations for two-dimensional flow expressed in the standard Cartesian coordinates (x, y) into the similarity variables (η, ζ) . Although the resulting set of equations are still partial differential equations that require solution by some numerical method, the form allows for a much simpler solution. The particular set of partial differential equations solved in the self-similarity variables used in this work can be found on p. 283 in the textbook by Anderson [37].

Numerical setup

A schematic of the computational domain for the laminar flat plate simulation is presented in Figure 5.7, where $L = 1.1$ m, $H = 0.4L$ and $h = 0.75H$. Three levels of mesh refinement were used in this study, the details of which are provided in Table 5.7. The G1 grid is also shown in Figure 5.7. The grid was clustered in both x - and y -directions. On the finest grid, the first cell off the wall had a width of 1.0×10^{-5} m in the y -direction, and the leading edge cell had a width of 4.0×10^{-3} m in the x -direction.

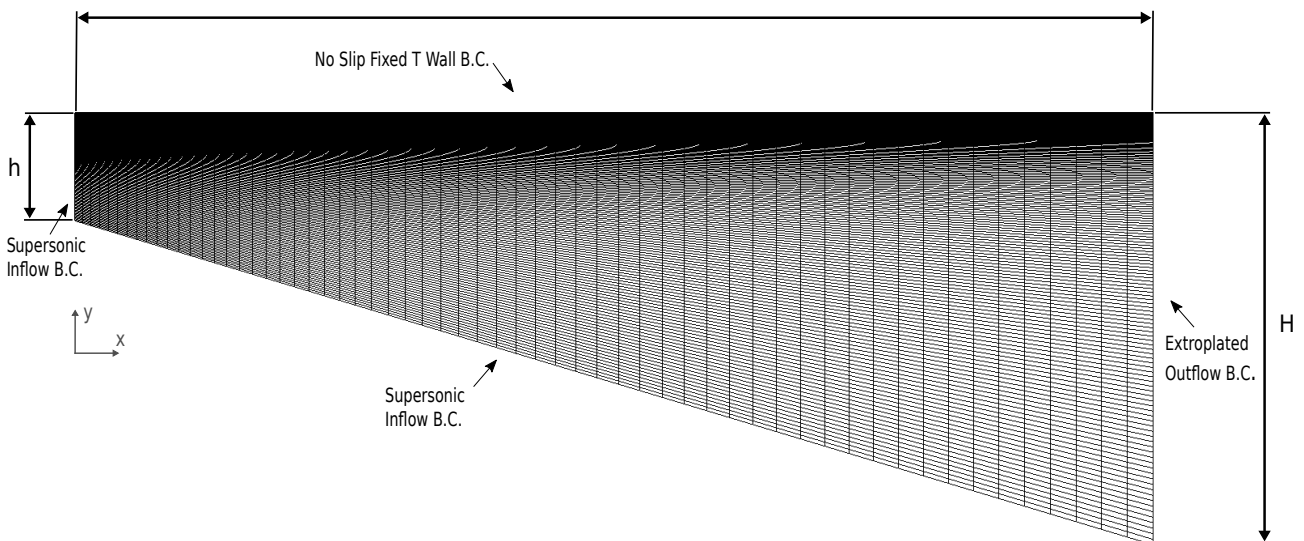


Figure 5.7: Laminar flat plate simulation schematic.

Table 5.7: Levels of grid refinement for laminar flat plate verification.

Grid	Dimensions
G1	61×193
G2	122×386
G3	244×722

Table 5.8: Freestream inflow conditions for laminar flat plate verification.

Mach number	4.0
Velocity, $\text{m}\cdot\text{s}^{-1}$	1390.0
Pressure, Pa	1013.0
Temperature, K	300.0

Results

The skin friction coefficient along the plate computed by the flow solver and the corresponding self-similar solution is presented in Figure 5.8. The flow solver is observed to be in excellent agreement with the self-similar solution downstream of the leading edge. Closer to the leading edge some discrepancy is observed. It is most probable that this is a consequence of the semi-infinite nature of the Navier-Stokes solution compared to the self-similar solution. Ghia et al. [150] states that an exact match between the two methods is only expected asymptotically far downstream of the leading edge. The boundary layer velocity and temperature profiles are compared at a location 1 m downstream of the leading edge in Figure 5.9. Again, excellent agreement is observed between the two solutions.

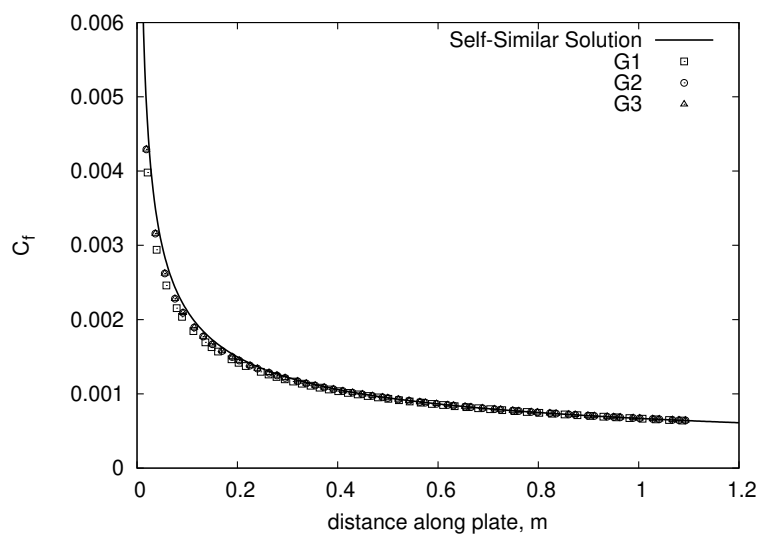
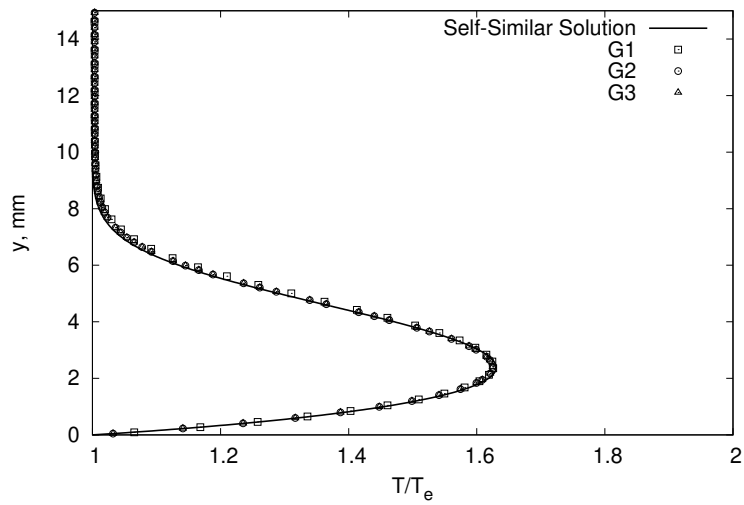
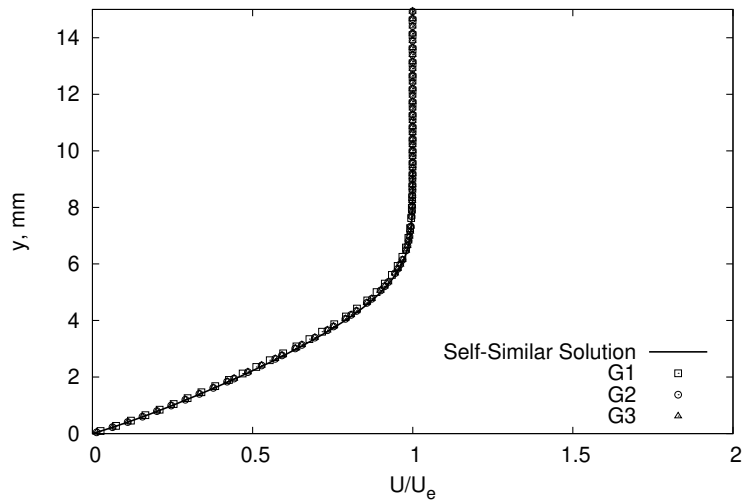


Figure 5.8: Laminar flat plate simulation, skin friction coefficient.



(a) Temperature



(b) Velocity

Figure 5.9: Laminar boundary layer profiles normalised by boundary layer edge flow state at $x = 1$ m.

5.2.4 Summary

The results from a verification study of the flow solver were presented in this section. The Method of Manufactured Solutions provided a formal means for determining the order of accuracy of the core flow solver routines. The results demonstrated that all configurations of the flow solver (Euler/Navier-Stokes/RANS, two- and three-dimensional solvers) achieve the expected second order of spatial accuracy. The flow solver’s shock and boundary layer capturing capabilities were also demonstrated by a qualitative comparison to analytic and numerical solutions, respectively. This concludes the verification of the flow solver.

5.3 Validation

Several validation cases have been taken from the literature to establish the appropriateness of the implemented physical models for design analysis in high-speed flow. Care has been taken to select experimental data that is well established in the literature for CFD code validation. The chosen validation cases are as follows:

- I. Turbulent flow over a flat plate.
- II. Shock wave/turbulent boundary layer interaction.
- III. Hypersonic flow over a sharp nose double cone.
- IV. Hypersonic flow over a 24° compression corner.
- V. The Joint Computational/Experimental Aerodynamic Project (JCEAP) Model.

5.3.1 Turbulent flat plate

The first validation case considers Mach 4.5 air flow over a flat plate. The experimental data for this validation case was originally reported by Mabey et al. [160], and was included in a critical compilation of compressible turbulent boundary data by AGARD [161]. This data has been used to validate the $k - \omega$ turbulence model in a previous version of Eilmer [88], and has been more recently used to validate the $k - \omega$ turbulence model in Eilmer's structured-grid explicit solver [162]. The flat plate was tested in the supersonic wind tunnel at R.A.E. Bedford, and spanned 0.9 m in width and was approximately 1.6 m in length. During the experimental campaign, the plate was thermally insulated. Boundary layer transition was forced by a roughness band consisting of several small glass spheres near the leading edge of the model. Skin friction was measured using a floating-element balance, and velocity and temperature profiles across the boundary layer were measured using a combined total temperature and Pitot pressure probe.

Numerical setup

A schematic of the computational domain for the turbulent flat plate simulation is presented in Figure 5.10, where $L = 1.4$ m, $H = 0.4L$ and $h = \frac{3}{4}H$. Four levels of grid refinement were used to demonstrate grid convergence. The grid dimensions are provided in Table 5.9. Also shown in Figure 5.10 is the G1 grid for this computational case. The grid was clustered in both x and y directions. On the finest grid, grid G4, the first cell off the wall had a width of 1.56×10^{-5} m in the y -direction, and the leading edge cell had a width of 5×10^{-4} m in the x -direction. This grid achieves a y^+ of less than 1, except at the leading edge. The inflow conditions for the simulation were taken from Chan [89] and are provided in Table 5.10.

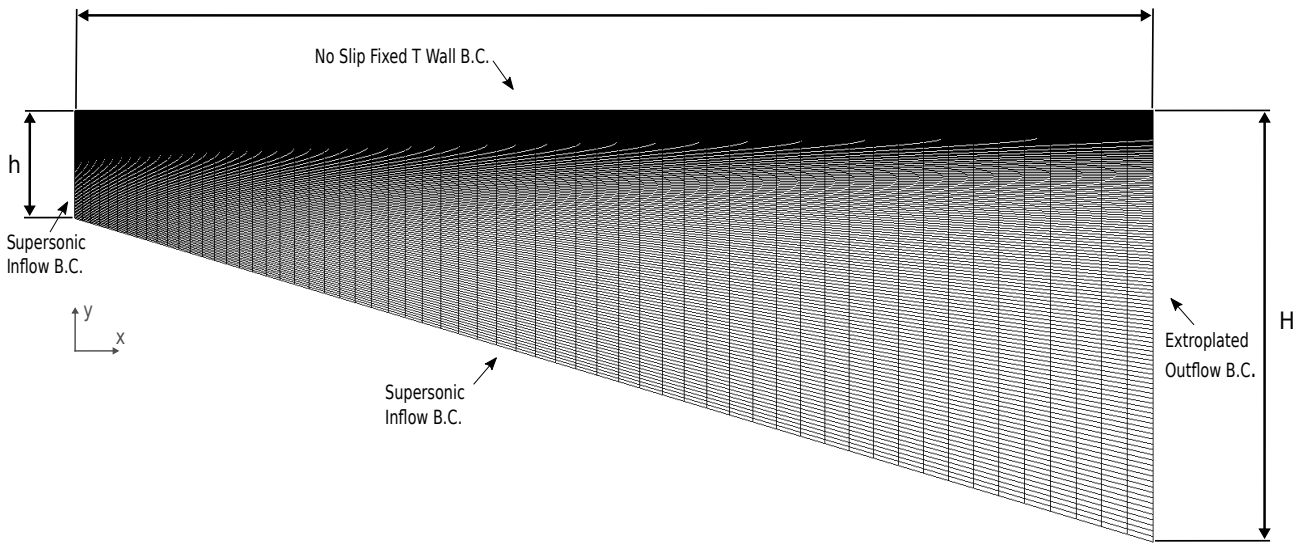


Figure 5.10: Turbulent flat plate simulation schematic.

Table 5.9: Levels of grid refinement for turbulent flat plate validation case.

Grid	Dimensions
G1	129 × 97
G2	258 × 194
G3	516 × 388
G4	1032 × 776

Table 5.10: Freestream inflow conditions for turbulent flat plate validation case.

Mach number	4.5
Velocity, $\text{m}\cdot\text{s}^{-1}$	712.9
Pressure, Pa	3160.0
Temperature, K	62.16
Turbulence Intensity, %	1.0
Eddy viscosity ratio, $\frac{\mu_{\text{lam}}}{\mu_{\text{turb}}}$	1.0

Results

Figure 5.11 shows the calculated skin friction coefficient along the flat plate. Good grid convergence is observed downstream of the leading edge for grids G2 and G3. Also shown in Figure 5.11 is the experimentally measured skin friction coefficient. Good agreement is observed between the calculated and experimental values. The boundary layer profiles for the velocity, Mach number, and temperature at a location 0.368 m downstream of the leading edge are shown in Figure 5.12. Again, good grid convergence is demonstrated on grids G2 and G3. Despite the degree of accuracy of the calculated skin friction coefficient, the boundary layer profiles are observed to deviate from the experimental data by some small amount. A similar result has been reported for Eilmer's structured grid explicit solver [162]. One proposed reason for this discrepancy is that the $k - \omega$ turbulence model is developing the boundary layer slower than in reality. Another possible reason is that the forced transition of the experiment has in some way altered the boundary layer development. Figure 5.13 shows a comparison of the boundary profiles from the numerical simulation at a distance 0.45 m downstream of the leading edge compared to the original location of $x = 0.368$ m. It is noted that better agreement is observed at $x = 0.45$ m. This result warrants a more in-depth analysis, not presented here. For the current work, accurate calculation of the skin friction along a surface is sufficient, and consequently, the results in Figure 5.11 demonstrate the flow solver's applicability in this context.

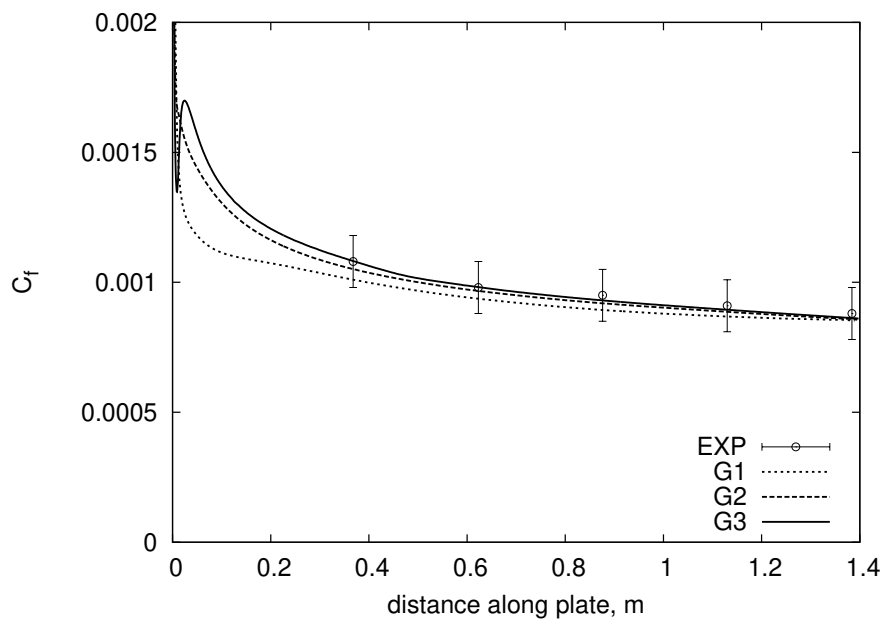
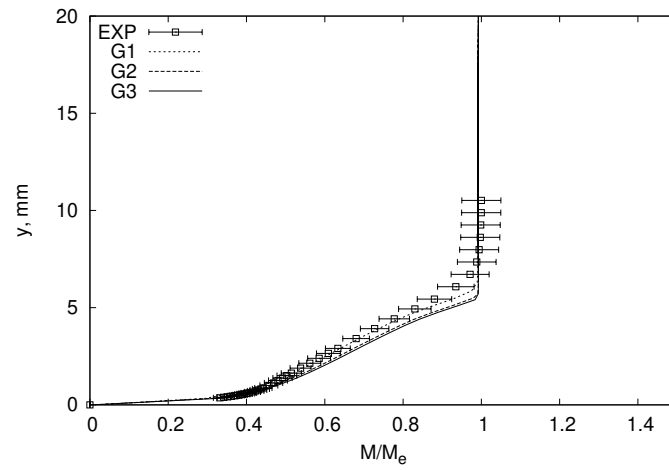
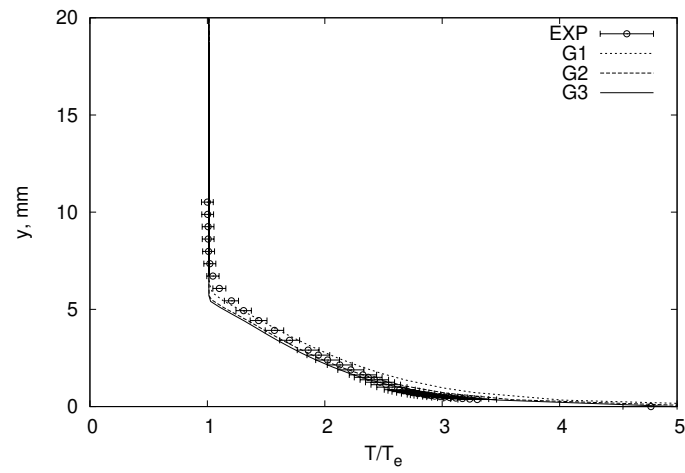


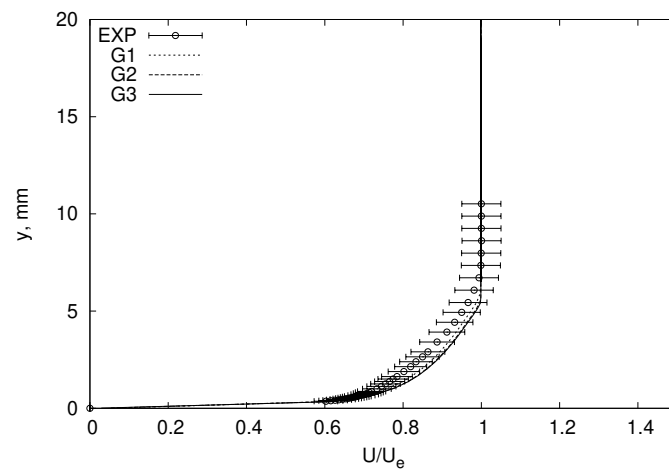
Figure 5.11: Turbulent flat plate simulation, skin friction coefficient. Experimental data from Mabey et al. [160].



(a) Mach number

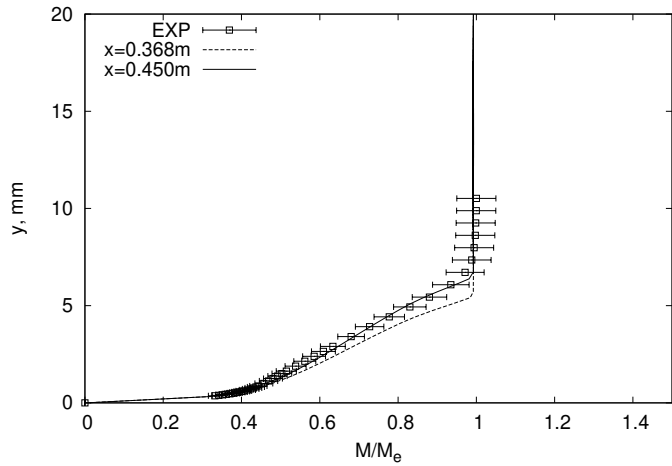


(b) Temperature

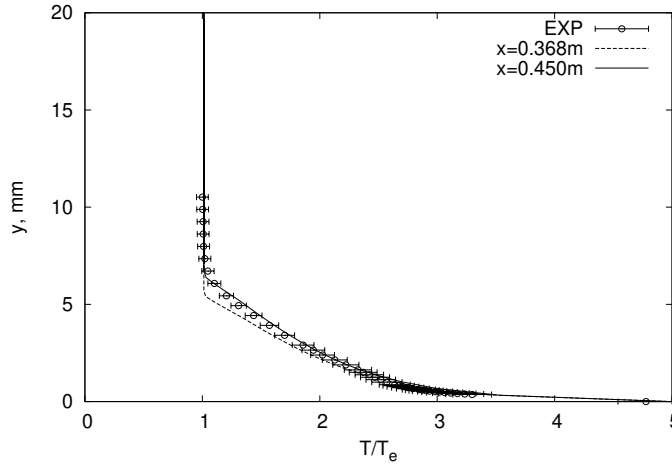


(c) Velocity

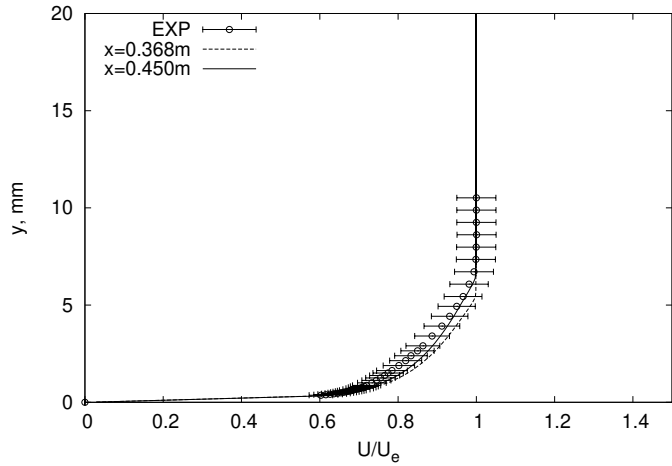
Figure 5.12: Turbulent boundary layer profiles at $x = 0.368$ m normalised by boundary layer edge flow state. Experimental data from Mabey et al. [160].



(a)



(b)



(c)

Figure 5.13: Comparison of boundary layer profiles at $x = 0.368\text{ m}$ and $x = 0.45\text{ m}$ normalised by boundary layer edge flow state. Experimental data from Mabey et al. [160].

5.3.2 Turbulent flat plate with impinging shock

The second validation case is of a shock impinging on a flat plate with a fully developed turbulent boundary layer. The particular configuration considered here is the Mach 5/10° shock generator case taken from the campaign of experiments published by Schülein et al. [163, 164]. The original model was tested in the DLR Ludwig Tube Wind Tunnel Facility (RWG, Tunnel B) by Schülein et al. [163], who later published further results for the same configuration [164]. The available experimental data includes static wall pressure, skin friction coefficient and the Stanton number along the flat plate. This data has been previously used to validate the $k - \omega$ turbulence model for several codes [165–169].

Numerical setup

A schematic of the computational domain for the simulation is presented in Figure 5.14, where $\theta = 10^\circ$. Three levels of grid refinement were used to demonstrate grid convergence. The grid dimensions are given in Table 5.11. Also shown in Figure 5.14 is the G1 grid for this computational case. The grid was clustered in the y -direction near the wall, and also in the x -direction near the interaction zone. On the finest grid, grid G4, the first cell off the wall had a width of 4.0×10^{-7} m in the y -direction. This grid achieves a y^+ of less than 1 along the length of the flat plate. The inflow conditions for the simulation were computed from the total pressure and total temperature values reported by Schülein et al. [164] assuming ideal air with $\gamma = 1.4$, and are provided in Table 5.12.

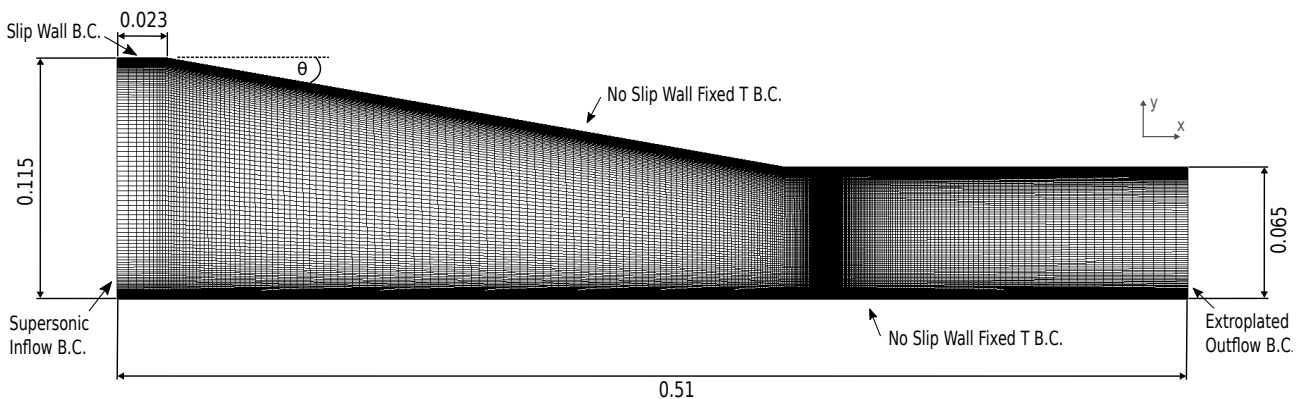


Figure 5.14: Shock wave turbulent boundary layer interaction simulation schematic (units in meters).

Table 5.11: Levels of grid refinement for shock wave turbulent boundary layer interaction validation case.

Grid	Dimensions
G1	155×100
G2	310×200
G3	620×400
G4	877×566

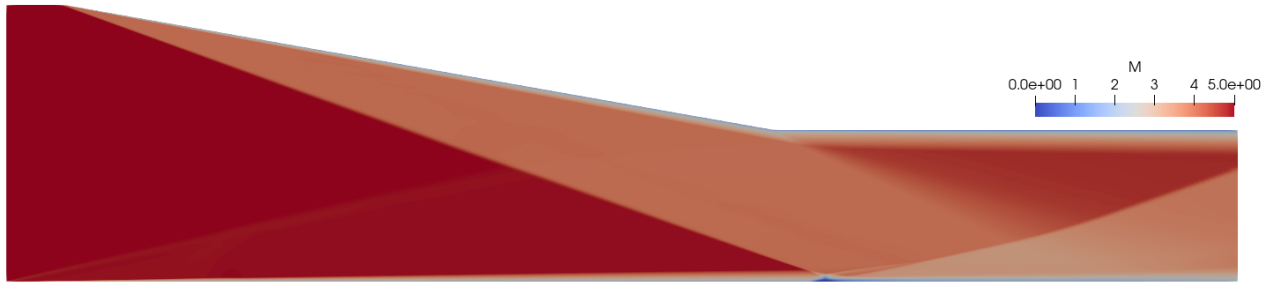
Table 5.12: Freestream inflow conditions for shock wave turbulent boundary layer interaction validation case.

Mach number	5.0
Velocity, $\text{m}\cdot\text{s}^{-1}$	828.5
Pressure, Pa	4070.0
Temperature, K	68.33
Turbulence Intensity, %	1.0
Eddy viscosity ratio, $\frac{\mu_{\text{lam}}}{\mu_{\text{turb}}}$	1.0

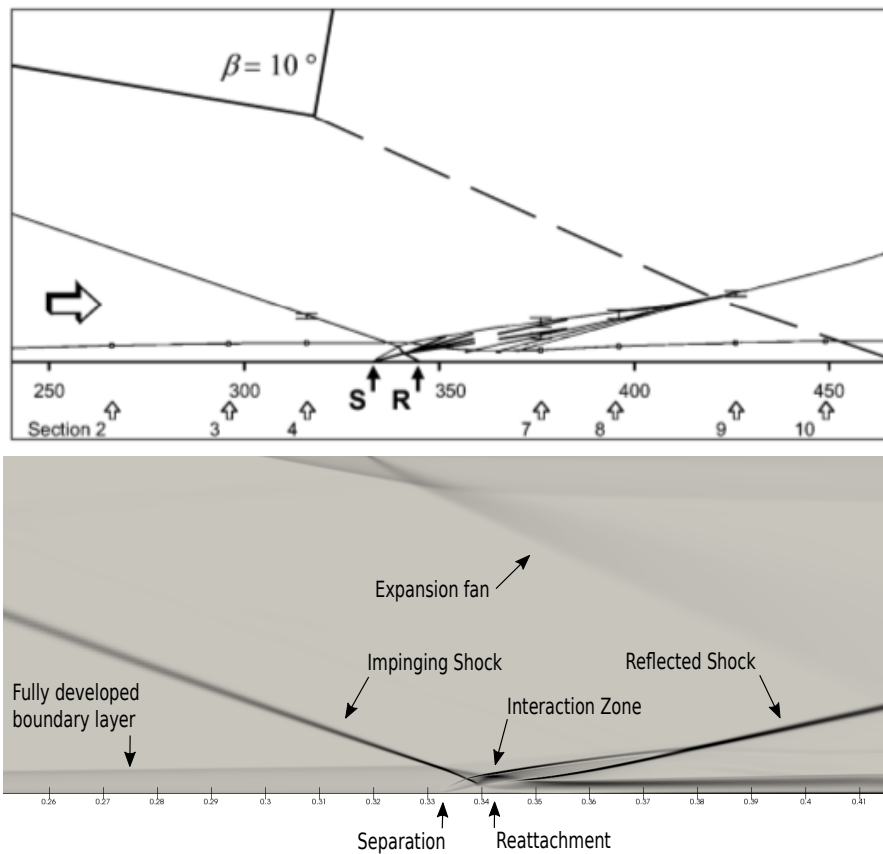
Results

A colour map of Mach number is shown in Figure 5.15(a), along with a numerical schlieren insert of the interaction zone. From a comparison of this computed flow field with the experimentally captured shadow-graph, Figure 5.15(b), it is observed that the shock impinges on the boundary layer at approximately the same location along the flat plate. Furthermore, the separation and reattachment points agree well between the calculation and the experiment. Figure 5.16 shows the calculated static wall pressure, skin friction coefficient, and Stanton number along the flat plate. It is observed that good grid convergence is achieved for grid G3 and grid G4, determined from the wall pressure and skin friction coefficient plots. However, it is noted that a grid-independent solution of the Stanton number has not been achieved in the interaction zone. Compared to the experimental results, an excellent agreement is observed for the wall pressure along the plate. The skin friction coefficient agrees well immediately upstream of the separation zone, however, there is some discrepancy downstream of the interaction. Despite this, the separation extent is in good agreement with the experimental data. Concerning the skin friction coefficient, Droske [169] has observed sensitivity in the flow downstream of the interaction when employing the SST turbulence model. It was observed that an increase in the turbulence intensity (from 1% to 5%) produced a better agreement between the experimentally observed skin friction coefficient downstream of the interaction and that calculated by the SST turbulence model. However, this also reduced the extent of separation. This suggests that a poor estimation of the freestream turbulence intensity alone is not sufficient to explain the discrepancy. It is interesting to note that some inconsistency is observed when comparing the skin friction distribution of several authors who have computed this experiment using the $k - \omega$ model [165–169]. Finally, it is observed

that the Stanton number agrees well upstream of the interaction and downstream of the interaction zone. A large deviation in agreement is observed within the interaction zone, where a severe spike in the Stanton number suggests an over-prediction of the heat-transfer in the interaction zone. This has been observed by others who have employed two-equation turbulence models [165, 166, 169]. Droske [169] cites Dolling [170] when attributing this behaviour as a common issue in simulating these flows, suggesting that it is most likely a deficiency of the two-equation turbulence models.



(a) Colour map of Mach number.



(b) Quantitative flow diagram from Schülein et al. [164] (top) and numerical schlieren (bottom).

Figure 5.15: Shock wave turbulent boundary layer interaction flow solution.

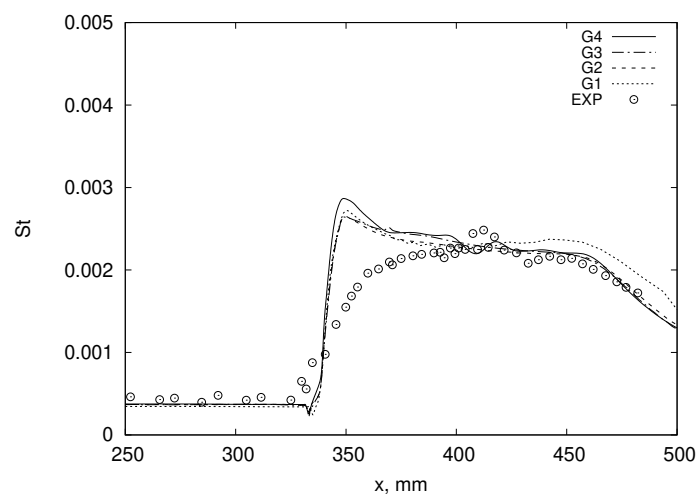
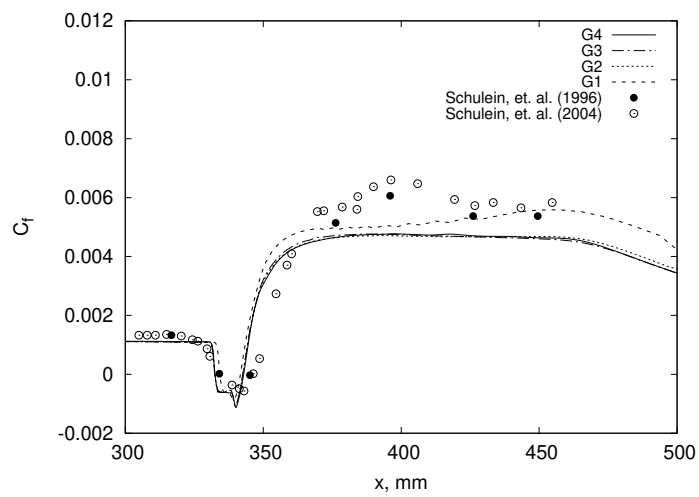
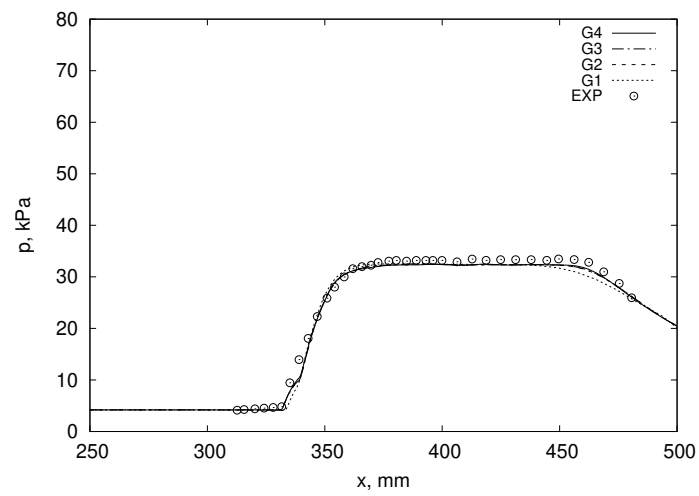


Figure 5.16: Calculated distributions for shock wave turbulent boundary layer interaction: (a) wall pressure, (b) skin friction and (c) Stanton number. Experimental data from Schülein et al. [163, 164].

5.3.3 Sharp nose double cone

The third validation case is Mach 12.5 nitrogen (N_2) flowing over a sharp nose double cone configuration. This validation case has been catalogued in the CUBRC (Calspan-University at Buffalo Research Center) database of aerothermal measurements in “building block“ experiments for CFD validation [171]. Several runs of the model are presented in the database. For this validation exercise, run 35 has been selected. The model was originally tested in the Calspan University shock tunnels. For the experiments, the gas flowing over the model was chosen to be pure nitrogen (N_2) to avoid high-temperature effects, since at the selected flow conditions significant dissociation of oxygen molecules would occur in air [172]. The dissociation of N_2 molecules for this flow condition has been shown to be insignificant [173]. It is reported in the literature that this particular case can be sensitive to thermal non-equilibrium effects via excitation of vibrational energy modes. In addition, a breakdown of the no-slip wall boundary condition assumption has been studied [174]. Despite these effects, good agreement to the experimental data has been demonstrated using a perfect gas assumption with standard no-slip wall boundary conditions for run 35 [172]. For further details on the experimental model and test facilities at Calspan University, refer to Holden et al. [175, 176]. This validation case has been used to validate a previous version of Eilmer [177].

Numerical setup

A schematic of the computational domain for the sharp nose double cone simulation is presented in Figure 5.17, where $\theta = 25^\circ$ and $\beta = 55^\circ$. Four levels of grid refinement were used to demonstrate grid convergence. The grid dimensions are given in Table 5.13. Also shown in Figure 5.17 is the G1 grid for this computational case. The grid was clustered in the y-direction near the wall, and also in the x-direction near the separation region and abrupt expansion region. On the finest grid, grid G4, the first cell off the wall had a width of 2.25×10^{-6} m in the y-direction. The inflow conditions for the simulation are given in Table 5.14.

Table 5.13: Levels of grid refinement for sharp nose double cone validation case.

Grid	Dimensions
G1	175×80
G2	350×160
G3	700×320
G4	990×453

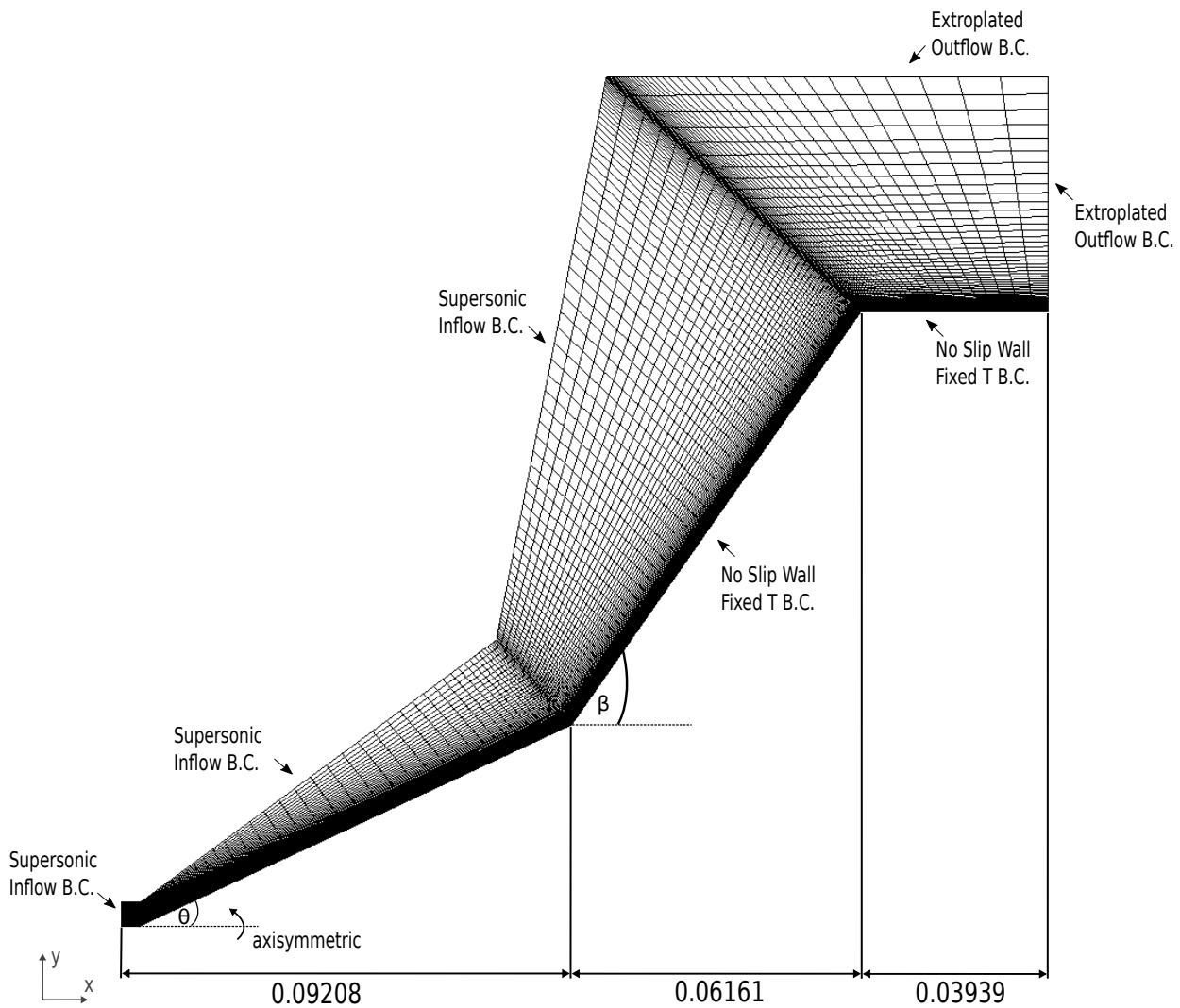


Figure 5.17: Sharp nose double cone simulation schematic (units in meters).

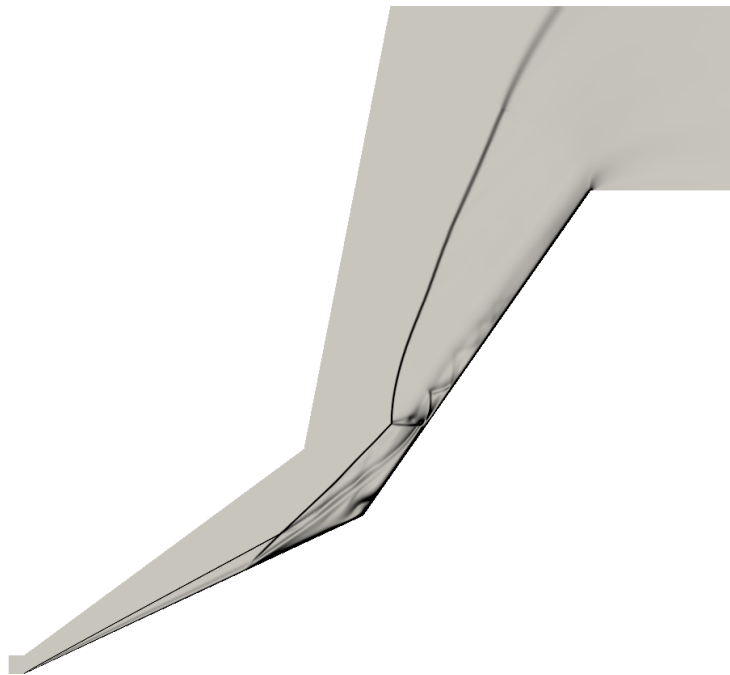
Table 5.14: Freestream inflow conditions for sharp nose double cone validation case.

Mach number	12.5
Velocity, $\text{m}\cdot\text{s}^{-1}$	2576.0
Pressure, Pa	18.55
Temperature, K	102.2

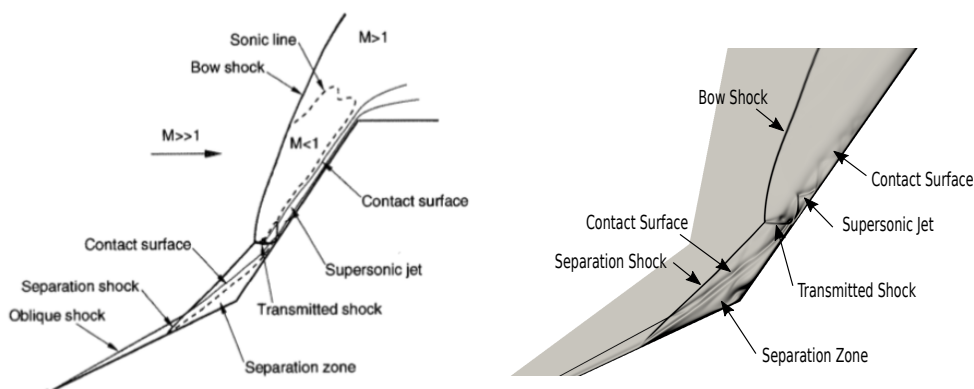
Results

Shown in Figure 5.18(a) is the computed numerical schlieren of the sharp nose double cone flowfield. The numerical schlieren displays all major flow features as identified by Nompelis et al. [174], as shown in Figure 5.18(b). The experimental schlieren image for this validation case is of a poor quality, owing to the low density of the flow, thus no meaningful comparison can be made. The calculated static wall pressure and heat transfer are shown in Figure 5.19. It is observed that good grid convergence is attained on grids G2 and G3. Also shown in Figure 5.19 are the experimentally measured

data points [175]. It is observed that the location and extent of separation has been calculated to a fair degree of accuracy, despite the perfect gas assumption. The reattachment location, where peak heating and pressure occurs, is also captured quite well by the flow solver. In the separated flow region, an under prediction of the heat transfer and a slight over-prediction of the static wall pressure is observed, as expected by the modelling assumptions [174]. The heat transfer downstream of the reattachment is captured well, however, there is a stronger oscillation in the static pressure, than observed in the experimental data. A similar result for the static wall pressure was also noticed by Maclean et al. [172].

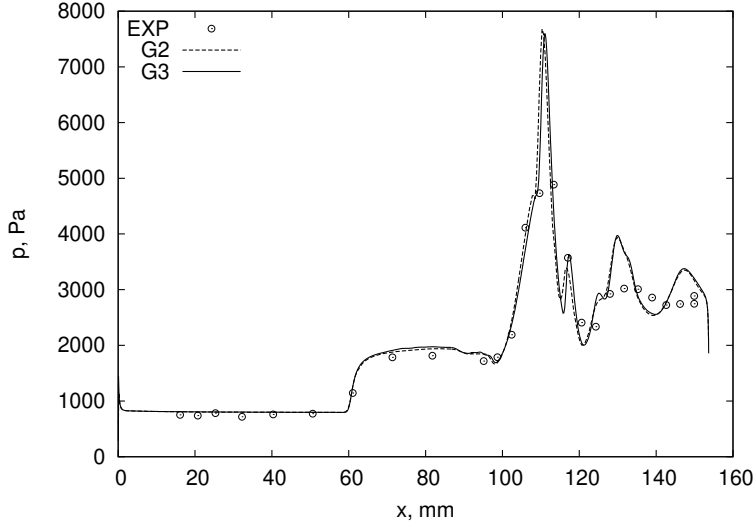


(a) Numerical schlieren.

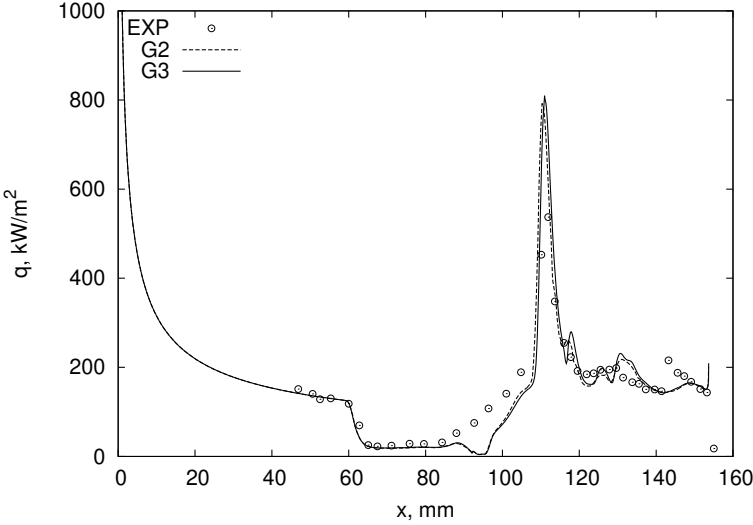


(b) Flow diagram from Nompelis et al. [174] (left), annotated numerical schlieren (right).

Figure 5.18: Sharp nose double cone simulation flow features.



(a)



(b)

Figure 5.19: Calculated distributions for double cone: (a) wall pressure, (b) wall heat transfer.

5.3.4 24° compression-corner

The fourth validation case is a flat plate/wedge configuration, herein referred to as a compression-corner. Holden et al. [178] tested several compression-corners with various wedge angles in the Calspan 48-in. shock tunnel. The 24° angle model has been selected for the current validation work. The experimental data for this particular configuration has been shown to include significant three-dimensional effects [179]. The combination of the steep inclination of the wedge and the insufficient spanwise width of the model resulted in spillage of gas over the model edge during the experimental campaign. Consequently, the pressure relief caused by the spillage significantly reduced the extent of the separated region, compared to what would be expected for true two-dimensional flow. This provides a suitable test to exercise the three-dimensional flow solver. Several authors have used this experimental data set to validate their flow solvers. The first published results were by Hung et al. [180]. For further details on the test facility refer to Holden et al. [178].

Numerical setup

A schematic of the computational domain for the 24° compression-corner simulation is presented in Figure 5.20, where $\theta = 24^\circ$. The width of the model was 2 ft (approximately 0.609 m). The model had a total length of 2.88 ft (approximately 0.8778 m), the flat plate length was 1.44 ft (approximately 0.4389 m). Only half the model was simulated, using a slip wall boundary condition on the symmetry plane. Four levels of grid refinement were used to demonstrate grid convergence. The grid dimensions are given in Table 5.15. Also shown in Figure 5.20 is the G1 grid for this computational case. The grid was clustered in the y-direction near the wall, and also in the z-direction near the model edge. On the finest grid, grid G4, the first cell off the wall had a width of 1.0×10^{-5} m in the y-direction. The inflow conditions for the simulation were computed from the total pressure and total temperature values reported by Holden et al. [178] assuming ideal air with $\gamma = 1.4$, and are provided in Table 5.16. The flow for the experimental campaign had a Reynolds number sufficiently low to ensure a fully laminar flow.

Table 5.15: Levels of grid refinement for 24° compression-corner validation case.

Grid	Dimensions
G1	$101 \times 51 \times 10$
G2	$202 \times 102 \times 20$
G3	$285 \times 144 \times 28$
G4	$404 \times 204 \times 40$

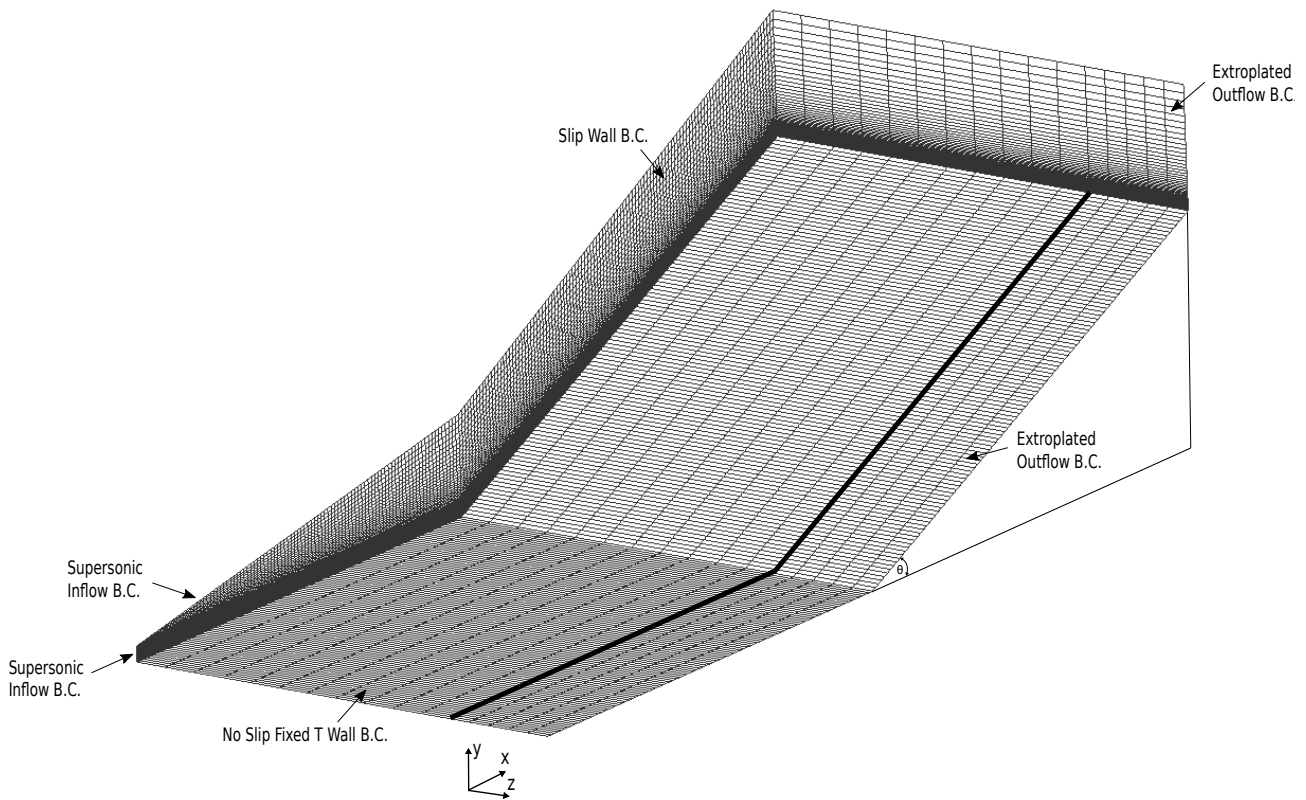


Figure 5.20: 24° compression-corner simulation schematic

Table 5.16: Freestream inflow conditions for 24° compression-corner validation case.

Mach number	14.1
Velocity, $\text{m}\cdot\text{s}^{-1}$	2408.93
Pressure, Pa	10.10
Temperature, K	72.64

Results

A colour map of the pressure on the model surface is shown in Figure 5.21. The three-dimensionality of the flow solution is evidenced by the curvature of the separation bubble. To demonstrate the significance of the three-dimensional flow features, results on a sequence of two-dimensional grids were also computed. A comparison of the pressure coefficient (C_p) for the two- and three-dimensional solutions is presented in Figure 5.22(a). This plot demonstrates the significance of the edge effects outlined in the previous section. The calculated skin friction coefficient and heat transfer coefficient for the three-dimensional simulation are shown in Figures 5.22(b) and 5.22(c). A good agreement between the calculations and the experimental data published by Holden et al. [178] is observed. Similar results have also been reported in the literature [179, 180].

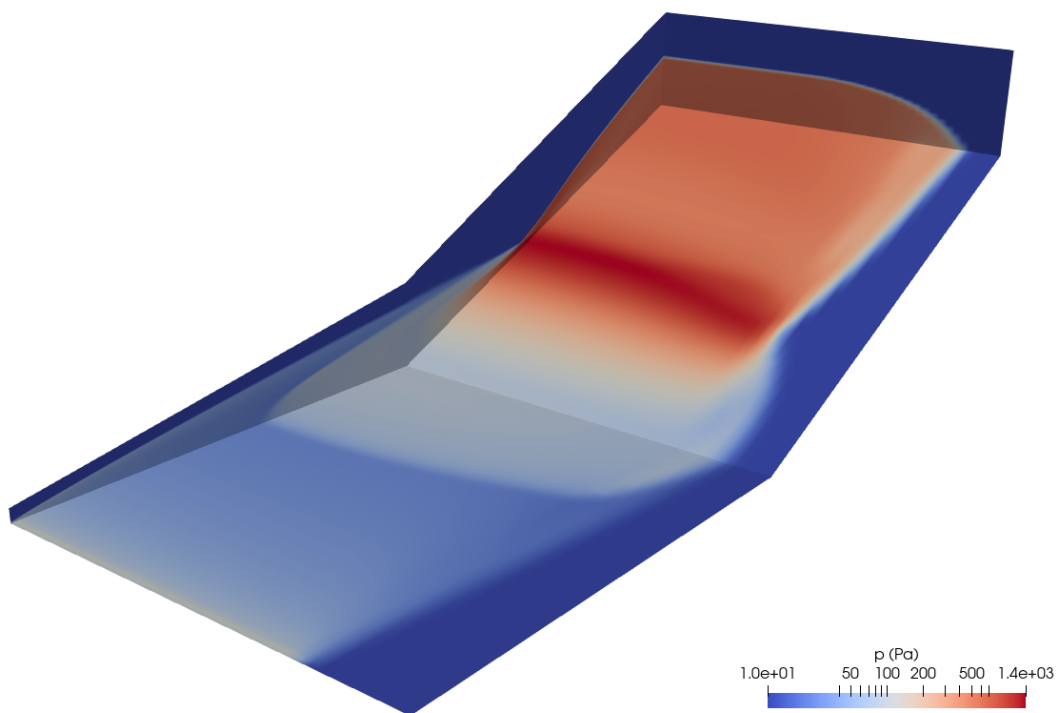
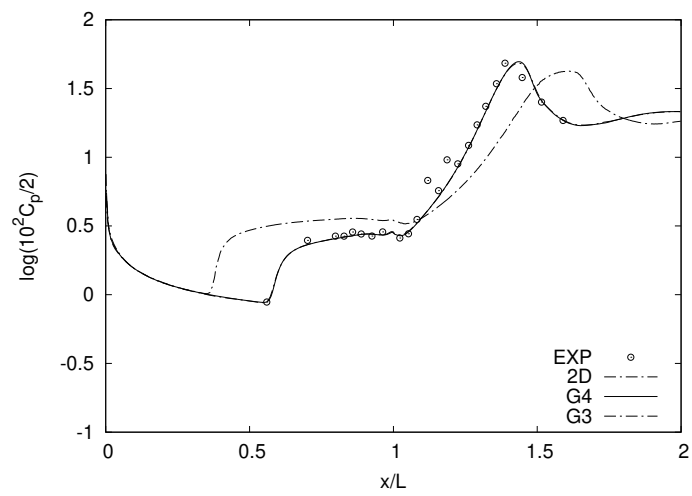
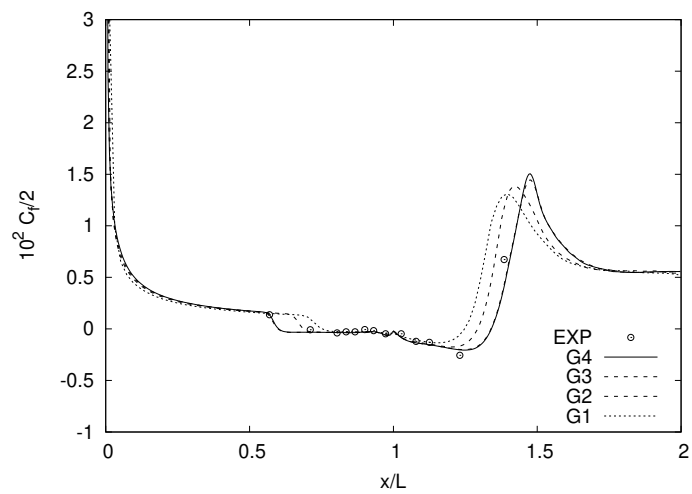


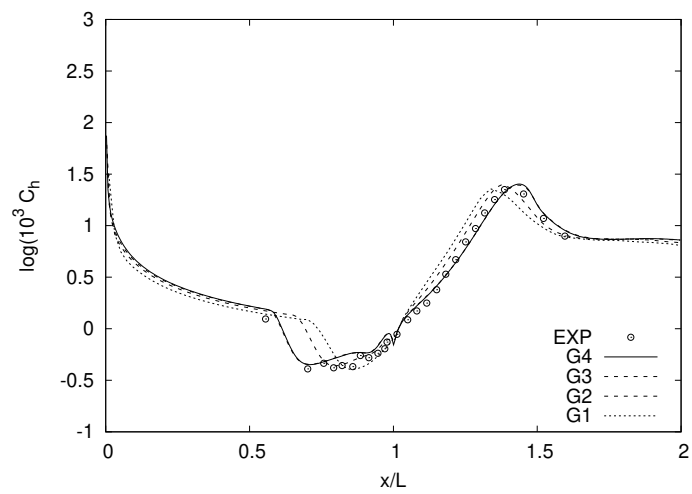
Figure 5.21: Colour map of pressure field for 24° compression corner.



(a)



(b)



(c)

Figure 5.22: .

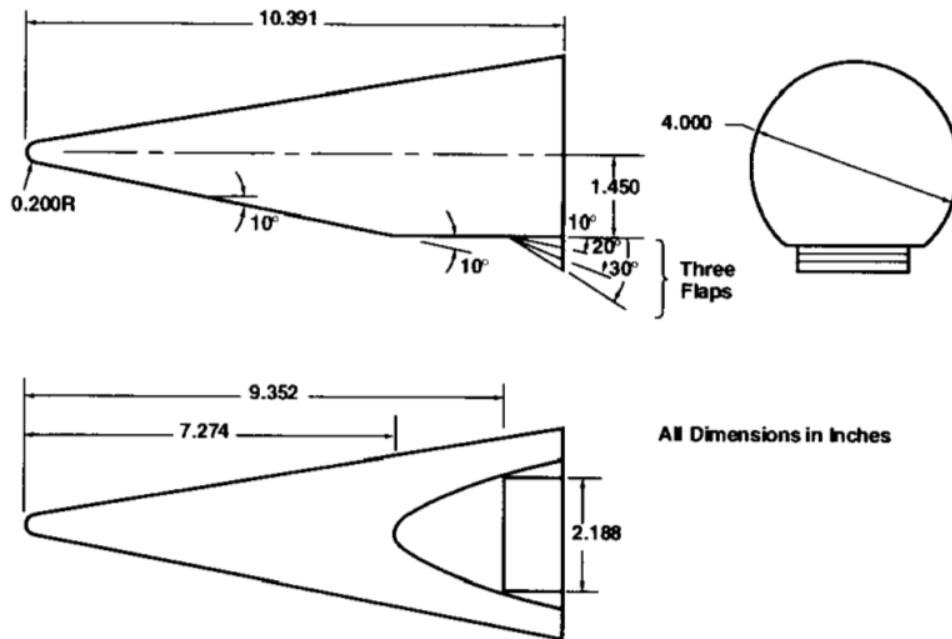
Calculated distributions for compression-corner: (a) pressure coefficient, (b) skin friction coefficient, and (c) heat transfer coefficient. Experimental data from Holden et al. [178]

5.3.5 JCEAP

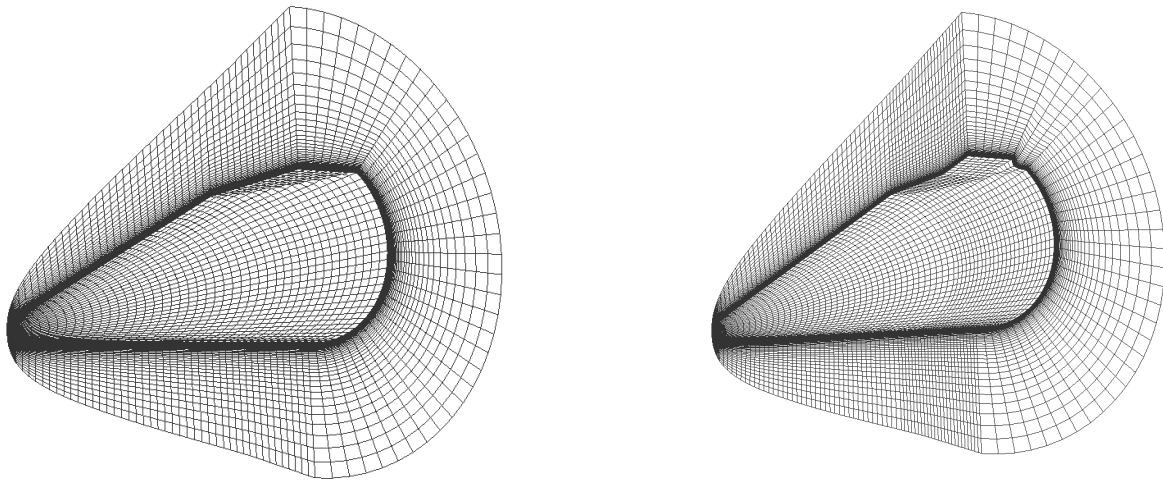
The final validation case is the Joint Computational/Experimental Aerodynamics Program model. The model was designed as part of a research program dedicated to producing an experimental methodology for CFD code validation [181]. The model itself is a spherically blunted 10° half-angle cone with a slice parallel to the vehicle axis. On the slice section, a flap can be attached. In the experiment, several flap angles were used. For the validation of Eilmer, the slice-only and 10° flap configurations were considered. The experiments were conducted in the Sandia Laboratories blow-down-to-vacuum hypersonic wind tunnel using a Mach 7.84 flow condition. The published experimental data available includes aerodynamic force and moment measurements for several orientations, in addition to static wall pressure measurements and surface flow visualisation results. The results of the experimental campaign were first published by Oberkampf et al. [182] and the corresponding computational results were published by Walker et al. [183]. Since the initial publications, further experimental results of the campaign have been published by Oberkampf et al. [184, 185]. The experimental data has also since been used for CFD validation [186–188]. For further details on the project and the Sandia Laboratory facility, refer to Oberkampf et al. [149].

Numerical setup

The dimensions of the JCEAP model are presented in Figure 5.23(a). The simulation used supersonic inflow and extrapolate outflow boundary conditions. Half the model was simulated, with a slip wall boundary condition applied along the symmetry plane. The model surface had a no-slip boundary condition, with a fixed temperature of 316.7 K as suggested by Walker et al. [183]. The inclusion of the base region in the simulation has been shown to have negligible effect [188]. Consequently, the wake region was not resolved for this work. Grid refinement studies were not performed for this validation case due to the large computing costs. Guidance in building the grids was taken from the lessons reported by Walker et al. [183]. In addition, axisymmetric simulations (presented in the following section) were employed to ensure the nose region was sufficiently refined. The resulting grid sizes developed for this work are presented in Table 5.17. Examples of the generated grids are shown in Figure 5.23(b). The grid was clustered towards the model surface. The first cell off the wall had a width of 5.0×10^{-6} m in the y-direction. The inflow conditions for the simulations are given in Table 5.18. It should be noted that, as first reported by Roy et al. [188], several of the original papers (including references [184, 187]) published an erroneous freestream static pressure. This work uses the corrected inflow conditions, reported in Roy et al. [188]. The flow for the experimental campaign had a low Reynolds number, ensuring a fully laminar flow. For the axial force coefficient calculations, the base drag was computed by assuming $p_{\text{base}} = p_\infty$ and a reference area of 11.525 in² [183].



(a) Model dimensions (inches) reproduced from Oberkampff et al. [149].



(b) slice-only configuration grid (left), 10° flap configuration grid (right).

Figure 5.23: JCEAP simulation details.

Table 5.17: Levels of grid refinement for JCEAP validation case.

model	no. cells
10° flap configuration	2095114
slice-only configuration	183726

Table 5.18: Freestream inflow conditions for JCEAP validation case.

Mach number	7.841
Velocity, $\text{m}\cdot\text{s}^{-1}$	1103.9
Pressure, Pa	290.9
Temperature, K	47.7

Results

Figure 5.24 shows the surface pressure results for the axisymmetric simulations used in the grid design. Similar agreement to the experimental data is observed as compared to the results published by Roy et al. [188]. Three-dimensional solutions were calculated for the slice-only configuration at 0° and 10° angle of attack, and for the 10° flap configuration at -5°, 0°, and 10° angle of attack. The calculated flowfield for the 10° flap configuration at -5° angle of attack is shown in Figure 5.25. The calculated axial force coefficient for all simulated configurations is shown in Figure 5.26. Good agreement is observed for all configurations.

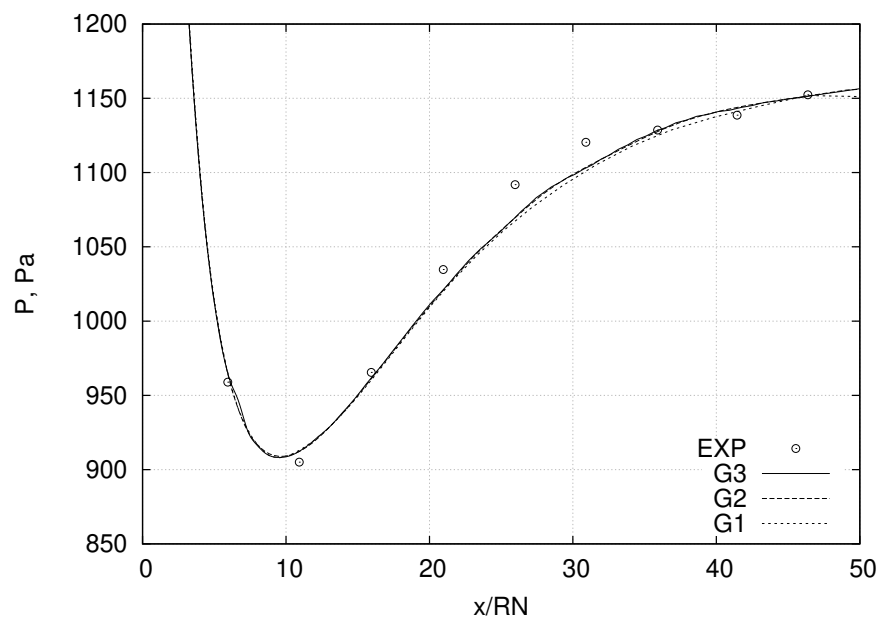


Figure 5.24: Surface pressure distribution for axisymmetric simulations. Experimental data from Roy et al. [188].

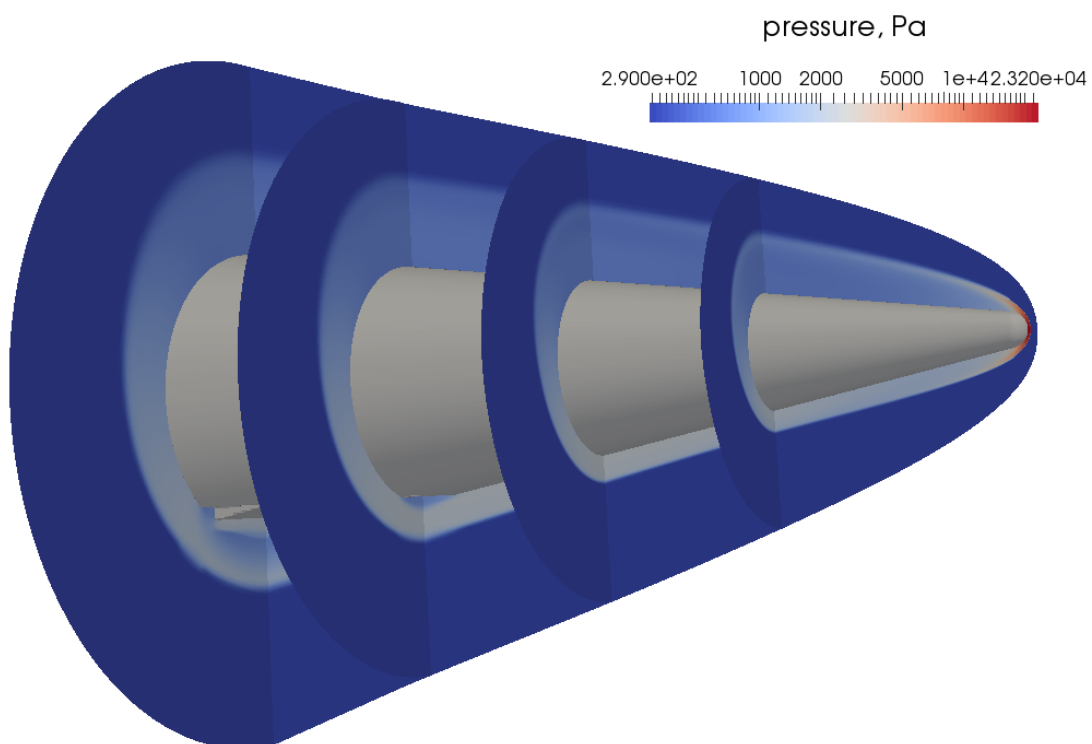
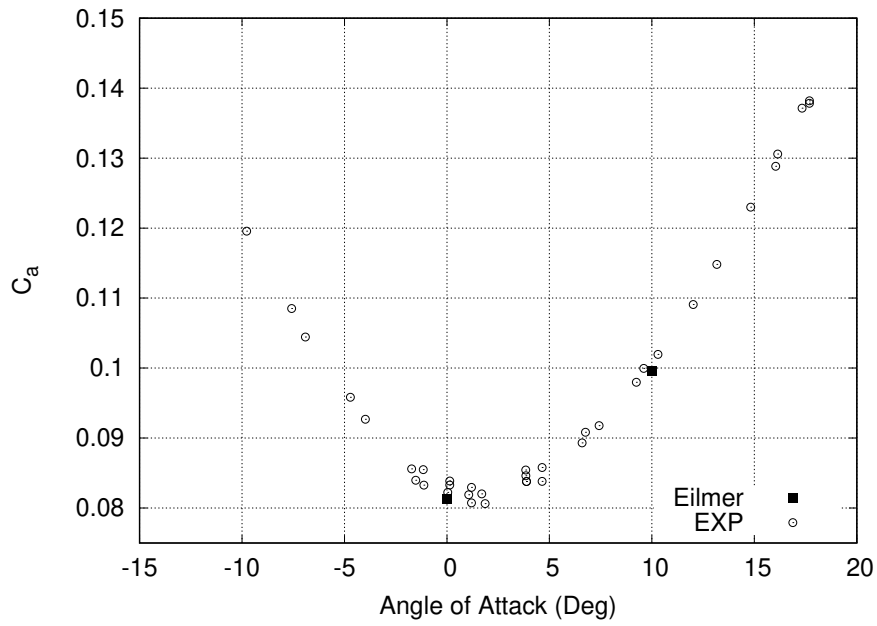


Figure 5.25: Colour map of pressure for JCEAP 10° flap configuration at -5° angle of attack.



5.3.6 Summary

The results from a validation study of the flow solver were presented in this section. Several high-speed flow configurations with strong inviscid/viscous interactions were simulated. The numerical results were compared to published experimental data. A majority of the comparisons were favourable. However, there are some details that the flow solver does not capture. These discrepancies, particularly in the interaction zone of a shock wave impinging on a turbulent boundary layer, and also for low-density high Mach number flows, are not unusual, and references to several published works that have observed similar behaviour have been presented. Despite these difficulties, we still believe that the RANS unstructured grid solver will be a useful discriminator between designs for optimisation work.

CHAPTER 6

ADJOINT SOLVER

This chapter documents the implementation of the numerical solver employed to solve the adjoint system of equations. The procedure for evaluating the total derivative is also presented. The chapter concludes by verifying the accuracy of the adjoint sensitivities by comparison to a direct-differentiation method using complex-variables. A helpful technique for debugging the adjoint operator is also outlined.

6.1 Discrete adjoint equations

The derivation of the discrete adjoint method was presented in Chapter 3. The adjoint system of equations are reprinted as Equation 6.1 for convenience.

$$\begin{bmatrix} \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \end{bmatrix}^T \lambda = \left(\frac{\partial J}{\partial \mathbf{Q}} \right)^T. \quad (6.1)$$

The primary challenge of implementing an adjoint solver is constructing the adjoint operator. Unlike the implicit operator of the flow solver, the adjoint operator must be an exact linearisation of the flow solver routines, and should not be some approximation. It has been shown that approximations such as forming the adjoint operator using a first-order stencil or assuming “frozen” eddy viscosity result in derivatives that are often inaccurate and can even be of incorrect sign [51]. The details about the formation of the adjoint operator and the numerical solution of Equation 6.1 will be covered in the subsequent sections.

6.2 Constructing the discrete adjoint operator

To construct the discrete adjoint operator requires an exact linearisation of the discretised residual vector. Several methods for performing the linearisation were outlined in Chapter 3, for this work, the complex-step derivative approximation is used.

When applying the complex-step derivative approach to linearise the residual vector, a primitive flow variable is perturbed along the imaginary axis in the complex plane by some perturbation size h . The flow solver residual routine is then executed, and the imaginary part is mined to evaluate one entry of the matrix, as shown in Eq. 6.2 for the i^{th} residual vector entry and j^{th} primitive flow variable. In this approach, by looping through all primitive variables, over all cells, the adjoint operator $(\frac{\partial \mathbf{R}}{\partial \mathbf{Q}})^T$ is constructed row by row, and stored in compressed row storage (CRS) format.

$$\frac{\partial \mathbf{R}_i}{\partial \mathbf{Q}_j} = \frac{Im[\mathbf{R}_i(\mathbf{Q}_j + ih)]}{h} \tag{6.2}$$

In practice, for a perturbed flow variable, the residual is not evaluated over the entire domain. Instead, to improve efficiency of construction, the residual is only evaluated on a subset of cells. We call this the residual stencil. The residual stencil is composed of all cells that will have an imaginary contribution from a perturbation of a particular cell. The residual stencil for the Euler equations in two-dimensions is illustrated in Figure 6.1. Although this example uses a structured arrangement of elements, the solver works on unstructured grids and is not limited to quad or hex cells.

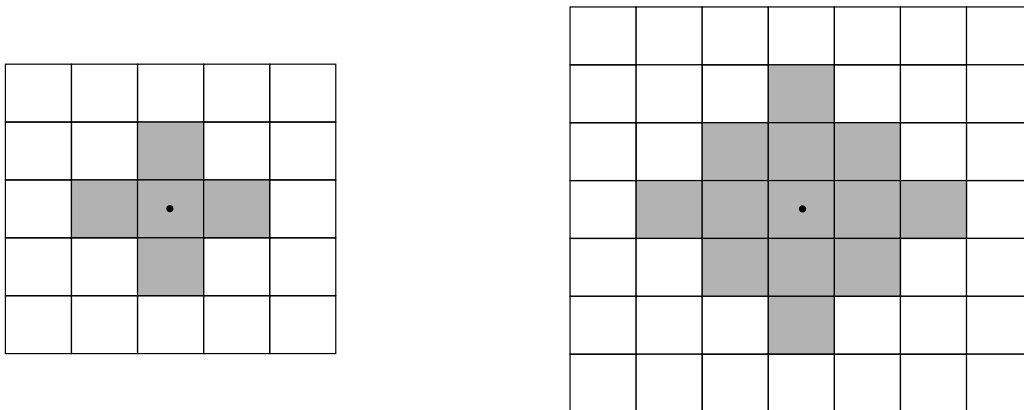


Figure 6.1: First-order residual stencil (left), second-order stencil (right), (·) denotes perturbed cell.

6.2.1 Parallelisation

The construction of the adjoint operator is parallelised using the same domain decomposition as the

flow solver. Following the approach by Saad [189] for solving linear systems on decomposed domains on distributed systems, each block in the decomposition stores a local Jacobian matrix (A_{loc}), and an external Jacobian matrix (A_{ext}), illustrated in the schematic presented as Figure 6.2(a). The local Jacobian matrix is a square matrix that stores the sensitivities of local cell residuals from perturbations of local cell primitive variables. The external Jacobian matrix is a non-square matrix that stores all the entries for perturbations of local cells that affect the residual calculation of cells in neighbouring blocks. Note that, although A_{ext} is represented as spanning the entire global matrix in Figure 6.2(a), in practice, this sub-matrix is a very sparse matrix. To achieve an exact linearisation for second-order spatially accurate simulations using a nearest-neighbour least-squares approach, the Jacobian routine requires one extra layer of ghost cells at block connections compared to the flow solver, as shown in Figure 6.2(b).

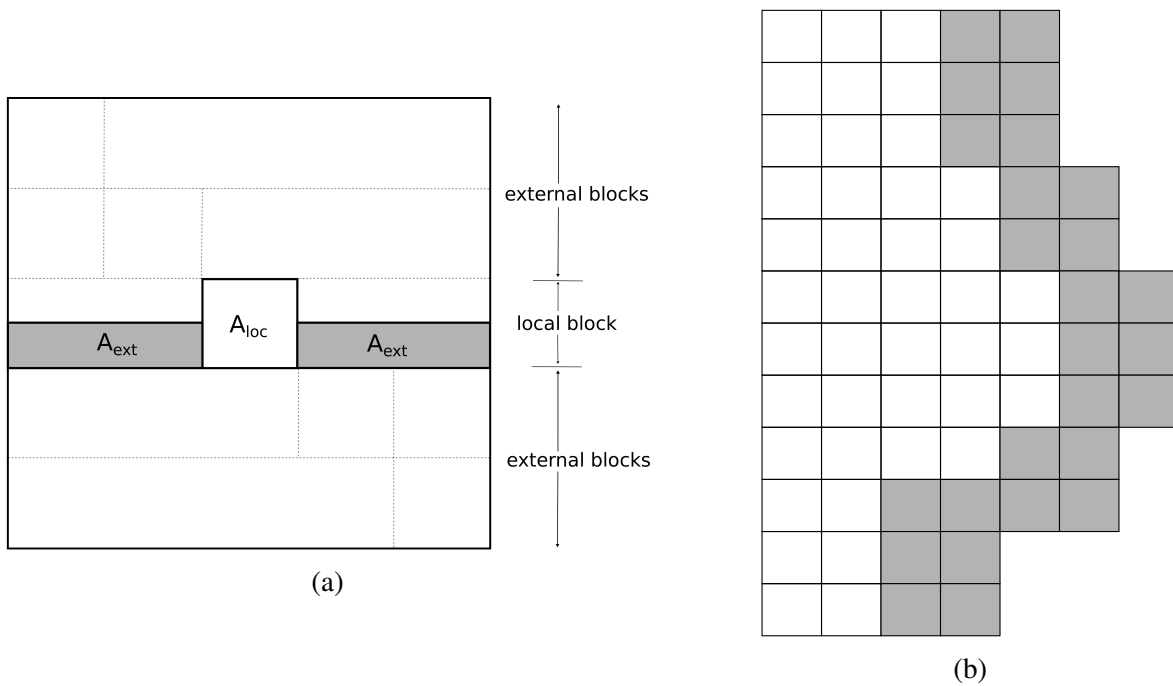


Figure 6.2: (a) Domain decomposed Jacobian. Each local block stores a local Jacobian matrix (A_{loc}) and an external Jacobian matrix (A_{ext}), (b) Domain decomposed block-block boundary, shaded cells are cells from the neighbouring block required in the external Jacobian construction.

6.2.2 Boundary conditions

Boundary conditions in the adjoint solver must be handled explicitly [190]. The explicit boundary conditions for the Jacobian can be written as Eq. 6.3, where R and Q are as defined above, and q

represents the ghost cell flow state variables.

$$\frac{\partial R}{\partial Q} = \left. \frac{\partial R}{\partial Q} \right|_{\text{int}} + \frac{\partial R}{\partial q} \frac{\partial q}{\partial Q} \quad (6.3)$$

The second term in Eq. 6.3 represents the application of the physical boundary conditions. The sensitivity $\frac{\partial R}{\partial q}$ is simply the resultant perturbation of the residual vector entry from a perturbation of a ghost cell flow state variable. It may be formed using a similar approach as the internal cells. The second sensitivity, $\frac{\partial q}{\partial Q}$, can be thought of as a transform from the internal flow state to the ghost cell flow state. This transform can be evaluated using the complex variable method, by perturbing an internal cell's flow state, applying the physical boundary condition, and mining the complex part of the ghost cell flow state. To improve efficiency for applying adjoint boundary conditions in Eilmer, our physical boundary conditions may be applied on a per interface basis. To provide an exact linearization of the flow solver, the physical interfacial boundary condition information should be communicated to neighbouring blocks during the construction of the external Jacobian matrix (A_{ext}). To simplify the implementation, the communication of this information is omitted in Eilmer. Although this omission results in some small error in several entries of the Jacobian, the effect is limited to only those entries where a physical boundary condition is intersected by a block-to-block boundary condition, i.e. in the corners of the decomposed domains. In Section 6.5, it is shown that this simplification does not substantially reduce the accuracy of the adjoint sensitivities.

6.2.3 Frozen limiter

A known problem for supersonic and hypersonic flow-codes is that the limiters, necessary to ensure stability, can oscillate causing a stalling of the convergence. This is sometimes referred to as limiter ringing [104, 105]. The level of convergence achieved before stalling is problem dependent. Since the adjoint method is predicated on the assumption that the residual vector has been reduced to machine precision (see Eq. 3.4), performing an adjoint solve on a stalled flow field residual can cause the adjoint solver to either diverge, or converge on an inaccurate solution. Furthermore, the routines in the flow code that evaluate the limiter values have been observed to be sensitive to complex variable perturbations [105]. This has also been observed in this current work. One approach to overcome this limitation is to freeze the limiter after stalled convergence [101]. In this approach, once convergence has stalled, the limiter value for each cell is frozen, i.e. the limiter value is no longer re-evaluated, but is still applied. When applying the Venkatakrisnan limiter [144], a scalar value (ϕ) for each primitive flow state variable is calculated and stored at the cell-centers. The flow state reconstruction

with gradient limiting is then evaluated as

$$Q_{j+1/2} = Q_j + \phi \cdot \nabla Q_j \cdot \frac{1}{2} \Delta \vec{x}_{jk}, \quad (6.4)$$

where $j + 1/2$ denotes a cell interface, and \vec{x}_{jk} is the cell-center position vector that points from cell j to cell k (refer to Section 4.2.2 for more details on the reconstruction procedure). When the limiter is frozen, ϕ is no longer re-evaluated, however, Eq. 6.4 is still employed during the reconstruction procedure using the frozen values. Thompson [105] applied this approach for the Van Albada [191] and Van Leer [192] limiters, in the context of high-speed adjoint optimisation. In this context, the adjoint solver does not re-compute the limiter values when constructing the Jacobian, but, instead, uses the frozen limiter values from the flow solver.

6.3 Numerical solution of the adjoint equations

In addition to the adjoint operator detailed in the previous section, the right hand side of the adjoint system of equations requires the formation of the partial derivative of the objective function with respect to the flow state variables, $\frac{\partial J}{\partial \mathbf{Q}}$. Entries from this partial derivative are evaluated by perturbing each flow state variable sequentially in the complex domain, and executing the objective function routine. The imaginary part of the objective function is then mined to construct the sensitivity vector. Since this vector is typically filled with mostly zeros, to improve efficiency, it is possible to perform targeted differentiation. In this approach, only flow states in cells that will effect the objective function are perturbed, the remaining entries are automatically filled with zero entries. For example, if the objective function is drag along a boundary, then only cells near the boundary need explicit evaluations of the perturbed objective function routine.

The resulting linear system (Equation 6.1) is solved using a preconditioned, restarted GMRES method. The GMRES algorithm is the same as that for the flow solver outlined in Section 4.3.5, however, the adjoint operator is explicitly constructed and stored in memory, and no column or row scaling is applied. Preconditioning of the system is accomplished by an incomplete LU factorisation of a transposed first-order flow Jacobian matrix using zero fill-in, ILU0. It is noted that to provide robust convergence for turbulent flows, the first-order transposed flow Jacobian matrix should only contain entries arising from the nearest-neighbour cells (i.e. first-order convective stencil). In practice, matrix entries with non-zero values outside of the convective first-order stencil, i.e. arising from any viscous contributions, are dropped. The use of Jacobi and block-Jacobi precondition matrices were also explored for their efficiency. However, poor convergence was observed.

6.4 Total derivative calculation

For the discrete adjoint method, the objective function total derivative is given by

$$\frac{dL}{d\mathbf{D}} = \frac{\partial J}{\partial \mathbf{D}} + \left[\frac{\partial \mathbf{R}}{\partial \mathbf{D}} \right]^T \lambda. \quad (6.5)$$

The residual vector partial derivative, $\frac{\partial \mathbf{R}}{\partial \mathbf{D}}$, is evaluated by perturbing each design variable sequentially in the complex domain, executing the mesh deformation routine, and then evaluating a flow solver residual routine over the entire flow domain. The imaginary part of the residual vector entries are then mined to construct the sensitivity matrix. So for ‘n’ design variables, ‘n’ residual evaluations are required. This is still typically a small cost compared to the number of residual evaluations required to converge a flow field calculation. As discussed in Section 3.3, the construction of $\frac{\partial \mathbf{R}}{\partial \mathbf{D}}$ can be computationally expensive as a result of the ‘n’ calls to the mesh deformation routine. However, for the particular mesh movement strategy implemented in this work (outlined in Section 7.2.2), this was not a difficulty. Note that the evaluation of the residual vector sensitivities includes the evaluation of grid sensitivities because the design variables do not explicitly appear in the residual equations [193]. The remaining partial derivative, $\frac{\partial J}{\partial \mathbf{D}}$, is evaluated similarly, by perturbing each design variable sequentially in the complex domain, executing the mesh deformation routine and then evaluating the objective function routine.

In practice, the code performs the following steps for each of the ‘n’ design variables:

1. Perturb design variable D_i in the complex domain.
2. Execute surface parameterisation routine:
 - \Rightarrow *Sensitivities are propagated to the mesh nodes along the parameterised surface.*
3. Execute mesh deformation routine:
 - \Rightarrow *Sensitivities are propagated to all mesh points.*
4. Execute flow solver residual routine:
 - \Rightarrow *Sensitivities are propagated to conserved residual quantities.*
5. Evaluate $\frac{\partial \mathbf{R}}{\partial D_i}$.
6. Execute objective function routine:
 - \Rightarrow *Sensitivities are propagated to the objective function scalar quantity.*
7. Evaluate $\frac{\partial J}{\partial D_i}$.

In this way only ‘n’ calls to the mesh deformation routine are required.

6.5 Verification

The role of code validation has no meaning in the context of the adjoint solver since there exists no experimental means of determining shape sensitivities. The verification of the adjoint solver outlined above is documented in this section.

6.5.1 Verification procedure

The adjoint gradients are verified by comparison to a direct differentiation via complex variables. Since both methods use complex variables, their evaluated gradients are expected to agree to several significant figures. This approach is now considered to be the standard practice for adjoint verification [24].

In complex variable direct-differentiation, a design variable is perturbed along the imaginary axis of the complex domain. The surface parameterisation routine and mesh deformation routine are both executed to propagate the sensitivities through to the mesh points. The flow solver is then run until a steady-state has been achieved. The imaginary component of an objective function evaluation is then used to compute the design sensitivity. This is repeated for all design variables. Mathematically, this procedure is written as

$$\frac{\partial J}{\partial \mathbf{D}_j} = \frac{\text{Im}[J(\mathbf{Q}(\mathbf{D}_j + ih), \mathbf{X}(\mathbf{D}_j + ih))]}{h}. \quad (6.6)$$

The application of complex variable direct-differentiation is the equivalent of applying forward-mode automatic-differentiation [61, 194]. Further reading on the direct complex variable method can be found in the work by Martins [60].

As discussed in Section 4.3.3, the flow solver itself employs complex arithmetic in evaluating the Fréchet derivative and forming the low-order flow Jacobian used as a preconditioner in the GMRES algorithm. This presents a complication when applying direct differentiation via complex variables since the imaginary part of a complex variable cannot store two independent sensitivities [132] (e.g. one for the design sensitivity and one for the Fréchet derivative evaluation). For this reason, the flow solver is run using real-valued Fréchet derivatives without preconditioning when used in direct-differentiation mode. However, this can lead to poor convergence of the flow solver, so in practice, a three-stage procedure is executed. First, a flow solution is converged to machine precision using the flow solver in its nominal state utilising complex arithmetic. The flow solution is then used to initialise a new simulation using the flow solver with real-valued Fréchet derivatives and with preconditioning switched off. This is typically run for several thousand steps, with a fixed CFL of 1. The limiter is frozen at the end of this stage and stored. This flow solution is then used as an input to the adjoint solver to compute the adjoint sensitivities. Finally, the flow solution is then used to initialise a third flow solver simulation, this time with direct-differentiation mode on. This is run for several thousand steps with a CFL of 1 until the sensitivities have converged to machine precision. Note that this

simulation also uses the frozen limiter values. The two sensitivities are then compared. This procedure is similar to the approach first reported by Thompson [105].

6.5.2 Verification test case

The chosen geometry and flow conditions for the verification test case are taken from the experimental campaign by Simenonides et al. [195], and are those for the Mach 6, 7.5-degree compression corner. The flow over the model was ideal air. A schematic of the problem is presented in Figure 6.3. The wedge surface was parameterised using a 4-point Bézier curve, the first and last points were fixed, the remaining two points, annotated on Figure 6.3 were the design variables. More details on the Bézier surface parameterisation can be found in Chapter 7. The mesh consisted of 50 cells in the x -direction and 25 cells in the y -direction, with cell stretching in the y -direction to capture the boundary layer. Although this mesh is not well resolved for engineering purposes, it serves its purpose to verify the adjoint gradients. The adjoint solver was verified for use with the Euler, Navier-Stokes and Reynolds-Averaged Navier-Stokes equations in Eilmer. Slip-wall boundary conditions were used for the Euler solutions, and for the viscous simulations, a no-slip isothermal wall at 300 K. The AUSMDV flux calculator was employed for this verification test. The blending parameter (s in Eq. 37 from Wada and Liou [99]) in the AUSMDV flux calculator that blends the AUSMD and AUSMV schemes was set to a fixed value of 0, such that both the AUSMD scheme and AUSMV scheme had equal contributions to the calculated flux. This was to accommodate the direct complex variable solver which was observed to be sensitive to the numerical routine employed in computing this parameter.

The sensitivities of the inviscid drag component acting on the wedge section are presented in Table 6.1. Here the inviscid drag is calculated as

$$F_D = \int_{\Omega} p(x, y) dA, \quad (6.7)$$

where Ω denotes the wedge surface and A is the area. The sensitivities presented in Table 6.1 are then $\frac{\partial F_D}{\partial D_0}$ and $\frac{\partial F_D}{\partial D_1}$. As expected, there is an excellent agreement between the two methods. A reduction in accuracy is observed for the multiblock RANS sensitivities, as a result of the simplified boundary implementation outlined in Section 6.2.2. Table 6.1 shows the effects of the simplification when the domain is decomposed into 4 and 8 blocks. The relative error in D_0 increased as the number of blocks increased, however, the relative error in D_1 decreased. The effect of the simplification has been observed to reduce as the number of cells in each block increases since the simplification is only applied at the intersection of a block-to-block boundary condition with a physical boundary condition. Increasing the number of cells in a block effectively decreases the ratio of simplified entries to exact entries in $\left(\frac{\partial R}{\partial Q}\right)^T$. To demonstrate that the solver is suitable for a practical decomposition, an additional verification test case was performed using a mesh consisting of 200 cells in the x -direction and 100 cells in the y -direction decomposed into 32 blocks. The observed relative error for this configuration

was of the same order of magnitude as the previous results. It will be demonstrated in Chapter 8 that this level of accuracy is sufficient for gradient-based optimisation.

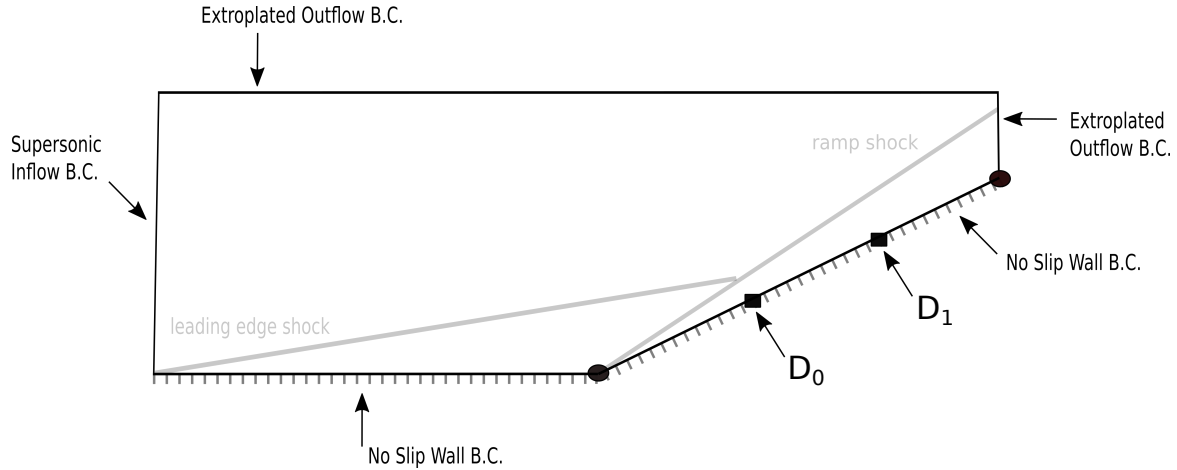


Figure 6.3: Schematic of flat-plate wedge geometry with Bézier points superimposed on wedge.

6.5.3 Debugging the adjoint operator

The verification procedure outlined above is effective for demonstrating the accuracy of the implementation. However, it provides no useful information to the code developer as to where an existing bug might exist in constructing the adjoint operator. Carpentieri [196] presented two verification techniques for the adjoint operator: (a) a real-valued Fréchet derivative approach, and (b) an automatic differentiation (AD) approach. Both methods evaluate matrix-vector products: the former is easier to implement but less accurate, and the latter requires complex AD tools (e.g. Tapenade [197]) but provides analytically accurate products. If the real-valued Fréchet derivative is replaced with a complex-step Fréchet derivative, then a method for evaluating matrix-vector products is both simple to implement and provides analytically accurate products [198]. This method is a powerful tool for debugging the adjoint operator $\left(\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}\right)^T$ since it can be used to give the location of an erroneous entry.

The procedure for debugging is outlined here. Ultimately we seek the relative error vector

$$\boldsymbol{\varepsilon} = \frac{\mathbf{Z}_1 - \mathbf{Z}_2}{\mathbf{Z}_1}, \quad (6.8)$$

where \mathbf{Z}_1 is the resulting vector from an explicit matrix-vector multiplication of the flow Jacobian (or transpose of the adjoint operator) and some test vector (\mathbf{v}),

$$\mathbf{Z}_1 = \left[\left(\frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \right)^T \right]^T \mathbf{v}, \quad (6.9)$$

Table 6.1: Sensitivities of the inviscid drag component acting on the wedge section. SB = Singleblock; 4MB = 4 blocks; 8MB = 8 blocks.

Equations	Method	D_0	D_1
Euler (SB)	direct	-4.594183086029831e+02	1.803545660822891e+02
	adjoint	-4.594183086029751e+02	1.803545660822925e+02
Euler (4MB)	direct	-4.578044643486712e+02	1.795628739115091e+02
	adjoint	-4.578044643486759e+02	1.795628739114956e+02
Navier-Stokes (SB)	direct	-1.5239846655859574e+03	-1.6470562968334468e+02
	adjoint	-1.5239846655944107e+03	-1.6470562969431143e+02
Navier-Stokes (4MB)	direct	-1.5239805058541165e+03	-1.6470271844914430e+02
	adjoint	-1.5240314478664918e+03	-1.6476360643609402e+02
RANS (SB)	direct	-8.683408385995121e+02	1.900431473981142e+03
	adjoint	-8.683408385995051e+02	1.900431473981131e+03
RANS (4MB)	direct	-8.683156434934184e+02	1.900457510584376e+03
	adjoint	-8.683653295253905e+02	1.900701298865737e+03
RANS (8MB)	direct	-8.6809937520721110e+02	1.9004431301468817e+03
	adjoint	-8.6833107690871600e+02	1.9004792612670160e+03

and \mathbf{Z}_2 is the Fréchet derivative,

$$\mathbf{Z}_2 = \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \mathbf{v} \approx \frac{Im[\mathbf{R}(\mathbf{Q} + \mathbf{v}ih)]}{h}. \quad (6.10)$$

Each entry in the relative error vector represents the error for a particular flow variable in a particular cell.

To demonstrate its use, consider performing the procedure on the compression-corner verification case outlined above. The error vector for the nominal code is shown in Figure 6.4. It is observed that a majority of entries achieve close to machine precision accuracy, as expected. The entries that have a slightly higher error have been identified as relating to the y-component of momentum along wall boundaries. The exact cause of this behaviour has not been identified. Regardless, this agreement is sufficient to achieve analytically accurate sensitivities as shown in Table 6.1. Figure 6.5 shows the error vector for the same problem but with a deliberate coding bug in the Jacobian matrix entry for a particular cell. In this instance, the deliberately placed bug was to switch the sign of the matrix entry. It is observed that this bug has caused four entries in the error vector, related to this particular cell, to exhibit a substantial error. During code development, this debugging technique can be used

to visualise where certain erroneous cells are in the flow domain, hence narrowing down the lines of code to be reviewed.

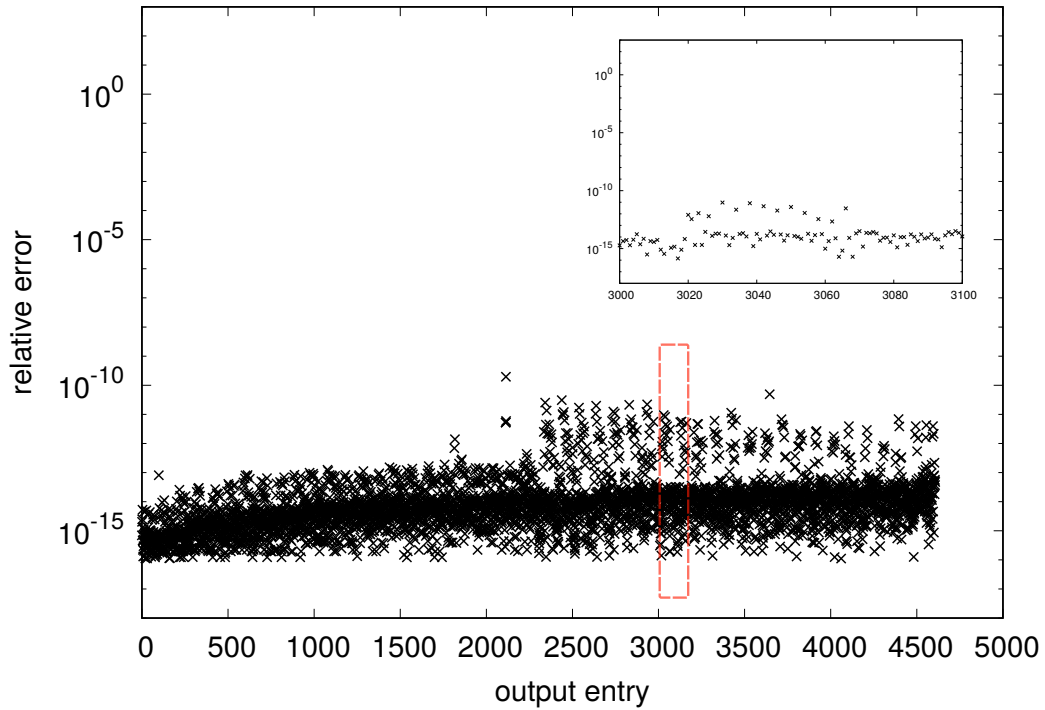


Figure 6.4: Matrix-vector product error vector (ϵ) for nominal case.

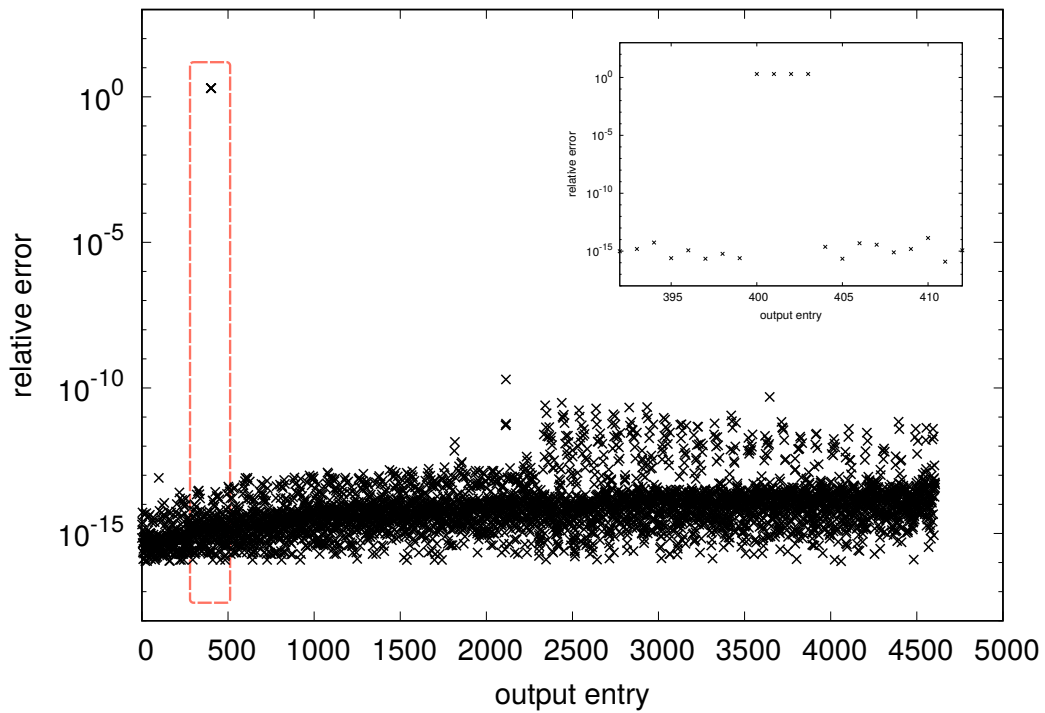


Figure 6.5: Matrix-vector product error vector (ϵ) for case with implemented bug.

SHAPE OPTIMISATION METHODOLOGY

The gradient-based optimisation methodology applied in this work is separated into five distinct steps: (i) surface parameterisation; (ii) mesh generation and movement strategy; (iii) gradient evaluation; (iv) objective function evaluation, or in the context of CFD-based optimisation, flow solution evaluation; and (v) numerical optimisation. Steps (iv) and (iii) were covered in Chapters 4 and 6 respectively. This chapter will detail the remaining steps: surface parameterisation, mesh generation and movement strategy, and numerical optimisation.

7.1 Surface parameterisation

7.1.1 A brief review of surface parameterisation methods

Surface parameterisation methods can be categorised into three types [199]: discrete, computer-aided design (CAD) based, and free-form deformation. In the discrete approach, every node on a surface is assigned as a design variable. This allows easy parameterisation for complex geometries, facilitates unconstrained shapes, allows easy calculation of movement sensitivity and enables strong local control. However, the discrete approach does not guarantee a smooth surface. More recently Soto et al. [200] presented a pseudo-shell discrete method which had the favourable trait of maintaining smooth surfaces.

For the CAD-based approach, the surface is parameterised using a CAD software package. Typically CAD software defines a surface by a combination of B-splines. Parameterisation using CAD software allows consistent parameterisation with a smaller number of design variables and allows smoothness of the surface to be controlled. However, the shape is limited to that of the parameterisation available in the CAD software, it is difficult to obtain the sensitivities, and allows very little local control (depending on the form of parametric curve). Anderson et al. [201] modelled two-dimensional

wing profiles using B-splines outside of a CAD environment. This allowed the sensitivities to be calculated directly from the surface definition, however, it is difficult to parameterise complex three-dimensional shapes using this method.

Free-form deformation is a type of parameterisation typically used in computer animation. In this approach, the surface being parameterised is contained within a three-dimensional hull. Movement of the nodes which define the hull are mapped to movements of the parameterised surface. Samareh presented this approach for aerodynamic shape optimisation in [202]. Free-form deformation allows consistent parameterisation of the surface; sensitivities to be determined analytically; allows for strong local control with fewer design variables than a discrete approach; and the smoothness is retained.

7.1.2 Bézier curves

In the current work, all design surfaces are parameterised by n-order Bézier curves. Mathematically, a parametric Bézier curve is defined by

$$P(t) = \sum_{i=0}^n B_i J_{n,i}(t) \quad 0 \leq t \leq 1 \quad (7.1)$$

where B_i are the Bézier control points, and the Bézier, or Bernstein, basis or blending is

$$J_{n,i}(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad (7.2)$$

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad (7.3)$$

Here, n , the degree of the Bernstein basis function and thus of the polynomial curve segment, is one less than the number of points in the Bézier curve. An example of a cubic Bézier curve is shown in Figure 7.1. For more details on Bézier curves, refer to the textbook by Rogers [203]. For the design work presented in Chapter 8, each Bézier control point has a fixed x-coordinate and a variable y-coordinate. Bézier curves were selected as a simple, robust solution to surface parameterisation in two-dimensions when the design surfaces have little complexity. Future three-dimensional design work will explore the use of Bézier surfaces or free-form deformation.

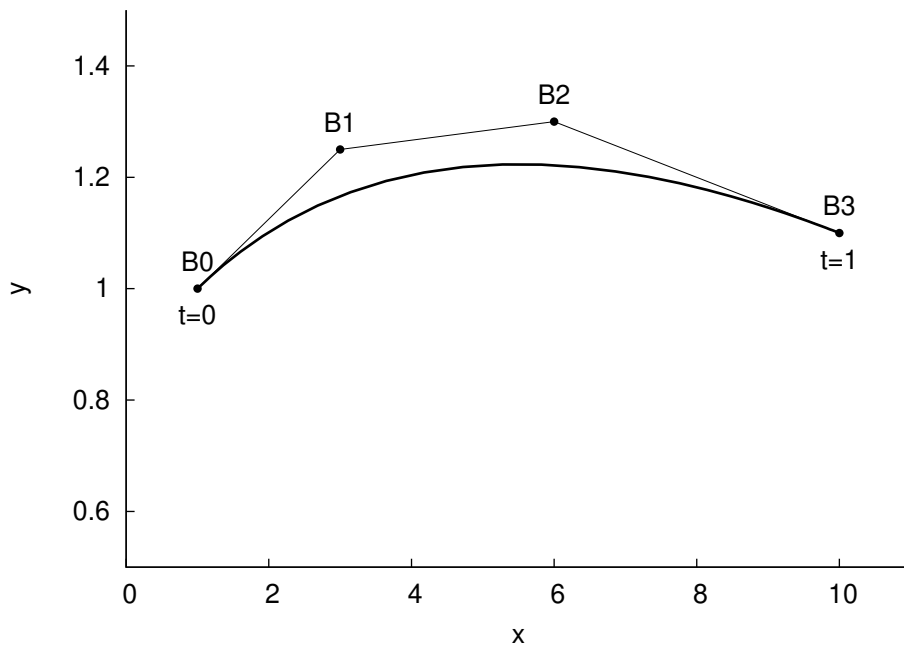


Figure 7.1: Cubic Bézier curve.

7.2 Mesh generation and movement strategy

The meshes used throughout this work have been generated using the Pointwise™ software (V18.1 R1) and domain decomposition of the mesh is achieved via METIS [141] (V5.1.0). The optimisation in this work requires continuous modification of geometric surfaces via perturbations of the Bézier control points. The deformation of the computational domain boundary requires consistent deformation of the internal mesh points to ensure meshes of sufficient quality throughout the optimisation procedure.

7.2.1 A brief review of mesh deformation methods

A straightforward method to accomplish mesh deformation is through linear interpolation of the surface movement. This is simple to apply to structured grids using an algebraic mesh movement strategy [204]. However, these methods do not extend to unstructured grids. For unstructured grids, mesh movement has been achieved using the spring tension analogy [205–207] and the torsional spring analogy [208, 209]. Nielsen et al. [210] found, however, that the use of spring analogies lacked robustness necessary for a design environment where large mesh movements are required. In the optimisation of three-dimensional aerofoils, Nielsen et al. [210] assumed the mesh obeyed the isotropic linear elasticity relations. It was shown via numerical experiments that this method was favourable for large geometry movement. An alternative method that perturbed the mesh by solving an elliptic equation with nonlinear diffusion has also been applied [211]. This method was shown to accommodate

large geometry deflection also. The most straightforward approach to mesh deformation for unstructured grids is inverse distance weighting (IDW). This approach has been shown to produce mesh qualities comparable to alternate methods at a fraction of the computational cost of the previously outlined methods [212]. For this reason, IDW has been applied in this work.

7.2.2 Inverse Distance Weighting method

Given a perturbed design surface, the new interior mesh point coordinates are given as

$$\mathbf{p}^{\text{new}} = \mathbf{p}^{\text{old}} + w(\mathbf{p}), \quad (7.4)$$

where $w(\mathbf{p})$ is the change in mesh point coordinates, given via the IDW method as

$$w(\mathbf{p}) = \frac{\sum_{i=1}^n v_i \phi(r_i)}{\sum_{i=1}^n \phi(r_i)}. \quad (7.5)$$

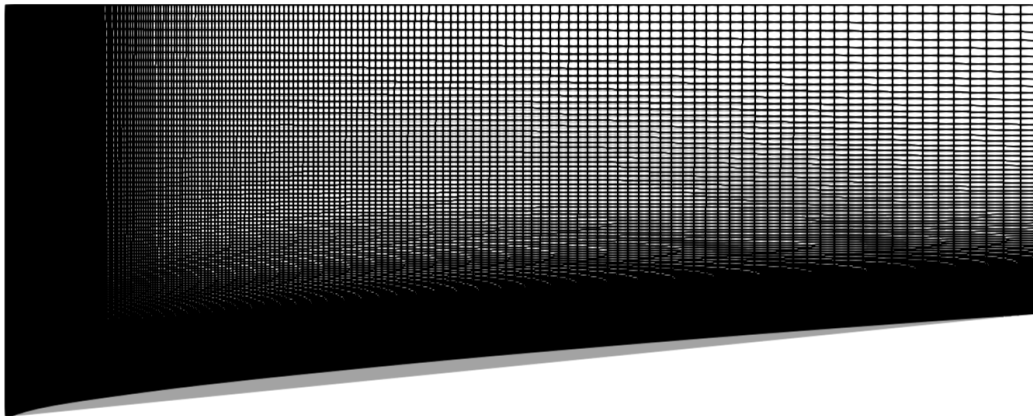
The vector v stores the change in mesh point coordinates for all points along the boundaries,

$$v_i = \mathbf{p}_{\text{boundary},i}^{\text{new}} - \mathbf{p}_{\text{boundary},i}^{\text{old}}, \quad (7.6)$$

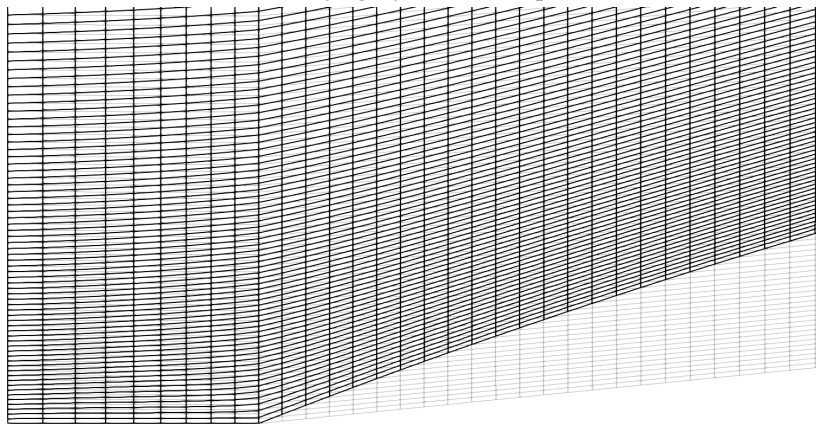
and the general weighting function $\phi(r_i)$ is defined as

$$\phi(r_i) = r_i^{-c}, \quad \text{where } r_i = \|\mathbf{p} - \mathbf{p}_{\text{boundary},i}\|. \quad (7.7)$$

Here c is set to 2 in the current implementation. In practice, after the design surface has been updated, the IDW routine loops over all interior mesh points and evaluates $w(\mathbf{p})$ via Equation 7.5. The summation in Equation 7.5 is over ‘n’ boundary mesh points. It is noted that only mesh points along the design surface will have a non-zero entry in v . A minor complication in implementing the IDW method is that it is not obvious whether to use the old or new boundary vertex coordinates in the computation of the weighting function (Equation 7.7). For this work, we choose to use the new vertex coordinate, since this provided better quality meshes for the types of problems presented. An example of a mesh perturbed using the IDW method is shown in Figure 7.2.



(a) Baseline mesh for conical body (grey lines) and perturbed mesh (black lines).



(b) Close-up of nose region.

Figure 7.2: Example of mesh deformation via IDW.

7.3 Numerical optimiser

The flow and gradient evaluation routines in the developed framework are coupled to the open-source optimisation software, DAKOTA [213]. DAKOTA provides access to several gradient-based numerical optimisation algorithms. The Fletcher-Reeves Conjugate Gradient (FRCG) method [214], has been used in the optimisation work presented in Chapter 8. FRCG is a method for solving unconstrained, nonlinear optimisation problems. Constraints may be handled via a penalty method.

7.4 Design optimisation algorithm

The design optimisation algorithm applied in this work is shown in Figure 7.3. The design loop takes as an input a parameterised surface and a computational grid. The loop begins with the flow solver computing the steady-state flow solution and evaluating the objective function. The converged flow solution is used as an input to the adjoint solver to construct the adjoint operator. The adjoint solver then determines the adjoint variables. The remaining partial derivatives are evaluated, and these are combined with the adjoint variables to evaluate the total derivative (or shape sensitivities). DAKOTA uses the shape sensitivities to take a step in the design space and returns an updated set of design parameters. If the shape sensitivities are below a certain magnitude or the relative change in the objective function evaluation is below a certain tolerance, then the optimiser returns the optimum parameters. Otherwise, the new design parameters are given as an input to the surface parameterisation routine to generate the new surface and the mesh deformation routine is executed to update the computational grid. The design loop is then repeated until one of the convergence criterion is satisfied.

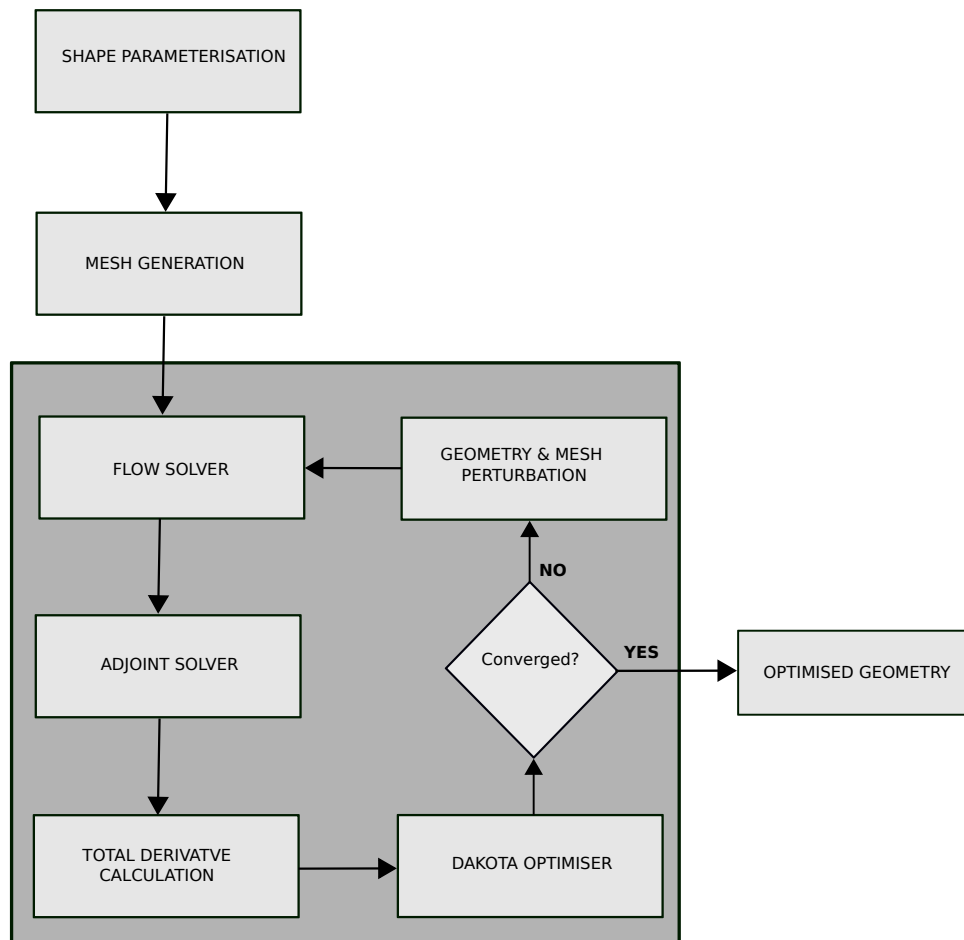


Figure 7.3: Design optimisation algorithm.

CHAPTER 8

OPTIMISATION APPLICATIONS

This chapter is to demonstrate the application of the developed optimisation package to shape design problems in high-speed flows. Two optimisation applications have been selected for this purpose. The first application seeks to minimise the wave drag of a slender body of revolution. Hypersonic minimum-drag slender bodies of revolution have been reported extensively in the literature. The accuracy of the optimisation package is demonstrated by comparison to several published optimum bodies for high-speed flow. The second application is the redesign of an existing hypersonic inlet. The design of hypersonic inlets is expected to benefit greatly from the addition of RANS-based analysis into the design phase. Consequently, this demonstration is of considerable interest to the hypersonic community.

8.1 Selection of grid topology

It is noted that, even when employing techniques to improve convergence, such as the limiter freezing procedure presented in Section 6.2.3, the achievable residual convergence is often not to a sufficient level for the adjoint system when operating on unstructured tetrahedral elements. An example of this is provided in White et al. [104], where only a 4 order of magnitude drop in the global residual was achieved when simulating a turbulent hypersonic inlet. For this reason, the adjoint work presented herein uses structured quadrilateral elements to achieve robust convergence. It is stressed, however, that despite the grid being made up of structured elements, the underlying flow and adjoint solvers operate on the grid using numerical routines developed for unstructured grids. This methodology is common in computational analyses of complex hypersonic flows [215, 216]. This work focussed on an unstructured grid implementation because we are optimistic that unstructured methods will mature in the near future to the point where residual convergence is no longer an issue.

8.2 Hypersonic minimum-drag slender bodies of revolution

In this application, we seek to find the optimal shape of a slender body of revolution that minimises the wave drag in a hypersonic flow. We constrain our interest to geometries that have a fixed length and base diameter. Both analytic and numerical solutions to this problem have been reported in the literature for flows ranging from supersonic to hypersonic. Perhaps the most well-known solution to the problem is the von Kármán ogive [217]. The derivation of the von Kármán ogive is based on linearised supersonic flow theory, and, as such, is only valid at low supersonic speeds [218]. For hypersonic flows, the optimal bodies differ substantially from the von Kármán ogive, since the flow at elevated Mach number is poorly modelled by linearised flow theory. Several approaches have been applied to deriving hypersonic minimum-drag bodies of revolution. Cole [219] and Eggers et al. [220] employed Newtonian flow theory to the problem. The optimal bodies that were produced are best approximated by a power law with an exponent n of 0.75, or 0.66 when centrifugal forces are included [221]. The general form of the power law is given by:

$$R = R_b \left(\frac{x}{L_0} \right)^n, \quad (8.1)$$

where R_b is the base radius, and L_0 is the body length. Fink [222] studied the hypersonic wave-drag minimisation problem based on hypersonic small-disturbance approximations to the shock-expansion method. The optimum bodies derived in this study were best approximated by a power-law body with an exponent n of $3/4$ (see Figure 2 from [222]). In the early 1990s, Mason et al. [223] performed a parametric study using solutions of the Euler equations to determine the optimum power-law exponent for several fineness ratios and Mach numbers, where fineness ratio is defined as the ratio of the length of the body divided by the diameter, herein denoted as L/D . The results published showed that for an L/D of 3, the optimum geometry was generally independent of the Mach number, and was best approximated by a power-law exponent n of 0.69. The results also showed a weak dependence on the fineness ratio, with a power-law exponent n of 0.7 better approximating the minimum wave-drag for L/D of 5. More recently Sahai et al. [224] applied a numerical optimisation method to derive hypersonic minimum-drag bodies. In the methodology, the flow was modelled by modified Newtonian theory. The published results showed that the optimal power-law exponent n was 0.7 for low L/D , similar to the results of Mason et al. [223]. However, for high L/D the power-law exponent deviated considerably from those previously reported. It is noted that, upon validating the optimum bodies using solutions of the Euler equations, the modified Newtonian theory was shown to poorly approximate the flow for large L/D . For this current work, hypersonic minimum-drag bodies of revolution are derived using the adjoint-based optimisation package. Specific details of the optimisation problem are presented in the following section.

8.2.1 Optimisation problem description

The objective of this optimisation problem is to find the optimum axisymmetric profile that minimises the wave drag subject to a fixed length and base radius. This section will detail several aspects of the problem.

Objective function

The objective function for this problem is simply written as:

$$J = D_{\text{wave}}, \quad (8.2)$$

where surface pressure integration (SPI) is applied along the body to approximate the wave drag (D_{wave}). It is noted that SPI computes the total drag, which for an inviscid flow is comprised of both wave drag and induced drag, e.g. drag generated from lift [39]. However, since axisymmetric bodies flying at 0° angle of attack are considered here, there is no induced drag component. Consequently, the application of SPI is appropriate. Since SPI is a near-field method of drag estimation, some spurious errors caused by numerical diffusion are expected [39, 225, 226]. However, for two-dimensional inviscid calculations, the SPI method has been shown to be relatively accurate in computing drag compared to far-field methods [227]. It will be shown later in this section that SPI provides sufficient accuracy for computing the drag on an axisymmetric body in inviscid flow. It is noted that the objective function is only based on drag. The constraint on length and base diameter is enforced in the choice of design parameters. In other words, the optimiser is asked to search the design space of bodies of revolution with a fixed length and base diameter.

Baseline geometry

The baseline geometry used as the initial state for the optimisation problem is a right circular cone. The cone has a half-angle of approximately 5.71° and a length of 10 m, this gives a base radius of 1 m corresponding to an L/D of 5. A second cone, with a half-angle of approximately 9.46° and a length of 10 m, resulting in an L/D of 3 was also considered in the optimisation study. A schematic of the problem is provided in Figure 8.1. The freestream conditions are presented in Table 8.2. A grid convergence study was performed for both baseline geometries. Here we show the results for the $L/D = 5$ case. The levels of grid refinement are shown in Table 8.2.1. Figure 8.2 shows grid G1. Note that although the simulations are only inviscid, clustering in both the x - and y -direction is applied to allow for a blunting of the leading edge during the optimisation. The simulations used second-order spatial reconstruction with the Venkatakrishnan limiter [101], with interfacial fluxes evaluated using the AUSMDV flux calculator [99]. Table 8.2.1 summarises the results of the calculations. The computed drag (ignoring base pressure) is compared with solutions of the Taylor-Maccoll equations [228,

229]. It is observed that the drag is approaching the value evaluated using the Taylor-Maccoll method monotonically. The relative error between the drag computed from the Euler solutions and that computed by the Taylor-Maccoll solutions is below 0.5% for all grid resolutions. It is also noted that the relative difference in computed drag between grid-level G1 and G3 is approximately 0.07%. Given this excellent agreement across all mesh refinements, G2 is selected as an appropriate refinement for the optimisation problem.

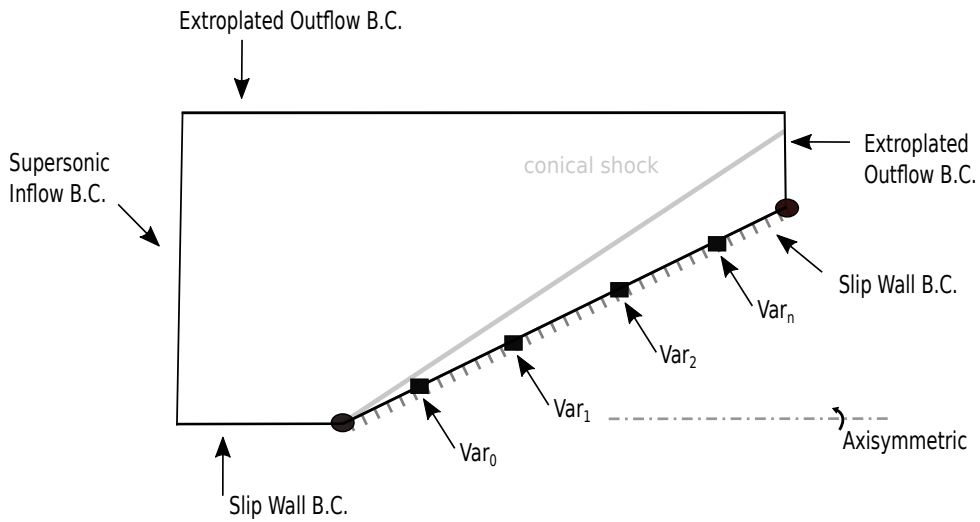


Figure 8.1: Schematic of conical shock simulation.

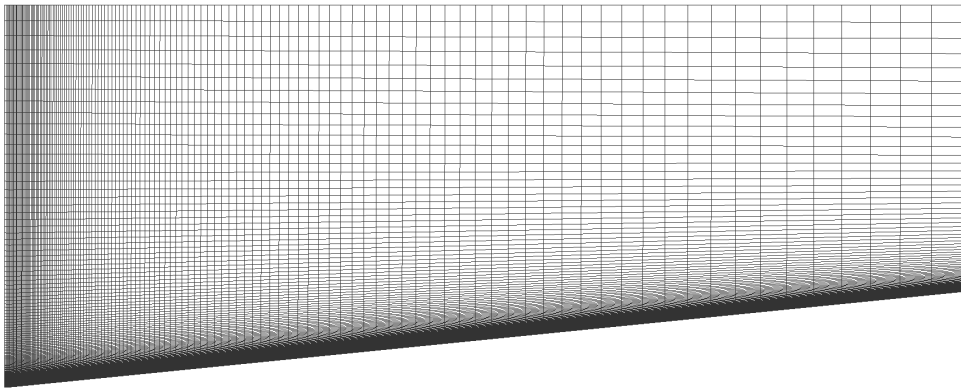


Figure 8.2: Baseline mesh for minimum-drag optimisation.

Table 8.1: Levels of grid refinement and calculated drag for baseline cone simulation.

Grid	Dimensions	$D_{\text{Euler}} (N)$	$D_{\text{Taylor-Maccoll}} (N)$
G1	130×25	54497.2	54664.6
G2	260×50	54517.6	54664.6
G3	520×100	54535.5	54664.6

Table 8.2: Freestream inflow conditions for baseline cone simulation.

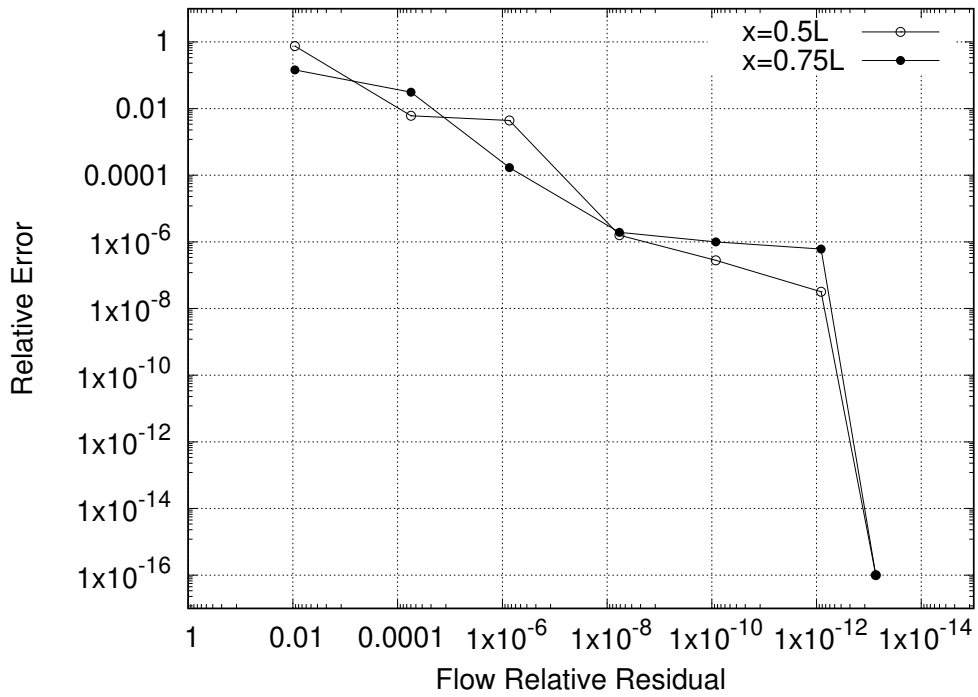
Mach number	6.28
Velocity, $\text{m}\cdot\text{s}^{-1}$	2180.345
Pressure, Pa	10,000.0
Temperature, K	300.0

Surface parameterisation

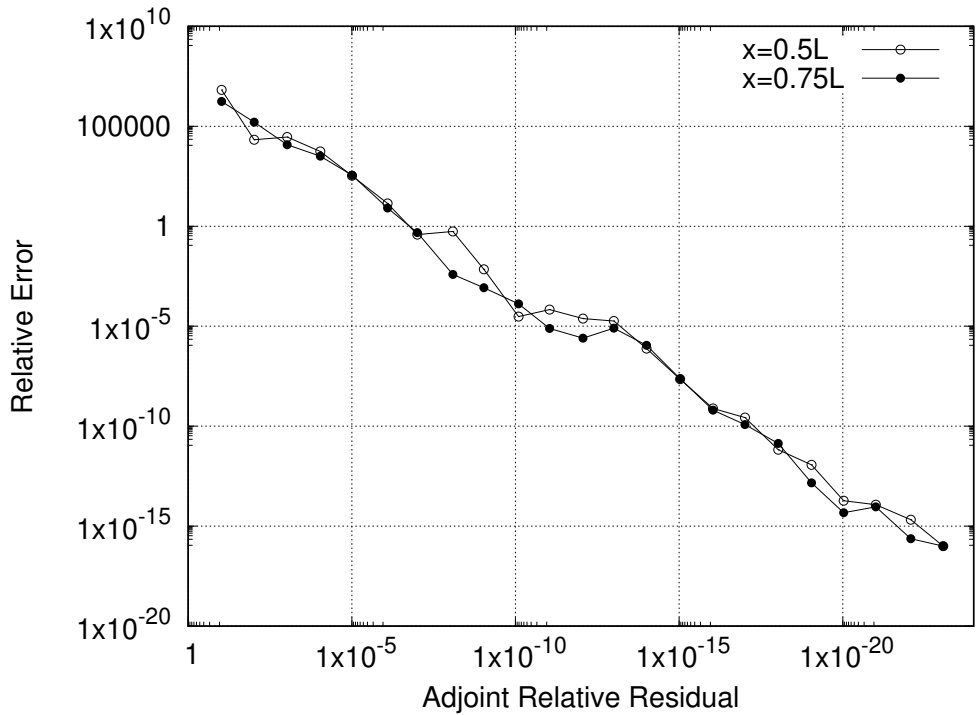
The design surface is parameterised using a Bézier curve. The points are spaced evenly along the x -direction, and the initial y -ordinates are selected such that the Bézier curve recovers the baseline conical body. The first and last points are fixed throughout the optimisation to enforce the fixed length and diameter constraints. The y -ordinates are used as the design variables in the optimisation. More details are provided in Appendix D.

8.2.2 Adjoint solver convergence characteristics

In this section, the convergence characteristics of the adjoint solver are examined. The accuracy of the sensitivity gradients computed by the adjoint method is dependent on the level of convergence of both the flow solution and the adjoint solution. If it is possible to compute sufficiently accurate sensitivity gradients on either a partially converged flow solution or a partially converged adjoint solution, then significant gains in efficiency could be achieved. To explore the convergence characteristics, flow-adjoint solutions were generated for the baseline conical geometry outlined in the previous section. The reference sensitivities, assumed to be exact, are evaluated on a flow solution converged to machine precision and an adjoint solution converged to machine precision. Figure 8.3(a) presents the results from partially converging the flow solution. Figure 8.3(b) presents the results from partially converging the adjoint solution. The variables shown correspond to Bézier points located at $x = 0.5L$ and $x = 0.75L$. It is observed from these results that sensitivity gradients accurate to within 0.1% are achievable by converging the flow solver 7 orders of magnitude and converging the adjoint solver 9 orders of magnitude. Although encouraging, these results should be taken with some caution. The adjoint method is predicated on the flow solution being converged to machine precision, and for second-order spatially accurate solutions of the RANS equations, the adjoint solver has been known to diverge for a poorly converged flow solution. As noted by Burdyslaw [230] the results for the convergence characteristics are most probably problem-dependent.



(a) Flow solver relative residual.



(b) Adjoint solver relative residual

Figure 8.3: Influence of convergence levels on sensitivity estimates for two selected variables.

8.2.3 Results

Minimum-drag profiles were found for fineness ratios of $L/D = 3$ and $L/D = 5$. For the $L/D = 3$ case, optimal profiles for Mach 3, Mach 6.28 and Mach 12 were found using a 20 point Bézier curve and the AUSMDV flux calculator. For the $L/D = 5$ case, optimal profiles for Mach 6.28 were found using combinations of the AUSMDV and Roe flux calculator, and by parameterising the surface with a 20-point and a 40-point Bézier curve. These particular configurations were taken from the work by Mason et al [223]. The objective function histories for each of the cases are shown in Figure 8.4. The optimiser was run for 50 design iterations. In all cases, the optimiser had virtually settled on a minimum point by the 50th iteration. The optimised profiles for $L/D = 3$ are shown in Figure 8.5(a). Also presented in Figure 8.5(a) are several of the previously derived optimum profiles, discussed earlier. The present results show some small Mach number dependence. However, this appears to be only for the Mach 3 case, it is observed that the Mach 6.28 and Mach 12 cases both share the same profile. It is also observed that all three Mach number cases are approximated well by the power-law body with an exponent n of 0.69 near the leading edge, however, the agreement deviates slightly downstream. The optimised profiles for the $L/D = 5$ case are shown in Figure 8.5(b). The results for this case appear to be independent of the number of design variables and also independent of the flux calculator employed. This case exhibits much better agreement with the results from Mason et al. [223], also plotted in Figure 8.5(a) as the power-law body with exponent $n = 0.7$. Overall these results demonstrate that the adjoint-based optimisation package is correctly finding the minimum point for the given design problem. The results presented here also appear to confirm that for large fineness ratio, hypersonic minimum-drag bodies of revolution are best approximated by a power-law with an exponent approximately equal to n of 0.7.

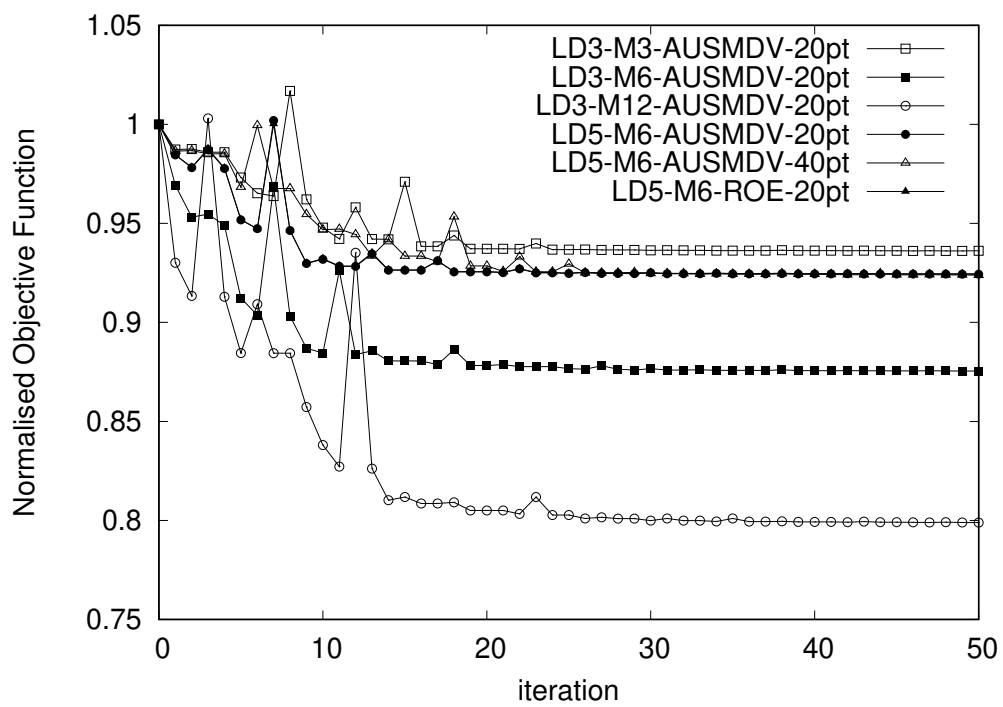
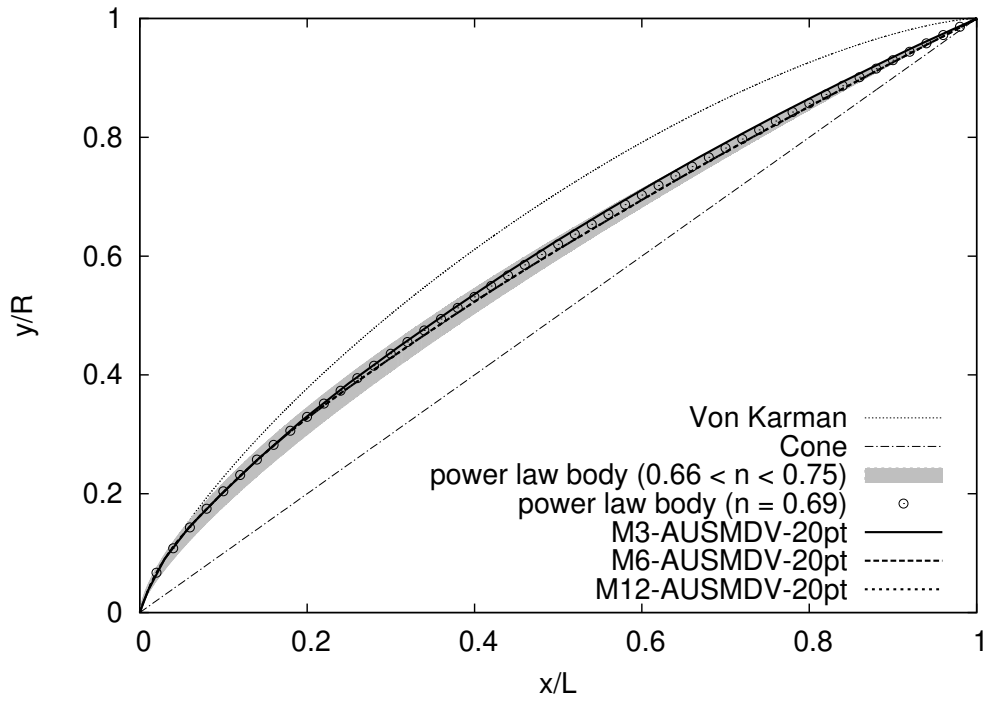
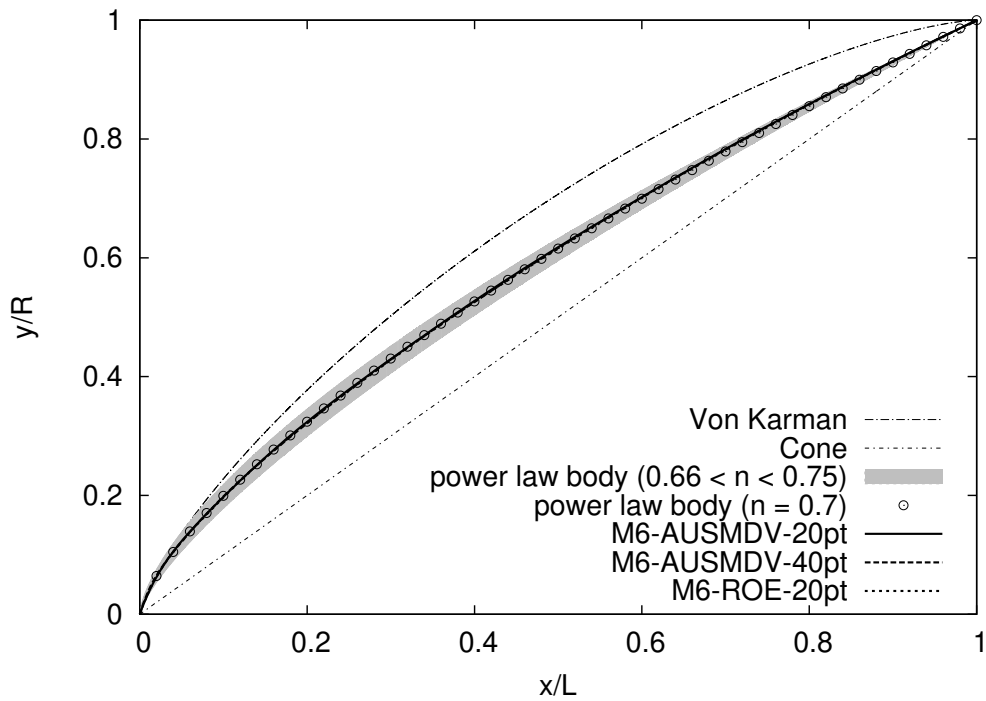


Figure 8.4: Objective function history.



(a) $L/D = 3$



(b) $L/D = 5$

Figure 8.5: Minimum-drag bodies.

8.3 Redesign of the NASA P2 hypersonic inlet

The redesign of the NASA P2 hypersonic inlet is the subject of the second application case. The following has been reproduced from Damm et al. [1].

8.3.1 Background for hypersonic inlet design

A key challenge to designing a hypersonic accelerating scramjet engine is achieving robust combustion of the fuel-air mixture over a wide range of Mach numbers [231]. The design of the hypersonic inlet, that is required to provide desired flow conditions to the combustor, is consequently of considerable importance to achieving realisable scramjet-powered accelerators.

A common hypersonic inlet design methodology is to integrate an inviscid flow analysis with boundary layer correction techniques to achieve the desired flow field compression. The P2 and P8 hypersonic inlets [232], developed in the 1970s at NASA, are an example of this procedure. The Rectangular-to-Elliptical-Shape-Transition (REST) inlet [233] is another, more modern example. Both of these classes of inlet have been known to suffer from adverse turbulent boundary layer interactions, not evident during the design process [232, 234].

Since the mid-1990s, several researchers have studied integrating high-fidelity Reynolds-Averaged Navier-Stokes (RANS) calculations into the hypersonic inlet design process. Gelsey et al. [235] and Shukla et al. [236, 237] presented some of the first RANS-based design optimisation results for hypersonic inlets: throughout a series of papers, the authors presented redesigns of the P2 and P8 hypersonic inlets using NASA's NPARC and GASP codes coupled with the k - ϵ turbulence model. Hasegawa et al. [238, 239] also utilised the GASP code in an automated design optimisation of generic 2D inlets using the k - ω turbulence model. More recently, Drayna et al. [240] developed a sensitivity analysis code within the US3D solver and applied it to the design optimisation of 3D hypersonic inward-turning inlets using the Spalart-Allmaras turbulence model. These works all suffered from poor scalability for a large number of design parameters, in that for each additional design variable, at least one additional objective function evaluation was required. This poor scaling of the design iteration is too restrictive for practical hypersonic inlet design, considering that a 3D hypersonic inlet requires 50+ design parameters to capture the geometric detail [20].

8.3.2 P2 hypersonic inlet

We have selected to redesign the P2 hypersonic inlet to demonstrate the applicability of the discrete adjoint method to hypersonic inlet design. This problem has been previously used to demonstrate the use of black-box finite difference style CFD optimisation for hypersonic inlet design [235–237], and, consequently, the inlet design community has some familiarity with the geometry. The P2 inlet (illustrated in Figure. 8.6) was a two-dimensional, planar, hypersonic inlet designed for a generic

accelerating vehicle at NASA in the early 1970s [232]. The original design objective was to achieve a desired compression ratio of 2 across the cowl shock while providing an approximately uniform static pressure distribution at the throat. From the experimental campaign, it was observed that the inlet did not achieve a uniform static pressure distribution across the throat. Instead, the cowl shock reflected off the bodyside of the inlet rather than being cancelled by an expansion. Gnos et al. [232] attributed the failure of the design approach to the analytical technique's poor capturing of the various interactions between the viscous and inviscid components of the flow, and in particular, the turbulent shockwave boundary layer interaction. Our goal, in this work, is to show how discrete adjoint optimisation can be used to develop a correction to the inlet shape that delivers on the original design intent.

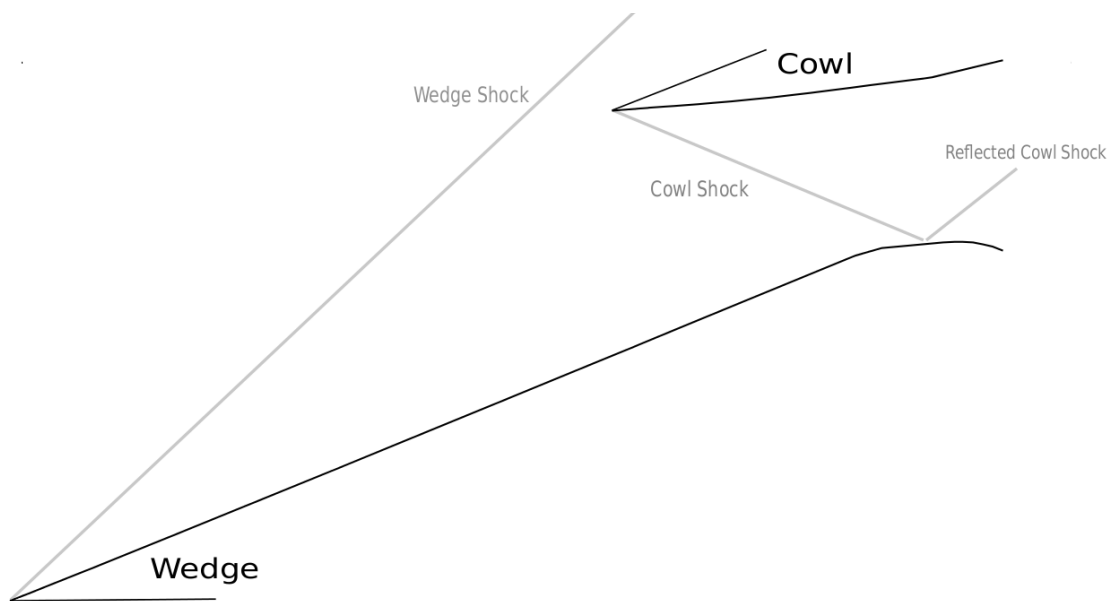


Figure 8.6: Schematic of the P2 Hypersonic Inlet.

8.3.3 Baseline flowfield

The flow solver has been validated for use in this inlet design context by comparison to the original experimental results of the P2 inlet presented in [232]. The sharp cowl approximation presented by Knight [241] is applied in this work to reduce the computational complexity of resolving the blunt leading edge. The P2 inlet walls were modelled as no-slip boundaries with a fixed temperature of 304 K, consistent with what was reported by the experimenters [232]. The inflow conditions were computed from the reported stagnation conditions, assuming $\gamma = 1.4$ for ideal air, and are presented in Table 8.3. No experimental data quantifying the turbulence intensity of the freestream is available for this inlet. Consequently, several combinations of freestream turbulence intensity and eddy viscosity ratio were applied, and are presented in Table 8.4. Transition was accommodated by separating the

domains into laminar regions and turbulent regions. The turbulent region was set at a distance of 40% of the fore-body wedge length. This distance matched that observed in experiment. First-order extrapolation of the flow state is applied at outflow boundaries. A grid independence study was undertaken, consisting of three levels of refinement, tabulated in Table 8.5. The G2 grid is presented in Figure. 8.7. Each mesh achieves a y^+ of less than 1 along the inlet bodyside wall, except G1 which has a maximum y^+ of 1.2 near the reflected shock region. The simulations were considered converged once the global L_2 error norm was reduced by at least 8 orders of magnitude. Mesh sequencing was utilised to accelerate convergence, with the current grid level being initialised with the previous grid levels converged solution. The G1 grid was initialised with the freestream condition and a fake boundary layer, generated by a linear blending of the wall properties and freestream conditions over 25 cells normal to the wall.

Table 8.3: Freestream inflow conditions for P2 inlet.

Mach number	7.4
Velocity, $\text{m}\cdot\text{s}^{-1}$	1221.8
Pressure, Pa	701.4
Temperature, K	67.85

Table 8.4: Testing of freestream turbulence properties for P2 inlet.

Simulation	I	$\frac{\mu_{\text{lam}}}{\mu_{\text{turb}}}$
S1	0.03	10
S2	0.03	100
S3	0.04	10
S4	0.05	10

Table 8.5: Levels of grid refinement used for validation.

Grid	No. Cells
G1	16240
G2	66125
G3	266845

Figure. 8.8 presents pressure contours for the converged solution on grid G3 for simulation S3. The strong reflected cowl shock observed in the experimental campaign is also present in the RANS calculation. Annotated on Figure. 8.8 are three stations at which pressure probes were placed during the experiment. The Pitot pressure at each station for simulation S3 is compared with the experimental data from [232] in Figures. 8.9(a), 8.9(b), and 8.9(c). Similar grid convergence trends were observed for simulations S1, S2 and S4; however, for brevity, the results are omitted. The results for simulations

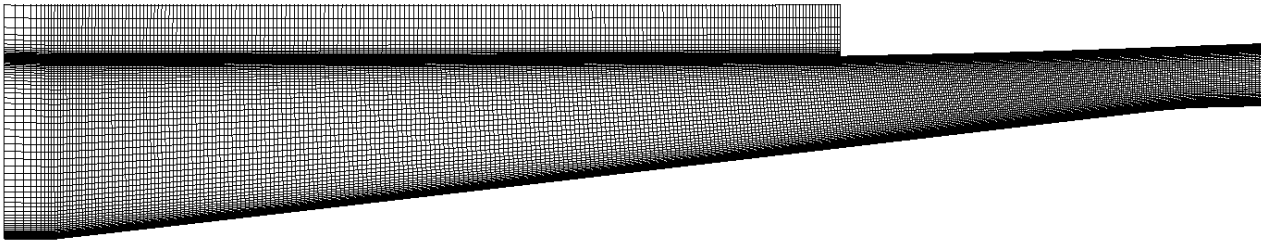


Figure 8.7: P2 inlet grid level G2.

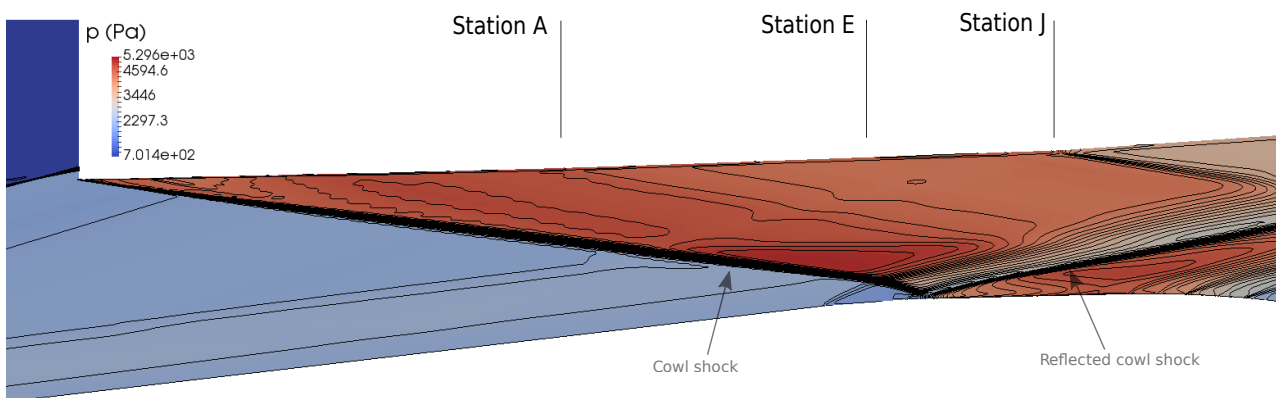


Figure 8.8: Baseline P2 inlet pressure colour map and contour lines for simulation S3 on grid level G3, with annotated experimental measurement locations.

S1 through S4, on grid G3, are presented in Figures. 8.10(a), 8.10(b), and 8.10(c). For all simulations, the largest discrepancy is noticeable near the cowl surface, where the sharp cowl approximation has not captured the entropy layer generated by the blunt leading edge. It is evident that the core-flow of the inlet is sensitive to the inflow freestream turbulence properties. Along the bodyside of the inlet, our simulations are in good agreement with the experimental data in the boundary layer region, however, an over-prediction in the core-flow region is observed. The numerical results compare well to others who have applied the sharp cowl approximation, however, prior works did not notice an over-prediction of the normalised Pitot pressures in the core flow [235–237, 241]. These authors used a patched-grid approach by dividing the inlet into self-contained regions, in comparison to simulating the inlet in a tip-to-tail fashion, as done in this current work. An over prediction of core-flow Pitot pressure has been reported for tip-to-tail simulations of the P8 hypersonic inlet by several authors [242], tested during the same experimental study [232]. Despite this sensitivity, the major flow features are captured by the flow solver, and hence the simulations will be suitable for demonstrating adjoint-based inlet design in the present work.

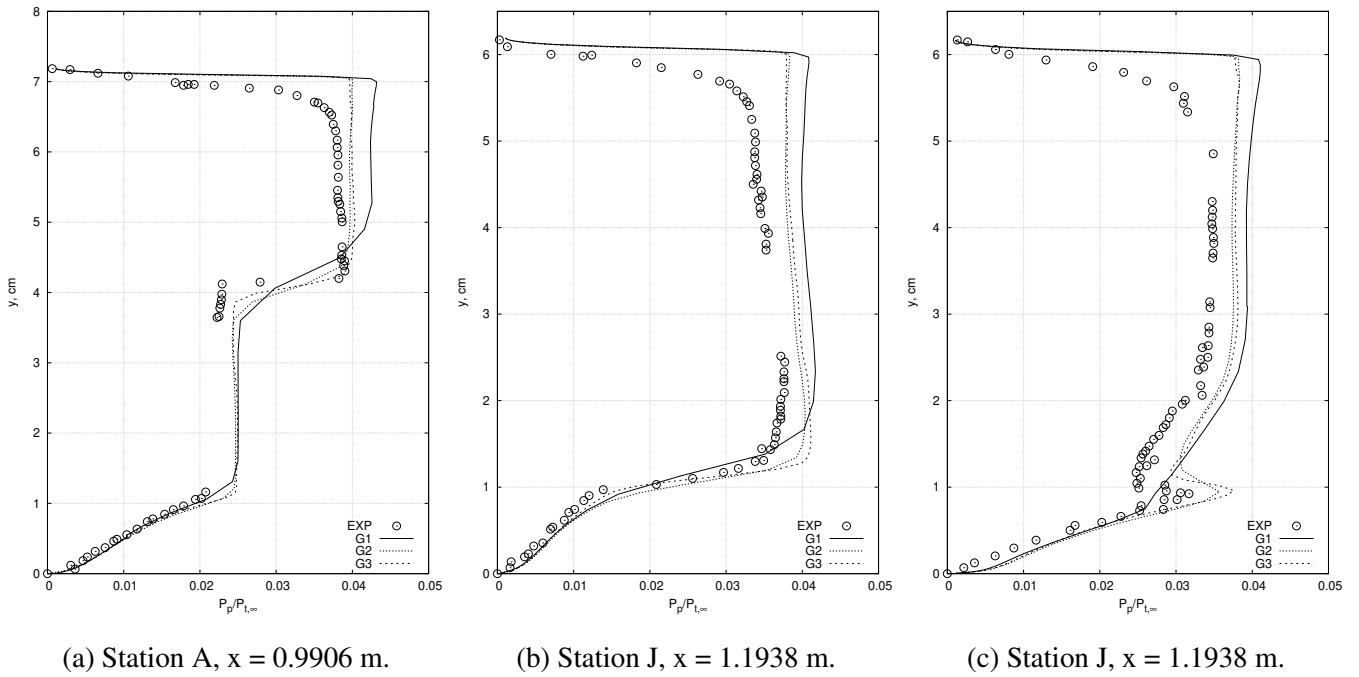


Figure 8.9: Simulation S3 Pitot pressure (P_p), normalised by $P_{t,\infty} = 4.14$ MPa. Experimental data from [232].

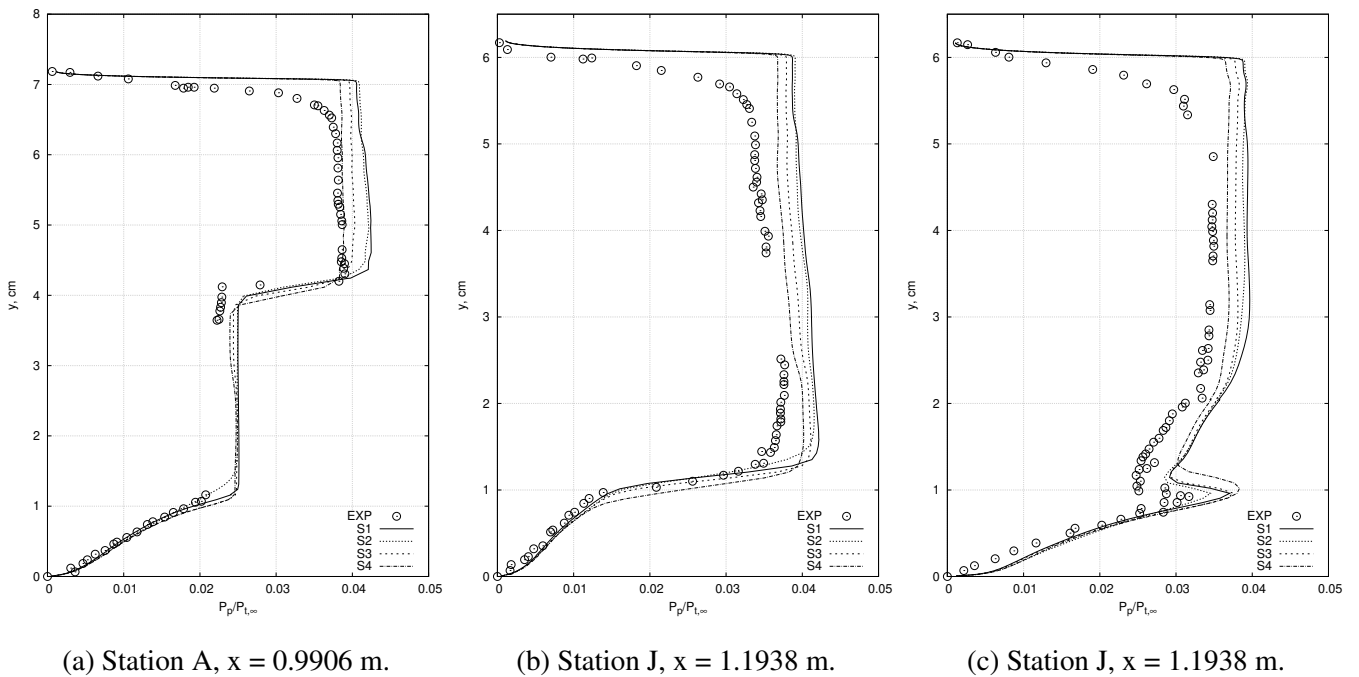


Figure 8.10: Simulation S3 on level G3 Pitot pressure (P_p), normalised by $P_{t,\infty} = 4.14$ MPa. Experimental data from [232].

8.3.4 Optimisation problem description

The original design objective for the P2 hypersonic inlet was to achieve a compression ratio of 2 across the cowl shock (corresponding to a compression ratio of 6.25 relative to the freestream), while providing a uniform pressure distribution across the throat of the inlet. Our chosen objective function for the redesign of the inlet is presented as Eq. 8.3.

$$J = \frac{\int_0^H (p(x,y) - p^*)^2 dA}{A_{\text{tot}}}, \quad (8.3)$$

Here, $p^* = r_p p_\infty$ is the desired pressure at the throat, and r_p is the desired compression ratio referenced to the freestream pressure. For our design optimisation we set $r_p = 6.25$ and $p_\infty = 701.4$ Pa, to be consistent with the original design objective documented by Gnos et al. [232]. The objective function (Eq. 8.3) was evaluated at the baseline inlet throat location, shown in Figure 8.11. Only a portion of the compression surface was chosen for redesign. The design surface was designated as the segment of the bodyside of the inlet spanning from the end of the forebody wedge, approximately 1.1 m downstream of the leading edge, to the end of the computational domain. The forebody wedge and cowl side geometry act to provide the core-flow compression, and since the goal is to remove the reflected shock, they remain fixed during the optimisation. The design surface segment of inlet was parameterised by a Bézier curve. The initial Bézier control points are fitted to the baseline P2 inlet geometry. To ensure smooth continuity between the fixed inlet surfaces and the design surface, the first, second and last Bézier control points have fixed y-ordinates. The second Bézier point is chosen to match the slope of the wedge. The remaining control points have a variable y-ordinate. Further details of the parameterisation can be found in the Appendix E. The turbulence properties were chosen to be those of simulation S3, and the optimisation was performed on grid level G2. The optimiser was run for a fixed number of design iterations.

Adjoint optimisation results for two Bézier curve parameterizations, an 11-point curve and a 20-point curve, are presented here. After 20 objective function evaluations and 4 adjoint solves, the objective function has already been reduced by more than 95% of the original value for both cases. The entire optimiser search history is shown in Figure 8.12. Also shown in Figure 8.12 is the same design problem optimised using standard finite-difference gradients with a step-size of 1×10^{-3} . Some observations can be made regarding the efficiency of the adjoint approach from the comparison of these results. The number of objective function evaluations required to sufficiently reduce the objective function scales excellently with the number of design variables for the adjoint approach: little difference is observed in the history of objective function evaluations when comparing 11-point and 20-point adjoint optimisations. On the other hand, the finite-difference approach is shown to scale very poorly. Given that, for this particular problem, an adjoint solve costs roughly the same as a flow solve, this demonstrates that the efficiency of the adjoint approach over the finite-difference approach

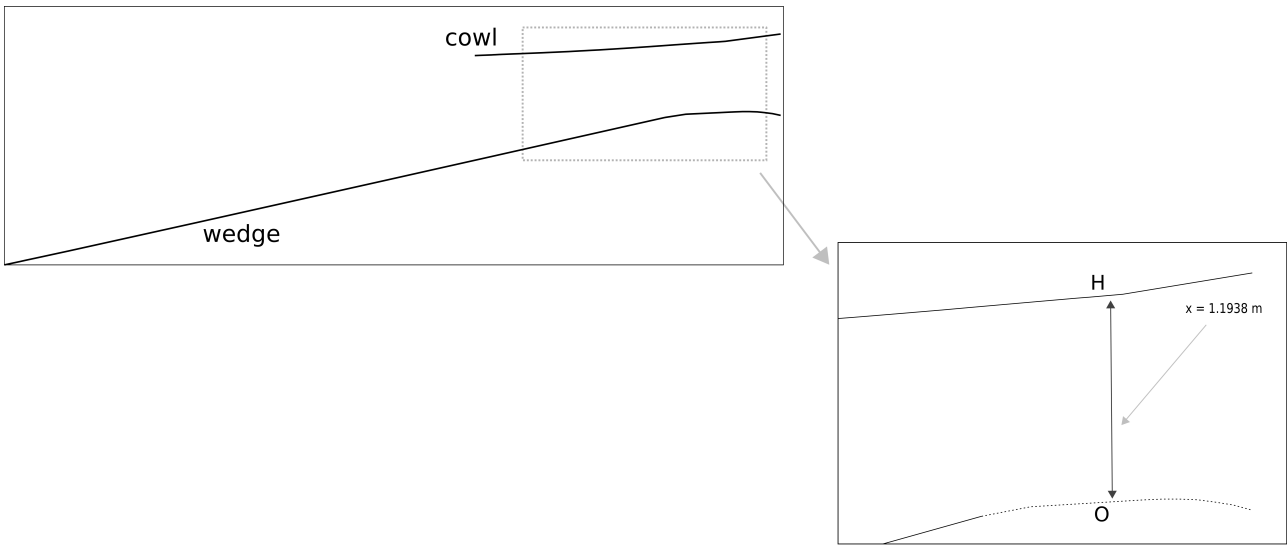


Figure 8.11: P2 inlet objective function integral path (design surface shown as dashed line).

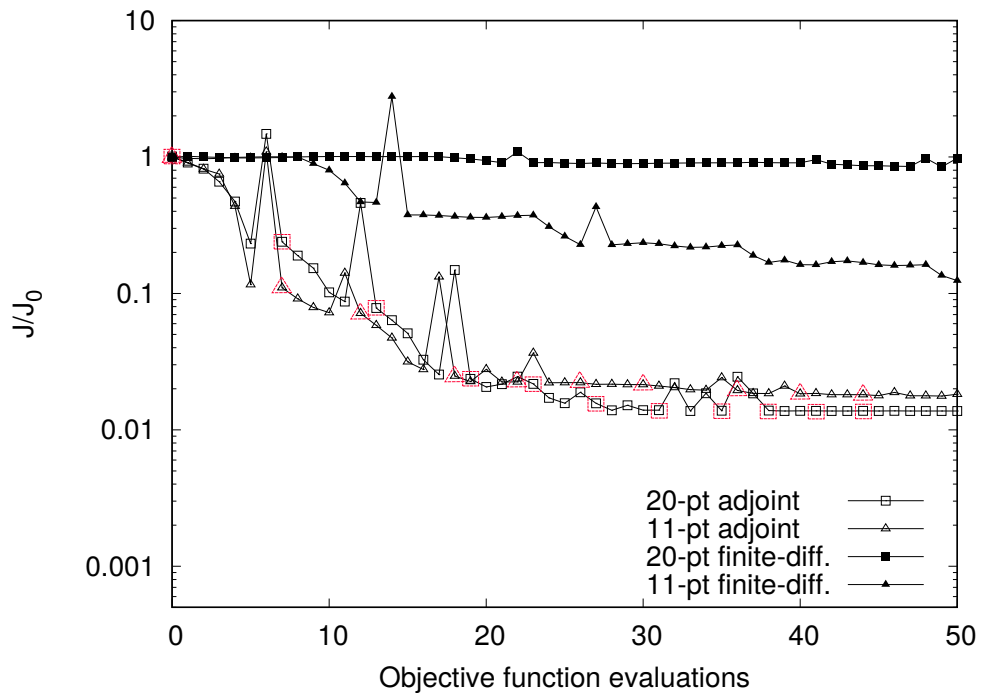


Figure 8.12: Objective function history for optimisation of the P2 inlet. Two design parameterisations are shown: a Bézier curve with 11 control points and a Bézier curve with 20 control points.

is substantial. Furthermore, for the adjoint optimisations, the wall-clock time taken to reach the fixed number of design iterations for the two parameterizations was within 2% of each other, where the 11-point optimisation (*not* the 20-point optimisation) incurred the longer wall-clock time, meaning the difference was not a direct result of the additional design parameters. So, an approximate doubling in the number of design variables has resulted in no additional compute time. This clearly demonstrates

the advantage of the adjoint approach over black-box style gradient-based methods.

8.3.5 Results

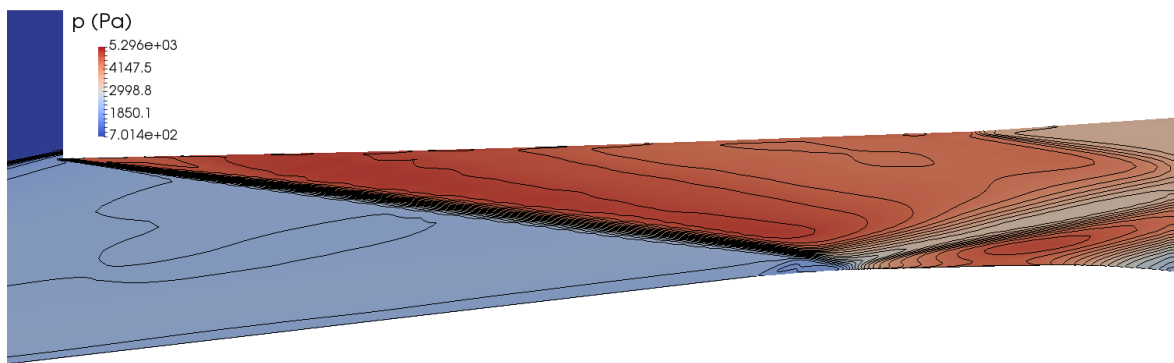
Figure 8.13 presents a pressure colour map with contour lines overlaid for the baseline and the 20-point Bézier curve optimised inlet. It is observed that the reflected shock has been virtually removed, without any substantial effects to the upstream flow features of the inlet. Similar pressure fields were obtained for the 11-point Bézier curve optimised inlet, but the plots have been omitted for brevity.

Plots of the Pitot pressure and static pressure normalised by their freestream values at the baseline throat location (J station) are presented in Figures. 8.14 and 8.15. It is observed that the characteristic pressure drop, from the shock-boundary layer interaction, has almost been completely removed for all cases, confirming the removal of the reflected shock. Furthermore, it is evident that similar profiles have been obtained for both the 11-point and 20-point parameterisation. A comparison of the two optimised Bézier curves are presented in Figure 8.16. It is observed that both curves share similar profiles up until approximately $x=1.1938$ m. After this location, the two curves deviate significantly from each other. This result is somewhat explicable since the flow is supersonic in the inlet, perturbations to the geometry downstream of the objective function evaluation location should have no influence on the value of the objective function, and consequently, have negligible effects on the design. This can be confirmed by examining the adjoint variables, which provide the sensitivity to the objective function with respect to conserved residuals in each cell. Figure 8.17 presents a colour map of the adjoint variable related to the conserved mass residual. It is observed that downstream of the objective function evaluation location, the adjoint variable has a magnitude of zero. This insensitivity to the objective function consequently means that the geometry here may be perturbed without any substantial effect on the objective function, noting that a perturbation in the design variables downstream could potentially result in a non-zero $\frac{\partial J}{\partial D}$ term in Eq. 6.5.

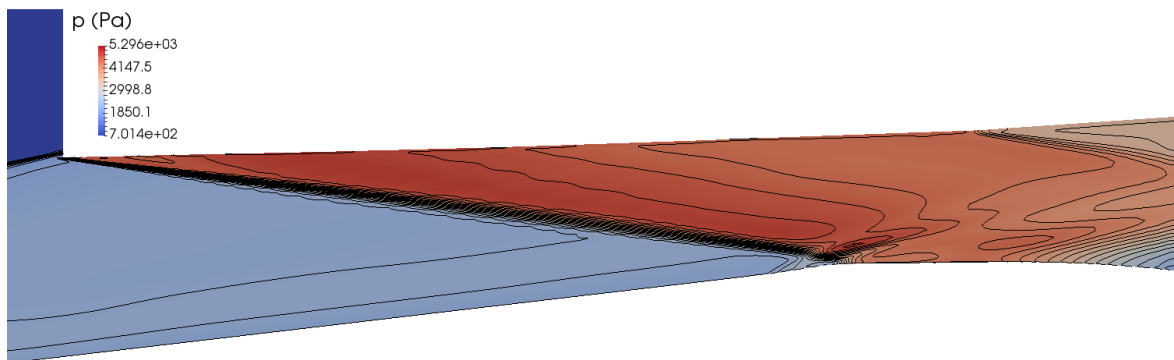
Table 8.6 lists several performance metrics for the baseline and optimised inlets. The adiabatic energy efficiency is computed from its definition:

$$\eta_{KE,ad} = \frac{h_{t_0} - h'_2}{h_{t_0} - h_0}, \quad (8.4)$$

where h'_2 is the enthalpy the flow would have if the gas state at the throat were expanded isentropically to the freestream pressure. The throat location has moved upstream by approximately 4% for both designs. Also, the throat width has been reduced by 4%. The total pressure recovery has improved for the optimised inlets, at no cost to the adiabatic kinetic energy efficiency, which has remained constant despite the geometric perturbations, a common trait of $\eta_{KE,ad}$ as discussed in the textbook by Heiser and Pratt (pg. 208) [32]. Note that the performance metrics in Table 8.6 were not evaluated at the objective function location, which was set to the baseline throat location. Instead, the performance



(a)



(b)

Figure 8.13: P2 inlet pressure colour map with contour lines overlaid for (a) Baseline (b) Optimised (20-point Bézier curve).

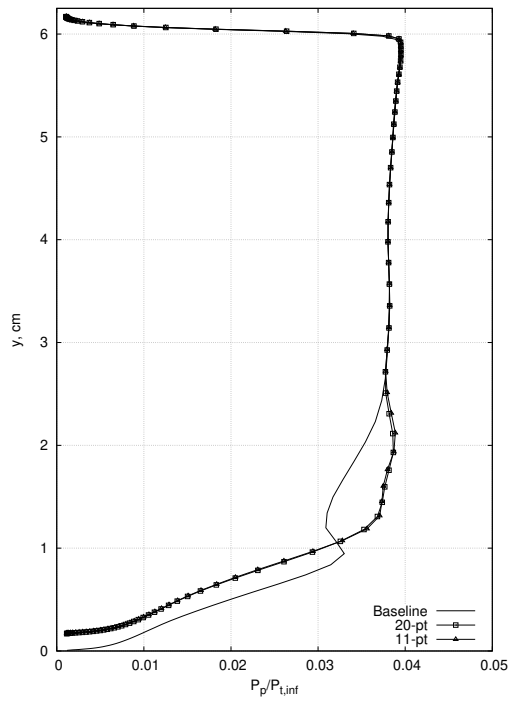


Figure 8.14: Comparison of Pitot pressure (P_p) at station J, $x = 1.1938$ m, normalised by $P_{t,\infty} = 4.14$ MPa for optimised inlets.

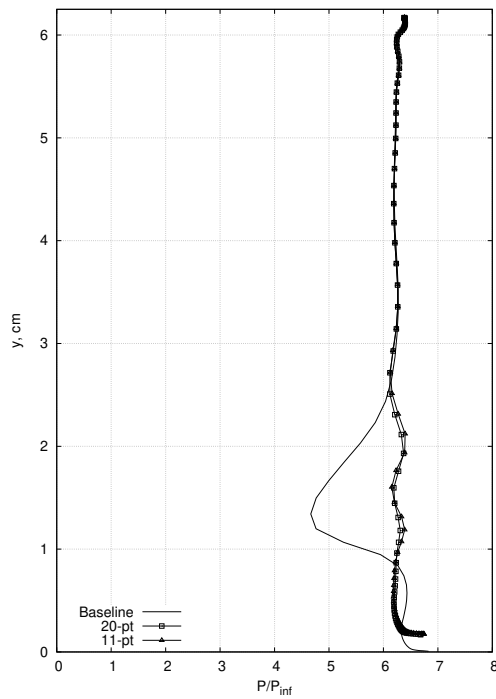


Figure 8.15: Comparison of pressure (p) at station J, $x = 1.1938$ m, normalised by $p_\infty = 701.4$ Pa for optimised inlets.

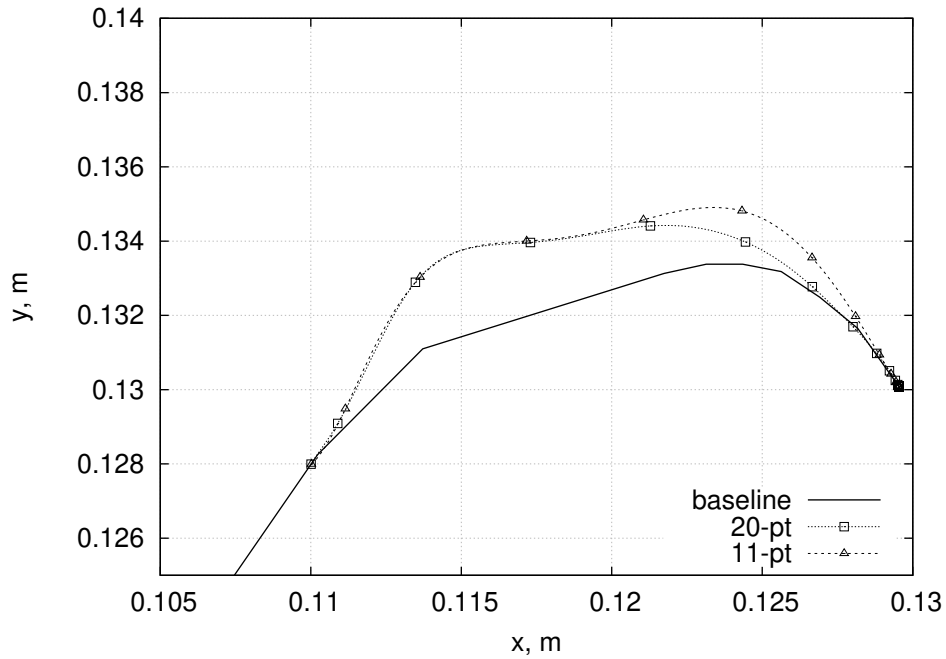


Figure 8.16: Comparison of baseline P2 inlet and optimised Bézier curves.

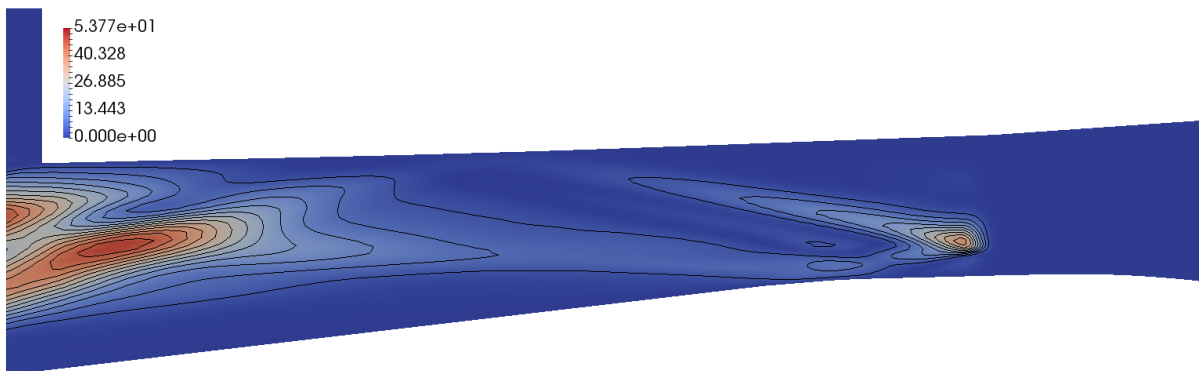


Figure 8.17: Colour map with contour lines of the absolute value of the adjoint variable related to Eq. 8.3 and the conserved mass residual for baseline P2 inlet.

metrics are evaluated at the throat location in the optimised inlet since it seems fairest to compare performance of inlets at their respective throat locations. However, in the interests of a direct comparison at the same physical location, the performance metrics for both redesigns at the baseline throat location were approximately $p/p_\infty = 6.22$, $p_t/p_{t,\infty} = 0.578$, $\eta_{KE,ad} = 0.985$. Summarising the results, the new inlets have achieved the desired compression ratio and removed the reflected shock, while retaining or improving the performance metrics.

Table 8.6: Performance metrics at respective throat locations extracted as flux-conserved quantities.

Performance metric	Baseline	20-point	11-point
Throat location (x_{throat} , m)	1.1938	1.1447	1.1459
Throat width (Δ_{OH} , mm)	61.18	58.86	58.80
Compression ratio (p/p_{∞})	5.987	6.367	6.366
Total pressure recovery ($p_t/p_{t,\infty}$)	0.576	0.584	0.584
Adiabatic kinetic energy efficiency ($\eta_{\text{KE,ad}}$)	0.985	0.985	0.985

8.4 Summary

This chapter presented two applications of adjoint-based optimisation in hypersonic flow. The two applications were: (a) wave drag minimisation of axisymmetric bodies; and (b) hypersonic inlet design. These examples span both external and internal high-speed flows, with both inviscid and viscous flow models. The success of these applications, demonstrated by the presented design results, show that adjoint-based optimisation is applicable for both external and internal hypersonic flow design analysis. The efficiency of the adjoint method was confirmed for hypersonic flows, with additional design parameters incurring no additional compute time per design iteration, as reported for the hypersonic inlet design case. Finally, by employing complex-variable differentiation to construct the adjoint operator and partial derivatives, no additional code extensions were required to accommodate different: (a) physics models (e.g. Euler and RANS); or (b) numerical methods (e.g. flux calculators). This is an encouraging result, given the development of numerical methods for the simulation of hypersonic compressible flows is still an active field of research.

9.1 Summary and discussion

The overall aim of this thesis was to

Investigate how high-fidelity CFD-based optimisation using the adjoint method can be applied to aerodynamic shape design in hypersonic flow.

This was divided into several objectives that were listed in Chapter 1 and are repeated here:

1. Implement a Newton-Krylov accelerated, unstructured grid, compressible flow solver.
2. Implement a discrete adjoint solver based on the developed flow solver.
3. Develop an optimisation package for aerodynamic shape design.
4. Apply the developed optimisation package to aerodynamic shape design in hypersonic flow.

With these objectives in mind, we may now summarise how the presented work addressed these objectives.

Chapter 2 provided a general background to hypersonic vehicle design. The chapter began with an introduction to airbreathing hypersonic vehicles. The discussion focused on the importance of airframe-engine integration in reducing overall drag and achieving positive thrust. Following this, the relevant flow phenomena for airbreathing hypersonic flight were presented. The key messages from these two sections were that: (a) hypersonic vehicles have complex geometries; and (b) the flow physics that describe a hypersonic vehicle's aerodynamics are complex. The chapter concluded with a critical review of current design methodologies. The findings suggested that an improved design system that employs numerical optimisation, which can account for and take advantage of the highly

integrated nature of hypersonic vehicles, is required for robust designs to be developed. The adjoint method was introduced at the end of Chapter 2 as a proposed means of efficiently accommodating the complex geometries and flow physics relevant to hypersonic vehicle design into an optimisation framework.

A review of adjoint-based optimisation was then presented in Chapter 3. The discrete adjoint method was derived and the key challenge of constructing the adjoint operator was discussed. Several popular techniques for constructing the adjoint operator were compared. Ultimately the complex-step derivative approach was chosen as the preferred method for this work as a consequence of its accuracy and ease of implementation. Chapter 3 concluded with a review of adjoint-based optimisation in hypersonic flow. The review uncovered that the application of the discrete adjoint method to RANS-based design in high-speed flows had not been extensively reported in the literature. This motivated the contributions made by this work, outlined below.

Chapter 4 introduced the flow solver developed for high-speed flow analysis. To provide flow analysis with a rapid turn-around, an unstructured steady-state RANS solver driven by a Jacobian-Free Newton-Krylov method was developed. In this work, turbulence was modelled using the two-equation $k - \omega$ turbulence model. The Newton method was globalised using the pseudo-transient approach. A restarted GMRES method was used to solve the system of linear equations arising when solving for the Newton steps. Evaluation of the matrix-vector products required in the GMRES algorithm was accomplished by Fréchet derivatives using imaginary perturbations in the complex plane. This was necessary to achieve robust convergence of the types of turbulent hypersonic flow considered in this work. Equation scaling ensured that the linear solver provides an adequate solution of the linear system, especially for turbulent flows, where the flow and turbulence variables differ by several orders of magnitude. Incomplete lower-upper preconditioning with zero-fill was used to accelerate linear system convergence. To achieve adequate residual convergence for flows with embedded shocks, limiter freezing was required to prevent early stall.

The results from a complete verification and validation of the developed flow solver were presented in Chapter 5. The Method of Manufactured Solutions was employed to quantitatively verify the spatial order of accuracy of the code. As expected, the order of accuracy was demonstrated to be 2 for both the two- and three-dimensional solvers. The flow solver's shock capturing and boundary layer capturing capabilities were verified by qualitative means. The shock capturing was verified by comparison to analytic solutions of an oblique shock wave. The boundary layer capturing capability was verified by comparison to numerical solutions of the boundary layer equations in self-similarity variables. The flow solver was validated using several validation cases taken from the literature to establish the appropriateness of the implemented physical models for design analysis in high-speed flow. The selected validation cases covered the spectrum of relevant flow phenomena, in particular, high-speed flows with strong inviscid/viscous interactions. Good agreement to the published experimental data for all cases was observed.

In Chapter 6, the accompanying adjoint solver developed for this work was presented. The primary complication of linearising the flow solver routines was handled via a complex-step derivative approach. Targeted differentiation was employed to provide an efficient means of constructing the adjoint operator. The evaluation of the remaining partial derivatives required in the total derivative calculation was also via a complex-step derivative approach. The adjoint gradients were verified against a complex variable direct-differentiation method. The gradients are shown to have excellent agreement. The chapter concluded by presenting a useful technique for debugging the adjoint operator.

Chapter 7 presented the remaining components of the developed optimisation package. Namely, the Bézier curve surface parameterisation and the inverse distance weighting (IDW) method for mesh deformation. The open-source optimisation library, DAKOTA, was utilised in this work for performing the design space search. Communication of the objective function evaluations and gradients between the developed software and DAKOTA was necessary. This coupling was outlined at the end of Chapter 7.

In Chapter 8, two optimisation applications were presented as a demonstration of the development work: (a) wave drag minimisation of an axisymmetric body; and (b) hypersonic inlet design optimisation. For the case of axisymmetric bodies, the optimal shape determined by this work compared favourably to several minimum-drag power law bodies published in the literature. The second application was the redesign of the P2 hypersonic inlet. The chosen objective function aimed at removing the reflected cowl shock whilst obtaining the desired compression ratio. The results presented show that the optimiser has removed the reflected shock while achieving the desired compression ratio, at no cost to the inlet performance metrics.

The conclusions that may be drawn from the work conducted in this thesis is that the discrete adjoint-based optimisation method does work well in a hypersonic flow context, and that use of the complex-step differentiation is a key enabler in the implementation. In particular, the inlet example presented in this work demonstrated the efficiency, accuracy, and applicability of discrete adjoint-based optimisation to design analysis in turbulent high-speed flow. Another conclusion from this work is that the complex Fréchet derivative works well for evaluating the matrix-vector products in a Jacobian-Free Newton-Krylov method in a hypersonic flow context. The advantage of setting the perturbation parameter independent of round-off error was instrumental in solving the large three-dimensional problems presented in Chapter 5. Moreover, the complex Fréchet derivative was an enabler for achieving sufficient levels of residual convergence for the turbulent high-speed flows simulated in this work.

9.2 Contributions

The primary contribution of this work is the description and application of a discrete adjoint solver in high-speed compressible flow optimisation. What is unique in this work is a demonstration that complex-step differentiation works well to linearise a second-order spatially accurate unstructured RANS solver. In particular, the approach presented in this work utilised the $k - \omega$ turbulence model in high-speed ducted flow configurations. We have not seen this reported elsewhere.

In addition to the adjoint development, this work contributes towards the advancement of Newton-Krylov methods for the primal solution of turbulent high-speed flows. In particular, it is believed that this is the first application of a Jacobian-Free Newton-Krylov method to solving the RANS equations coupled with the $k - \omega$ turbulence model on unstructured grids using a spatially second-order numerical discretisation. Furthermore, the application of a complex-step Fréchet derivative for evaluating the matrix-vector products is believed to be novel.

More local contributions are the new capabilities for the in-house flow solver, Eilmer, produced by this work. Historically, Eilmer has utilised explicit time-stepping to calculate the steady-state flows achieved in the experimental facilities at the University of Queensland. The addition of a steady-state solver to the software suite will provide a more efficient means of simulating flows for comparison to experimental data. Furthermore, it is envisioned that the optimisation package will also provide an efficient means of model design for future experimental campaigns.

9.3 Recommendations for future work

Over the course of this investigation, a number of research directions were identified which extended well beyond the scope of this work. First, some specific recommendations for directly extending this work are presented, and, second, more general recommendations for the adjoint-based optimisation community are discussed.

Extend surface parameterisation for 3D geometries: The Bézier curve surface parameterisation is difficult to extend to three-dimensional surfaces. To facilitate the design of three-dimensional geometries, the geometry parameterisation in Eilmer will need to be extended. Several candidate methods were highlighted in Chapter 7, the simplest extension to the Bézier curve routine already implemented are Bézier surfaces. However, free-form deformation has shown to be robust for complex geometries and is under consideration.

Perform optimisation of 3D geometries: The flow and adjoint solvers developed for this work operate on two- and three-dimensional meshes. However, due to the limitations of the surface parameterisation, the applications presented here were restricted to two-dimensional and axisymmetric geome-

tries. Future work will include the optimisation of three-dimensional geometries. Several candidate applications include modern hypersonic inlets, such as modular shape-transition inlets [243], and the SPARTAN vehicle airframe.

Implement robust three-dimensional mesh movement strategy: The IDW method produced deformed meshes of sufficient quality for the two-dimensional optimisation applications presented in this work. Although the IDW method applies to three-dimensional mesh deformation, several alternate candidate methods have been selected for future work, such as the application of linear elastic solvers.

Extend adjoint solver to include mesh adjoint: The IDW method is computationally inexpensive, and as such a mesh adjoint was not considered in this work. If a more computationally expensive mesh deformation method is implemented, then a mesh adjoint will be necessary to retain the scalability of the adjoint solver.

Extend flow and adjoint solvers to include chemical non-equilibrium: The flow and adjoint solvers in this work were developed for a single species gas. A natural extension to both these solvers is the inclusion of chemical non-equilibrium flows. The baseline solver, Eilmer, already includes routines for finite-rate chemistry, currently utilised by the explicit solver. Future work will extend the steady-state flow and adjoint solvers to include the use of finite-rate chemistry for modelling and design in non-equilibrium flows. This can then be applied to the propulsion flowpath design for scramjet engines.

More generally, a limitation of the adjoint method experienced in this work is achieving satisfactory convergence of the flow solver residual for flows with embedded shocks. Flows with strong shock waves present are known to suffer from stalled convergence due to the ringing of the limiter demonstrated in Chapter 4. Although freezing the limiter worked well for the applications presented in Chapter 8, this approach is not ideal, since it is known to be problem dependent and has been shown to exhibit small-scale oscillations in the flow solution for some limiters [105]. For these reasons, focus should be given to the development of innovative numerical methods that retain a high order of spatial accuracy whilst removing the need for flux or gradient limiting. Several techniques are currently in development [244–247]. However, the applicability of adjoint-based optimisation for these numerical methods will need to be investigated given the requirements for a convergent adjoint system with embedded shocks discussed in Section 3.5 [63–66]

Finally, this work was motivated by the promise that adjoint-based multidisciplinary design optimisation has for the design of hypersonic vehicles, such as the SPARTAN vehicle introduced in Chapter 1. As a first step, this work has investigated the application of a discrete adjoint-based opti-

misation framework for high-speed design of the vehicle aerodynamics. Towards the goal of adjoint-based MDO, the components of this framework can be used as one of several modules in an adjoint-based MDO framework. The coupling of several disciplines in an adjoint-based MDO framework is an important next step in this research field. The long term goal is using adjoint-based optimisation to perform high-fidelity design analysis of entire vehicle configurations.

REFERENCES

- [1] K.A. Damm, R.J. Gollan, P.A. Jacobs, M.K. Smart, S. Lee, E. Kim, and C. Kim. “Discrete adjoint optimization of a hypersonic inlet”. In: *AIAA Journal* (2019).
- [2] K.A. Damm, R.J. Gollan, and A. Veeraragavan. “Acceleration of combustion simulations using GPUs”. In: *Australian Combustion Symposium*. Melbourne, Australia, 2015, pp. 148–151.
- [3] K.A. Damm, R.J. Gollan, and A. Veeraragavan. “On the effect of workload ordering for reacting flow simulations using GPUs”. In: *20th Australasian Fluid Mechanics Conference*. Paper Number: 748. Perth, Australia, 2016, pp. 748.1–748.4.
- [4] K.A. Damm, R.J. Gollan, and P.A. Jacobs. “Verification of the least-squares procedure within an unstructured-grid flow solver”. In: *20th Australasian Fluid Mechanics Conference*. Paper Number: 703. Perth, Australia, 2016, pp. 703.1–703.4.
- [5] K.A. Damm, R.J. Gollan, P.A. Jacobs, S. Lee, E. Kim, and C. Kim. “Adjoint design optimization for the P2 hypersonic inlet”. In: *Korean Society of Computational Fluid Engineering Conference*. Jeju, South Korea, 2018.
- [6] K.A. Damm, R.J. Gollan, P.A. Jacobs, S. Lee, E. Kim, and C. Kim. “Adjoint design optimization for the P2 hypersonic inlet”. In: *10th National Congress on Fluids Engineering Conference*. Yeosu, South Korea, 2018.
- [7] K. A. Damm. *Shock Fitting mode for Eilmer*. Mechanical Engineering Report 2016/15. Centre for Hypersonics, University of Queensland, 2016.
- [8] Elizabeth Buchen. “2013 Small Satellite Market Observations”. In: *AIAA/USU Conference on Small Satellites*, Technical Session VII: Opportunities, Trends and Initiative. 2013.
- [9] Australian Government. *The high growth potential of Australia’s space economy*. ACT, 2013.
- [10] Dawid Preller and Michael K Smart. “Reusable launch of small satellites using scramjets”. In: *Journal of Spacecraft and Rockets* 54.6 (2017), pp. 1317–1329. DOI: 10.2514/1.A33610.

REFERENCES

- [11] Dawid Preller and Michael K. Smart. “Scramjets for Reusable Launch of Small Satellites”. In: *20th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*. AIAA paper 2015-3586. Reston, Virginia, 2015. DOI: 10.2514/6.2015-3586.
- [12] Michael K. Smart. “Scramjets”. In: *The Aeronautical Journal* 111.1124 (2007), pp. 605–619. DOI: 10.1017/S0001924000004796.
- [13] Michael K. Smart and Matthew R. Tetlow. “Orbital delivery of small payloads using hypersonic airbreathing propulsion”. In: *Journal of Spacecraft and Rockets* 46.1 (2009), pp. 117–125. DOI: 10.2514/1.38784.
- [14] Kevin G. Bowcutt. “A perspective on the future of aerospace vehicle design”. In: *12th AIAA International Space Planes and Hypersonic Systems and Technologies*. AIAA paper 2003-6957. Norfolk, Virginia, 2003. DOI: 10.2514/6.2003-6957.
- [15] Kevin G. Bowcutt. “Multidisciplinary optimization of airbreathing hypersonic vehicles”. In: *Journal of Propulsion and Power* 17.6 (2001), pp. 1184–1190. DOI: 10.2514/2.5893.
- [16] Thomas Jazra, Dawid Preller, and Michael K. Smart. “Design of an airbreathing second stage for a rocket-scramjet-rocket launch vehicle”. In: *Journal of Spacecraft and Rockets* 50.2 (2013), pp. 411–422. DOI: 10.2514/1.A32381.
- [17] Thomas Jazra. “Optimisation of Hypersonic Vehicles for Airbreathing Propulsion”. PhD thesis. University of Queensland, 2010.
- [18] Dawid Preller. “Multidisciplinary design and optimisation of a pitch trimmed hypersonic airbreathing accelerating vehicle”. PhD thesis. University of Queensland, 2018.
- [19] Kevin G. Bowcutt, Geojoe Kuruvila, Thomas A. Grandine, Thomas A. Hogan, and Evin J. Cramer. “Advancements in multidisciplinary design optimization applied to hypersonic vehicles to achieve closure”. In: *15th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*. AIAA paper 2008-2591. Dayton, Ohio, 2008. DOI: 10.2514/6.2008-2591.
- [20] Paul G. Ferlemann and Rowan J. Gollan. “Parametric geometry, structured grid generation, and initial design study for REST-Class hypersonic inlets”. In: *JANNAF 43rd Combustion & 31st Airbreathing Joint Meeting*. Salt Lake City, Utah, 2010.
- [21] Olivier Pironneau. “On optimum design in fluid mechanics”. In: *Journal of Fluid Mechanics* 64.1 (1974), pp. 97–110. DOI: 10.1017/S0022112074002023.
- [22] Dominique Thévenin and Gábor Janiga. *Optimization and Computational Fluid Dynamics*. New York: Springer Science & Business Media, 2008.

- [23] Byung Joon Lee and Chongam Kim. “Aerodynamic redesign using discrete adjoint approach on overset mesh system”. In: *Journal of Aircraft* 45.5 (2008), pp. 1643–1653. DOI: 10.2514/1.34112.
- [24] Eric J. Nielsen. “Adjoint-Based Aerodynamic Design of Complex Aerospace Configurations”. In: *ASME 2016 Fluids Engineering Division Summer Meeting collocated with the ASME 2016 Heat Transfer Summer Conference and the ASME 2016 14th International Conference on Nanochannels, Microchannels, and Minichannels*. American Society of Mechanical Engineers. Washington, DC, 2016. DOI: 10.1115/FEDSM2016-7573.
- [25] Joaquim R. R.A. Martins, Juan J. Alonso, and James J. Reuther. “High-fidelity aerostructural design optimization of a supersonic business jet”. In: *Journal of Aircraft* 41.3 (2004), pp. 523–530. DOI: 10.2514/1.11478.
- [26] Sriram K. Rallabhandi, Eric J. Nielsen, and Boris Diskin. “Sonic-boom mitigation through aircraft design and adjoint methodology”. In: *Journal of Aircraft* 51.2 (2014), pp. 502–510. DOI: 10.2514/1.C032189.
- [27] Antony Jameson. “Re-engineering the design process through computation”. In: *Journal of Aircraft* 36.1 (1999), pp. 36–50. DOI: 10.2514/2.2412.
- [28] Antony Jameson and John C. Vassberg. “Computational fluid dynamics for aerodynamic design: Its current and future impact”. In: *39th AIAA Aerospace Sciences Meeting*. Vol. 538. AIAA Paper 2001-0538. Reno, Nevada, 2001. DOI: 10.2514/6.2001-538.
- [29] Joseph Shea. *Report of the Defense Science Board Task Force on the National Aerospace Plane (NASP)*. Report AD-A201 124. Defense Science Board, Office of the Under Secretary of Defense for Acquisition, 1988.
- [30] Ajay Kumar, J. Philip Drummond, Charles R. McClinton, and James L. Hunt. “Research in hypersonic airbreathing propulsion at the NASA Langley Research Center”. In: *Fifteenth International Symposium on Airbreathing Engines*. Bangalore, India, 2001.
- [31] Robert A. Jones and Paul W. Huber. “Airframe-integrated propulsion system for hypersonic cruise vehicles”. In: *11th Congress of the International Council of the Aeronautical Science*. Lisbon, Portugal, 1978.
- [32] William H. Heiser and David T. Pratt. *Hypersonic Airbreathing Propulsion*. Reston, Virginia: AIAA, 1994.
- [33] Sholto O. Forbes-Spyratos, Michael P. Kearney, Michael K. Smart, and Ingo H. Jahn. “Trajectory Design of a Rocket–Scramjet–Rocket Multistage Launch System”. In: *Journal of Spacecraft and Rockets* 56.1 (2018), pp. 53–67. DOI: 10.2514/1.A34107.

REFERENCES

- [34] Thomas A. McLaughlin. “Viscous optimized hypersonic waveriders for chemical equilibrium flow”. PhD thesis. University of Maryland, 1990.
- [35] John David Anderson Jr., Jinhwa Chang, and T.A. McLaughlin. “Hypersonic waveriders: effects of chemically reacting flow and viscous interaction”. In: *30th Aerospace Sciences Meeting and Exhibit*. AIAA paper 1992-0302. Reno, Nevada, 1992. DOI: 10.2514/6.1992-302.
- [36] Thomas R.A. Bussing and Scott Eberhardt. “Chemistry associated with hypersonic vehicles”. In: *Journal of Thermophysics and Heat Transfer* 3.3 (1989), pp. 245–253. DOI: 10.2514/3.28772.
- [37] John David Anderson Jr. *Hypersonic and High Temperature Gas Dynamics*. Reston, Virginia: AIAA, 2000.
- [38] Robert H. Korkegi. “Survey of viscous interactions associated with high Mach number flight”. In: *AIAA Journal* 9.5 (1971), pp. 771–784. DOI: 10.2514/3.6275.
- [39] Andy J. Keane and Prasanth B. Nair. *Computational Approaches for Aerospace Design*. Vol. 582. Wiley Online Library, 2005.
- [40] Ed Obert. *Aerodynamic Design of Transport Aircraft*. IOS press, 2009.
- [41] Jaroslaw Sobieszczanski-Sobieski. “Multidisciplinary design optimization: an emerging new engineering discipline”. In: *Advances in Structural Optimization*. Springer, 1995, pp. 483–496. DOI: 10.1007/978-94-011-0453-1_14.
- [42] Jing Che and Shuo Tang. “Research on integrated optimization design of hypersonic cruise vehicle”. In: *Aerospace Science and Technology* 12.7 (2008), pp. 567–572. DOI: 10.1016/j.ast.2008.01.008.
- [43] Kevin Bowcutt, Steve Hollowell, Mark Gonda, and Ted Ralston III. “Performance, operational and economic drivers of reusable launch vehicles”. In: *38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*. AIAA paper 2002-3901. Indianapolis, Indiana, 2002. DOI: 10.2514/6.2002-3901.
- [44] John A. Nelder and Roger Mead. “A simplex method for function minimization”. In: *The Computer Journal* 7.4 (1965), pp. 308–313. DOI: 10.1093/comjnl/7.4.308.
- [45] Atsushi Ueno, Hideyuki Taguchi, and Kojiro Suzuki. “Aerodynamic shape optimization of hypersonic airliners considering multi-design-point”. In: *27th Congress of International Council of the Aeronautical Sciences*. Vol. 2. 3. Nice, France, 2010, p. 2010.
- [46] Daisuke Sasaki, Shigeru Obayashi, and Hyoung-Jin Kim. “Evolutionary algorithm vs. adjoint method applied to SST shape optimization”. In: *The Annual Conference of CFD Society of Canada, Waterloo*. Waterloo, Canada, 2001.

- [47] T.H. Pulliam, Marian Nemec, Terry Holst, and D.W. Zingg. “Comparison of evolutionary (genetic) algorithm and adjoint methods for multi-objective viscous airfoil optimizations”. In: *AIAA Journal* 298.4 (2003). DOI: 10.2514/6.2003-298.
- [48] David W. Zingg, Marian Nemec, and Thomas H. Pulliam. “A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization”. In: *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique* 17.1-2 (2008), pp. 103–126. DOI: 10.3166/remn.17.103-126.
- [49] Shigeru Obayashi and Takanori Tsukahara. “Comparison of optimization algorithms for aerodynamic shape design”. In: *AIAA Journal* 35.8 (1997), pp. 1413–1415. DOI: 10.2514/2.251.
- [50] Antony Jameson. “Aerodynamic design via control theory”. In: *Journal of Scientific Computing* 3.3 (1988), pp. 233–260. DOI: 10.1007/BF01061285.
- [51] Eric J. Nielsen. “Aerodynamic design sensitivities on an unstructured mesh using the Navier-Stokes equations and a discrete adjoint formulation”. PhD thesis. Virginia Polytechnic University, 1998.
- [52] M. Giles, Devendra Ghate, and Mihai C. Duta. *Using automatic differentiation for adjoint CFD code development*. Oxford University Research Archive Report 05/25. Oxford University.
- [53] Jacques E.V. Peter and Richard P. Dwight. “Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches”. In: *Computers & Fluids* 39.3 (2010), pp. 373–391. DOI: 10.1016/j.compfluid.2009.09.013.
- [54] Eric J. Nielsen and Michael A. Park. “Using an adjoint approach to eliminate mesh sensitivities in computational design”. In: *AIAA Journal* 44.5 (2006), pp. 948–953. DOI: 10.2514/1.16052.
- [55] Joaquim Martins, Ilan Kroo, and Juan Alonso. “An automated method for sensitivity analysis using complex variables”. In: *38th Aerospace Sciences Meeting and Exhibit*. AIAA paper 2000-0689. Reno, Nevada, 2000, p. 689. DOI: 10.2514/6.2000-689.
- [56] Eric J. Nielsen and William L. Kleb. “Efficient construction of discrete adjoint operators on unstructured grids using complex variables”. In: *AIAA Journal* 44.4 (2006), pp. 827–836. DOI: 10.2514/1.15830.
- [57] W. Kyle Anderson, James C. Newman, David L. Whitfield, and Eric J. Nielsen. “Sensitivity analysis for Navier-Stokes equations on unstructured meshes using complex variables”. In: *AIAA Journal* 39.1 (2001), pp. 56–63. DOI: 10.2514/2.1270.

REFERENCES

- [58] Chad E. Burdysshaw and W. Kyle Anderson. “A general and extensible unstructured mesh adjoint method”. In: *Journal of Aerospace Computing, Information, and Communication* 2.10 (2005), pp. 401–413. DOI: 10.2514/1.15932.
- [59] James N Lyness and Cleve B Moler. “Numerical differentiation of analytic functions”. In: *SIAM Journal on Numerical Analysis* 4.2 (1967), pp. 202–210. DOI: 10.1137/0704019.
- [60] Joaquim R. Martins. “A coupled-adjoint method for high-fidelity aero-structural optimization”. PhD thesis. Stanford University, 2002.
- [61] J. R. R. A. Martins, Peter Sturdza, and Juan Alonso. “The connection between the complex-step derivative approximation and algorithmic differentiation”. In: *39th Aerospace Sciences Meeting and Exhibit*. AIAA paper 2001-0921. Reno, Nevada, 2001. DOI: 10.2514/6.2001-921.
- [62] Joaquim R.R.A. Martins and Andrew B. Lambe. “Multidisciplinary design optimization: a survey of architectures”. In: *AIAA Journal* 51.9 (2013), pp. 2049–2075. DOI: 10.2514/1.J051895.
- [63] Michael B. Giles. “Discrete adjoint approximations with shocks”. In: *Hyperbolic problems: theory, numerics, applications*. Springer, 2003, pp. 185–194. DOI: 10.1007/978-3-642-55711-8_16.
- [64] Mike Giles and Stefan Ulbrich. “Convergence of linearized and adjoint approximations for discontinuous solutions of conservation laws. Part 1: Linearized approximations and linearized output functionals”. In: *SIAM Journal on Numerical Analysis* 48.3 (2010), pp. 882–904. DOI: 10.1137/080727464.
- [65] M. Giles and S. Ulbrich. “Convergence of linearized and adjoint approximations for discontinuous solutions of conservation laws. Part 2: Adjoint approximations and extensions”. In: *SIAM Journal on Numerical Analysis* 48.3 (2010), pp. 905–921. DOI: 10.1137/09078078X.
- [66] Alfonso Bueno-Orovio, Carlos Castro, Karthikeyan Duraisamy, Francisco Palacios, and Enrique Zuazua. *When the ‘Exact’ Discrete Gradient is not the Best Choice in Optimal Shape Design*. AIAA paper 2011-1298. Orlando, Florida, 2011. DOI: 10.2514/6.2011-1298.
- [67] R. Löhner, Orlando Soto, and Chi Yang. “An adjoint-based design methodology for CFD optimization problems”. In: *41st Aerospace Sciences Meeting and Exhibit*. AIAA paper 2003-299. Reno, Nevada, 2003. DOI: 10.2514/6.2003-299.
- [68] Sean R. Copeland, Francisco Palacios, and Juan J. Alonso. “Adjoint-based aerothermodynamic shape design of hypersonic vehicles in non-equilibrium flows”. In: *52nd Aerospace Sciences Meeting*. National Harbor, Maryland, 2014, AIAA paper 2014-0513. DOI: 10.2514/6.2014-0513.

- [69] Sean R. Copeland. “A Continuous Adjoint Formulation for Hypersonic Flows in Thermochemical Nonequilibrium”. PhD thesis. Stanford University, 2015.
- [70] Heather L. Kline, Francisco Palacios, and Juan J. Alonso. “Sensitivity of the performance of a 3-dimensional hypersonic inlet to shape deformations”. In: *19th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*. AIAA paper 2014-3228. Atlanta, Georgia, 2014. DOI: 10.2514/6.2014-3228.
- [71] Heather L. Kline, Francisco Palacios, Thomas D. Economon, and Juan J. Alonso. “Adjoint-based optimization of a hypersonic inlet”. In: *22nd AIAA Computational Fluid Dynamics Conference*. AIAA paper 2015-3060. Dallas, Texas, 2015. DOI: 10.2514/6.2015-3060.
- [72] Heather L. Kline, Thomas D. Economon, and Juan J. Alonso. “Multi-objective optimization of a hypersonic inlet using generalized outflow boundary conditions in the continuous adjoint method”. In: *54th AIAA Aerospace Sciences Meeting*. AIAA paper 2016-0912. San Diego, California, 2016. DOI: 10.2514/6.2016-0912.
- [73] Philippe Spalart and Steven Allmaras. “A one-equation turbulence model for aerodynamic flows”. In: *30th Aerospace Sciences Meeting and Exhibit*. Reno, Nevada, 1992, AIAA paper 1992-439. DOI: 10.2514/6.1992-439.
- [74] Oktay Baysal and Mohamed E. Eleshaky. “Aerodynamic design optimization using sensitivity analysis and computational fluid dynamics”. In: *AIAA Journal* 30.3 (1992), pp. 718–725. DOI: 10.2514/3.10977.
- [75] Marian Nemec and Michael Aftosmis. “Aerodynamic shape optimization using a Cartesian adjoint method and CAD geometry”. In: *24th AIAA Applied Aerodynamics Conference*. AIAA paper 2006-3456. San Francisco, California, 2006. DOI: 10.2514/6.2006-3456.
- [76] S. Eyi and M. Yumusak. “Design optimization in hypersonic flows”. In: *18th AIAA/3AF International Space Planes and Hypersonic Systems and Technologies Conference*. AIAA paper 2012-3977. San Jose, California, 2012. DOI: 10.2514/6.2013-3977.
- [77] Sinan Eyi and Mine Yumuşak. “Aerothermodynamic shape optimization of hypersonic blunt bodies”. In: *Engineering Optimization* 47.7 (2015), pp. 909–926. DOI: 10.1080/0305215X.2014.933822.
- [78] Tugba Piskin and Sinan Eyi. “Analysis and Adjoint Design Optimization of Hypersonic Blunt Bodies”. In: *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*. AIAA paper 2014-4022. Cleveland, Ohio, 2014. DOI: 10.2514/6.2014-4022.
- [79] Andre Calado Marta. “Rapid development of discrete adjoint solvers with applications to magnetohydrodynamic flow control”. PhD thesis. Stanford University, 2007.

REFERENCES

- [80] Charles A. Mader and Joaquim R.R.A. Martins. “Towards Aerodynamic Shape Optimization of an Oblique Wing using the ADjoint Approach”. In: *Proceedings of the CASI Conference*. Toronto, Ontario, 2007.
- [81] W. Bright. *The D programming language*. 2018. URL: <http://dlang.org> (visited on 12/05/2018).
- [82] R. Gollan and P.A. Jacobs. “About the formulation, verification and validation of the hypersonic flow solver Eilmer”. In: *International Journal for Numerical Methods in Fluids* 73.1 (2013), pp. 19–57. DOI: 10.1002/flid.3790.
- [83] Peter Jacobs and Rowan Gollan. “Implementation of a compressible-flow simulation code in the D programming language”. In: *Applied Mechanics and Materials*. Vol. 846. Trans Tech Publ. 2016, pp. 54–60. DOI: 10.4028/www.scientific.net/AMM.846.54.
- [84] David C. Wilcox. *Turbulence Modeling for CFD*. Vol. 3. DCW Industries La Canada, CA, 2006.
- [85] Jiri Blazek. *Computational Fluid Dynamics: Principles and Applications*. Butterworth-Heinemann, 2015.
- [86] Christopher J. Roy and Frederick G. Blottner. “Review and assessment of turbulence models for hypersonic flows”. In: *Progress in Aerospace Sciences* 42.7-8 (2006), pp. 469–530. DOI: 10.1016/j.paerosci.2006.12.002.
- [87] Turan Coratekin, J. Van Keuk, and J. Ballman. “Performance of upwind schemes and turbulence models in hypersonic flows”. In: *AIAA Journal* 42.5 (2004), pp. 945–957. DOI: 10.2514/1.9588.
- [88] W.Y.K. Chan, P.A. Jacobs, and D.J. Mee. “Suitability of the $k-\omega$ turbulence model for scramjet flowfield simulations”. In: *International Journal for Numerical Methods in Fluids* 70.4 (2012), pp. 493–514. DOI: 10.1002/flid.2699.
- [89] Wilson Chan. “Effects of flow non-uniformities on the drag reduction by boundary layer combustion”. PhD thesis. University of Queensland, 2012.
- [90] Jimmy-John O.E. Hoste. “Scramjet combustion modeling using eddy dissipation model”. PhD thesis. University of Strathclyde, 2018.
- [91] Boris Diskin, James L. Thomas, Eric J. Nielsen, Hiroaki Nishikawa, and Jeffery A. White. “Comparison of node-centered and cell-centered unstructured finite-volume discretizations: viscous fluxes”. In: *AIAA Journal* 48.7 (2010), pp. 1326–1338. DOI: 10.2514/1.44940.
- [92] Axel Schwöppe and Boris Diskin. “Accuracy of the cell-centered grid metric in the DLR TAU-code”. In: *New Results in Numerical and Experimental Fluid Mechanics VIII*. Springer, 2013, pp. 429–437. DOI: 10.1007/978-3-642-35680-3_51.

- [93] Emre Sozer, Christoph Brehm, and Cetin C. Kiris. “Gradient calculation methods on arbitrary polyhedral unstructured meshes for cell-centered CFD solvers”. In: *52nd Aerospace Sciences Meeting*. National Harbor, Maryland, 2014, AIAA paper 2014-1440. DOI: 10.2514/6.2014-1440.
- [94] Alexandros Syrakos, Yannis Dimakopoulos, Apostolos Goulas, and John Tsamopoulos. “A critical analysis of some popular methods for the calculation of the gradient in finite volume methods, with suggestions for improvements”. In: *arXiv preprint arXiv:1606.05556* (2016).
- [95] Philip L. Roe. “Approximate Riemann solvers, parameter vectors, and difference schemes”. In: *Journal of Computational Physics* 43.2 (1981), pp. 357–372. DOI: 10.1016/0021-9991(81)90128-5.
- [96] P.L. Roe and J. Pike. “Efficient construction and utilisation of approximate Riemann solutions”. In: *Proc. of the Sixth Int’l. Symposium on Computing Methods in Applied Sciences and Engineering, VI*. North-Holland Publishing Co. 1985, pp. 499–518.
- [97] D. Hänel, R. Schwane, and G. Seider. “On the accuracy of upwind schemes for the solution of the Navier-Stokes equations”. In: *8th Computational Fluid Dynamics Conference*. AIAA paper 1987-1105. Honolulu, Hawaii, 1987. DOI: 10.2514/6.1987-1105.
- [98] Yasuhiro Wada and Meng-Sing Liou. “An accurate and robust flux splitting scheme for shock and contact discontinuities”. In: *SIAM Journal on Scientific Computing* 18.3 (1997), pp. 633–657. DOI: 10.1137/S1064827595287626.
- [99] Yasuhiro Wada and Meng-Sing Liou. “A flux splitting scheme with high-resolution and robustness for discontinuities”. In: *32nd Aerospace Sciences Meeting and Exhibit*. Reno, Nevada, 1994, AIAA paper 1994-83. DOI: 10.2514/6.1994-83.
- [100] Timothy J. Barth and Dennis C. Jespersen. “The design and application of upwind schemes on unstructured meshes”. In: *27th Aerospace Sciences Meeting*. AIAA paper 1989-366. Reno, Nevada, 1989. DOI: 10.2514/6.1989-366.
- [101] Venkat Venkatakrishnan. “Convergence to steady state solutions of the Euler equations on unstructured grids with limiters”. In: *Journal of Computational Physics* 118.1 (1995), pp. 120–130. DOI: 10.1006/jcph.1995.1084.
- [102] Jin Seok Park, Sung-Hwan Yoon, and Chongam Kim. “Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids”. In: *Journal of Computational Physics* 229.3 (2010), pp. 788–812. DOI: 10.1016/j.jcp.2009.10.011.
- [103] Jin Seok Park and Chongam Kim. “Multi-dimensional limiting process for finite volume methods on unstructured grids”. In: *Computers & Fluids* 65 (2012), pp. 8–24. DOI: 10.1016/j.compfluid.2012.04.015.

REFERENCES

- [104] Jeffery A. White, Robert A. Baurle, Bradley J. Passe, Seth C. Spiegel, and Hiroaki Nishikawa. “Geometrically flexible and efficient flow analysis of high speed vehicles via domain decomposition, part 1: unstructured-grid solver for high speed flows”. In: *2017 JANNAF - Intera-gency Propulsion Committee Meeting*. Report no. NF1676L-28704. Newport News, Virginia, 2017.
- [105] Kyle Bonner Thompson. “Aerothermodynamic Design Sensitivities for a Reacting Gas Flow Solver on an Unstructured Mesh Using a Discrete Adjoint Formulation.” PhD thesis. North Carolina State University, 2017.
- [106] James J. Quirk. “A contribution to the great Riemann solver debate”. In: *Upwind and High-Resolution Schemes*. Springer, 1997, pp. 550–569. DOI: 10.1007/978-3-642-60543-7_22.
- [107] Bernd Einfeldt. “On Godunov-type methods for gas dynamics”. In: *SIAM Journal on Numerical Analysis* 25.2 (1988), pp. 294–318. DOI: 10.1137/0725021.
- [108] Bernd Einfeldt, Claus-Dieter Munz, Philip L. Roe, and Björn Sjögreen. “On Godunov-type methods near low densities”. In: *Journal of Computational Physics* 92.2 (1991), pp. 273–295. DOI: 10.1016/0021-9991(91)90211-3.
- [109] Andreas Haselbacher and Jiri Blazek. “On the accurate and efficient discretisation of the Navier-Stokes equations on mixed grids”. In: *14th Computational Fluid Dynamics Conference*. AIAA paper 1999-3363. Norfolk, Virginia. DOI: 10.2514/6.1999-3363.
- [110] Andreas Haselbacher and Jiri Blazek. “Accurate and efficient discretization of Navier-Stokes equations on mixed grids”. In: *AIAA Journal* 38.11 (2000), pp. 2094–2102. DOI: 10.2514/2.871.
- [111] Hiroaki Nishikawa. “Beyond interface gradient: a general principle for constructing diffusion schemes”. In: *40th Fluid Dynamics Conference and Exhibit*. AIAA paper 2010-5093. Chicago, Illinois, 2010. DOI: 10.2514/6.2010-5093.
- [112] Dana A. Knoll and David E. Keyes. “Jacobian-free Newton–Krylov methods: a survey of approaches and applications”. In: *Journal of Computational Physics* 193.2 (2004), pp. 357–397. DOI: 10.1016/j.jcp.2003.08.010.
- [113] Carl Timothy Kelley and David E. Keyes. “Convergence analysis of pseudo-transient continuation”. In: *SIAM Journal on Numerical Analysis* 35.2 (1998), pp. 508–523. DOI: 10.1137/S0036142996304796.
- [114] Harvard Lomax, Thomas H. Pulliam, and David W. Zingg. *Fundamentals of Computational Fluid Dynamics*. Springer Science & Business Media, 2013.

- [115] Jason Hicken and David Zingg. “Globalization strategies for inexact-Newton solvers”. In: *19th AIAA Computational Fluid Dynamics*. San Antonio, Texas, 2009, AIAA paper 2009-4139. DOI: 10.2514/6.2009-4139.
- [116] Ron S. Dembo, Stanley C. Eisenstat, and Trond Steihaug. “Inexact newton methods”. In: *SIAM Journal on Numerical Analysis* 19.2 (1982), pp. 400–408. DOI: 10.1137/0719025.
- [117] Stanley C. Eisenstat and Homer F. Walker. “Choosing the forcing terms in an inexact Newton method”. In: *SIAM Journal on Scientific Computing* 17.1 (1996), pp. 16–32. DOI: doi.org/10.1137/0917003.
- [118] Todd T. Chisholm and David W. Zingg. “A Jacobian-free Newton–Krylov algorithm for compressible turbulent fluid flows”. In: *Journal of Computational Physics* 228.9 (2009), pp. 3490–3507. DOI: 10.1016/j.jcp.2009.02.004.
- [119] Yousef Saad. *Iterative Methods for Sparse Linear Systems*. Vol. 82. SIAM, 2003.
- [120] Yousef Saad and Henk A. Van Der Vorst. “Iterative solution of linear systems in the 20th century”. In: *Numerical Analysis: Historical Developments in the 20th Century*. Elsevier, 2001, pp. 175–207. DOI: 10.1016/S0377-0427(00)00412-X.
- [121] Youcef Saad and Martin H. Schultz. “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems”. In: *SIAM Journal on Scientific and Statistical Computing* 7.3 (1986), pp. 856–869. DOI: 10.1137/0907058.
- [122] Todd Thomas Chisholm. “A fully coupled Newton-Krylov solver with a one-equation turbulence model”. PhD thesis. University of Toronto, 2007.
- [123] Alberto Pueyo. “An efficient Newton-Krylov method for the Euler and Navier-Stokes equations.” PhD thesis. University of Toronto, 1999.
- [124] Marian Nemec. “Optimal shape design of aerodynamic configurations: A Newton-Krylov approach.” PhD thesis. University of Toronto, 2004.
- [125] Jason Hicken. “Efficient algorithms for future aircraft design: Contributions to aerodynamic shape optimization”. PhD thesis. University of Toronto, 2009.
- [126] Tony F. Chan and Kenneth R. Jackson. “Nonlinearly preconditioned Krylov subspace methods for discrete Newton algorithms”. In: *SIAM Journal on Scientific and Statistical Computing* 5.3 (1984), pp. 533–542. DOI: 10.1137/0905039.
- [127] Peter N. Brown and Alan C. Hindmarsh. “Matrix-free methods for stiff systems of ODEs”. In: *SIAM Journal on Numerical Analysis* 23.3 (1986), pp. 610–638. DOI: 10.1137/0723039.
- [128] Peter N. Brown and Youcef Saad. “Hybrid Krylov methods for nonlinear systems of equations”. In: *SIAM Journal on Scientific and Statistical Computing* 11.3 (1990), pp. 450–481. DOI: 10.1137/0911026.

REFERENCES

- [129] Hong Luo, Joseph D. Baum, and Rainald Löhner. “A fast, matrix-free implicit method for compressible flows on unstructured grids”. In: *Journal of Computational Physics* 146.2 (1998), pp. 664–690. DOI: 10.1006/jcph.1998.6076.
- [130] William Squire and George Trapp. “Using complex variables to estimate derivatives of real functions”. In: *SIAM Review* 40.1 (1998), pp. 110–112. DOI: 10.1137/S003614459631241X.
- [131] David W. Zingg and Todd T. Chisholm. “Jacobian-free Newton-Krylov methods: issues and solutions”. In: *Computational Fluid Dynamics 2006*. Springer, 2009, pp. 237–242. DOI: 10.1007/978-3-540-92779-2_35.
- [132] Awad H. Al-Mohy and Nicholas J. Higham. “The complex step approximation to the Fréchet derivative of a matrix function”. In: *Numerical Algorithms* 53.1 (2010), p. 133. DOI: 10.1007/s11075-009-9323-y.
- [133] N.P.C. Marques and J.C.F. Pereira. “Comparison of matrix-free acceleration techniques in compressible Navier–Stokes calculations”. In: *International Journal for Numerical Methods in Engineering* 61.3 (2004), pp. 455–474. DOI: 10.1002/nme.1076.
- [134] Andrew Chapman, Yousef Saad, and Larry Wigton. “High-order ILU preconditioners for CFD problems”. In: *International Journal for Numerical Methods in Fluids* 33.6 (2000), pp. 767–788. DOI: 10.1002/1097-0363(20000730)33:6<767::AID-FLD28>3.0.CO;2-C.
- [135] Marco Fossati, Wagdi G. Habashi, Song Gao, Peter Yin, Dario Isola, Guido Baruzzi, and Isik Ozcer. “Development of a Jacobian-free Finite Element Solver for Aerothermodynamic Design”. In: *53rd AIAA Aerospace Sciences Meeting*. Kissimmee, Florida, 2015, AIAA paper 2015-1261. DOI: 10.2514/6.2015-1261.
- [136] Ioannis Nompelis, Tian Wan, and Graham Candler. “Performance comparisons of parallel implicit solvers for hypersonic flow computations on unstructured meshes”. In: *18th AIAA Computational Fluid Dynamics Conference*. AIAA paper 2007-4334. Miami, Florida, 2007. DOI: 10.2514/6.2007-4334.
- [137] Matthew MacLean and Todd White. “Implementation of Generalized Minimum Residual Krylov Subspace Method for Chemically Reacting Flows”. In: *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. AIAA paper 2012-441. Nashville, Tennessee, 2012. DOI: 10.2514/6.2012-441.
- [138] Paul D. Orkwis. “Comparison of Newton’s and quasi-Newton’s method solvers for the Navier-Stokes equations”. In: *AIAA Journal* 31.5 (1993), pp. 832–836. DOI: 10.2514/3.11693.
- [139] David E. Keyes. “Aerodynamic applications of Newton-Krylov-Schwarz solvers”. In: *Fourteenth International Conference on Numerical Methods in Fluid Dynamics*. Springer, 1995, pp. 1–20. DOI: 10.1007/3-540-59280-6_95.

- [140] Jack Dongarra. *CRS-based Factorization Transpose Solve*. 1995. URL: http://www.netlib.org/linalg/html_templates/node103.html (visited on 02/01/2019).
- [141] George Karypis and Vipin Kumar. *MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0*. <http://www.cs.umn.edu/~metis>. University of Minnesota, Minneapolis, MN, 2009.
- [142] Dana A. Knoll and David E. Keyes. “Jacobian-free Newton–Krylov methods: a survey of approaches and applications”. In: *Journal of Computational Physics* 193.2 (2004), pp. 357–397. DOI: 10.1016/j.jcp.2003.08.010.
- [143] K.A. Damm, R.J. Gollan, and P.A. Jacobs. “Verification of the Least-Squares Procedure within an Unstructured-Grid Flow Solver”. In: *20th Australasian Fluid Mechanics Conference*. Perth, Australia, 2016.
- [144] Venkat Venkatakrishnan. “On the accuracy of limiters and convergence to steady state solutions”. In: *31st Aerospace Sciences Meeting*. AIAA paper 1993-880. Reno, Nevada, 1993. DOI: 10.2514/6.1993-880.
- [145] Barry W. Boehm. *Software Engineering Economics*. Vol. 197. Prentice-hall Englewood Cliffs (NJ), 1981. DOI: 10.1109/TSE.1984.5010193.
- [146] Frederick G. Blottner. “Accurate Navier-Stokes results for the hypersonic flow over a spherical nosetip”. In: *Journal of Spacecraft and Rockets* 27.2 (1990), pp. 113–122. DOI: 10.2514/3.26115.
- [147] Patrick J. Roache. “Verification of codes and calculations”. In: *AIAA Journal* 36.5 (1998), pp. 696–702. DOI: 10.2514/2.457.
- [148] Patrick J. Roache and Stanly Steinberg. “Symbolic manipulation and computational fluid dynamics”. In: *AIAA Journal* 22.10 (1984), pp. 1390–1394. DOI: 10.1016/0021-9991(85)90045-2.
- [149] William L. Oberkampf and Christopher J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, 2010.
- [150] Urmila Ghia, Sami Bayyuk, Sami Habchi, Chris Roy, Tom Shih, Terrence Conlisk, Charles Hirsch, and Joseph M. Powers. “The AIAA code verification project-test cases for CFD code verification”. In: *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. AIAA paper 2010-125. Orlando, Florida, 2010. DOI: 10.2514/6.2010-125.
- [151] Christopher J. Roy. “Review of code and solution verification procedures for computational simulation”. In: *Journal of Computational Physics* 205.1 (2005), pp. 131–156. DOI: 10.1016/j.jcp.2004.10.036.

REFERENCES

- [152] Aaron Meurer et al. “SymPy: symbolic computing in Python”. In: *PeerJ Computer Science* 3 (2017), e103. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.103.
- [153] Christopher John Roy, C.C. Nelson, T.M. Smith, and C.C. Ober. “Verification of Euler/Navier–Stokes codes using the method of manufactured solutions”. In: *International Journal for Numerical Methods in Fluids* 44.6 (2004), pp. 599–620. DOI: 10.1002/flid.660.
- [154] Christopher Roy, Eric Tendeau, Subramanya Veluri, Rifki Rifki, Edward Luke, and Shelley Hebert. “Verification of RANS turbulence models in Loci-CHEM using the method of manufactured solutions”. In: *18th AIAA Computational Fluid Dynamics Conference*. AIAA paper 2007-4203. Miami, Florida, 2007. DOI: 10.2514/6.2007-4203.
- [155] Christopher J. Roy, Thomas M. Smith, and Curtis C. Ober. “Verification of a compressible CFD code using the method of manufactured solutions”. In: *32nd AIAA Fluid Dynamics Conference and Exhibit*. AIAA paper 2002-3110. St. Louis, Missouri, 2002. DOI: 10.2514/6.2002-3110.
- [156] Subrahmanya P. Veluri, Christopher J. Roy, and Edward A. Luke. “Comprehensive code verification techniques for finite volume CFD codes”. In: *Computers & Fluids* 70 (2012), pp. 59–72. DOI: 10.1016/j.compfluid.2012.04.028.
- [157] Patrick J. Roache. “Code verification by the method of manufactured solutions”. In: *Journal of Fluids Engineering* 124.1 (2002), pp. 4–10. DOI: 10.1115/1.1436090.
- [158] Aniruddha Choudhary, Christopher J. Roy, Edward A. Luke, and Subrahmanya P. Veluri. “Code verification of boundary conditions for compressible and incompressible computational fluid dynamics codes”. In: *Computers & Fluids* 126 (2016), pp. 153–169. DOI: 10.1016/j.compfluid.2015.12.003.
- [159] Ryan B. Bond, Curtis C. Ober, Patrick M. Knupp, and Steven W. Bova. “Manufactured solution for computational fluid dynamics boundary condition verification”. In: *AIAA Journal* 45.9 (2007), pp. 2224–2236. DOI: 10.2514/1.28099.
- [160] D. G. Mabey and W. G. Sawyer. *Experimental studies of the boundary layer on a flat plate at Mach numbers from 2.5 to 4.5*. Technical Report R. and M. No. 3874. Aeronautical Research Council Reports and Memoranda, 1976.
- [161] Hans-Hermann Fernholz and P.J. Finley. *A critical compilation of compressible turbulent boundary layer data*. Advisory Group for Aerospace Research and Development Report AGARD-AG-223. Nice, France: North Atlantic Treaty Organization, 1977.
- [162] Samuel Stennett, Wilson Chan, David E. Gildfind, and Peter Jacobs. “Validating the $k-\omega$ Turbulence Model for 3D Flows within the CFD Solver Eilmer”. In: *Applied Mechanics and Materials*. Vol. 846. 2016, pp. 67–72. DOI: 10.4028/www.scientific.net/AMM.846.67.

- [163] E. Schülein, P. Krogmann, and E. Stanewsky. *Documentation of two-dimensional impinging shock/turbulent boundary layer interaction flow*. Technical Report IB 223-96 A 49. German Aerospace Center DLR, 1996.
- [164] Erich Schülein. “Skin friction and heat flux measurements in shock/boundary layer interaction flows”. In: *AIAA Journal* 44.8 (2006), pp. 1732–1741. DOI: 10.2514/1.15110.
- [165] I. Lindblad, S. Wallin, A. Johansson, R. Friedrich, R. Lechner, P. Krogmann, E. Schuelein, J.C. Courty, M. Ravachol, and D. Giordano. “A prediction method for high speed turbulent separated flows with experimental verification”. In: *29th AIAA, Fluid Dynamics Conference*. Albuquerque, New Mexico, 1998, AIAA paper 1998-2547. DOI: 10.2514/6.1998-2547.
- [166] N.N. Fedorova, I.A. Fedorchenko, and E. Schülein. “Experimental and Numerical Investigation of the Oblique Shock Wave/Turbulent Boundary Layer Interaction at $M=5$ ”. In: *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik* 81.S3 (2001), pp. 773–774. DOI: 10.1002/zamm.200108115195.
- [167] Johan Steelant. “Effect of a compressibility correction on different turbulence models”. In: *Engineering Turbulence Modelling and Experiments 5*. Elsevier, 2002, pp. 207–216. DOI: 10.1016/B978-008044114-6/50019-3.
- [168] Kevin Lenahan and Manan Vyas. *Mach 5 Shock Wave Boundary Layer Interaction: Study I*. 2015. URL: <https://www.grc.nasa.gov/WWW/wind/valid/m5swbli/study01/m5swbli1.html> (visited on 06/05/2018).
- [169] Nils Christoph Dröske. “Investigation of thermal loads onto a cooled strut injector inside a scramjet combustion chamber”. PhD thesis. University of Stuttgart, 2016.
- [170] David S. Dolling. “Fifty years of shock-wave/boundary-layer interaction research: what next?” In: *AIAA Journal* 39.8 (2001), pp. 1517–1531. DOI: 10.2514/2.1476.
- [171] M.S. Holden and T.P. Wadhams. “A database of aerothermal measurements in hypersonic flows in ‘building block’ experiments for CFD validation”. In: *41st Aerospace Sciences Meeting and Exhibit*. AIAA paper 2003-1137. Reno, Nevada, 2003. DOI: 10.2514/6.2003-1137.
- [172] Matthew MacLean and Michael Holden. “Validation and comparison of WIND and DPLR results for hypersonic, laminar problems”. In: *42nd AIAA Aerospace Sciences Meeting and Exhibit*. AIAA Paper 2004-529. Reno, Nevada, 2004. DOI: 10.2514/6.2004-529.
- [173] Graham Candler, Ioannis Nompelis, and Michael Holden. “Computational analysis of hypersonic laminar viscous-inviscid interactions”. In: *38th Aerospace Sciences Meeting and Exhibit*. AIAA paper 2000-532. Reno, Nevada, 2000. DOI: 10.2514/6.2000-532.

REFERENCES

- [174] Ioannis Nompelis, Graham V Candler, and Michael S. Holden. “Effect of vibrational nonequilibrium on hypersonic double-cone experiments”. In: *AIAA Journal* 41.11 (2003), pp. 2162–2169. DOI: 10.2514/2.6834.
- [175] Michael S. Holden and Timothy P. Wadhams. “A review of experimental studies for DSMC and Navier-Stokes code validation in laminar regions of shock/shock and shock/boundary layer interaction including real gas effects in hypervelocity flows”. In: *36 th AIAA Thermophysics Conference*. AIAA paper 2003-3641. Orlando, Florida, 2003. DOI: 10.2514/6.2003-3641.
- [176] Michael S. Holden, Timothy P. Wadhams, John K. Harvey, and Graham V. Candler. *Comparisons between measurements in regions of laminar shock wave boundary layer interaction in hypersonic flows with navier-stokes and DSMC solutions*. Conference Paper RTO-TR-AVT-007-V3. Buffalo, New York: Aerothermal and Aero-optics Evaluation Center, 2006.
- [177] Peter A. Jacobs and Rowan J. Gollan. *The Eilmer3 Code: User Guide and Example Book 2015 Edition*. School of Mechanical and Mining Engineering Report 2015/07. University of Queensland,
- [178] Michael S. Holden. *Theoretical and experimental studies of the shock wave-boundary layer interaction on compression surfaces in hypersonic flow*. Office of Aerospace Research Final Report ARL 70-0002. Aerospace Research Laboratories, 1970.
- [179] David H. Rudy, James L. Thomas, Ajay Kumar, Peter A. Gnoffo, and Sukumar R. Chakravarthy. “Computation of laminar hypersonic compression-corner flows”. In: *AIAA Journal* 29.7 (1991), pp. 1108–1113. DOI: 10.2514/3.10710.
- [180] C.M. Hung and R.W. MacCormack. “Numerical solutions of supersonic and hypersonic laminar compression corner flows”. In: *AIAA Journal* 14.4 (1976), pp. 475–481. DOI: 10.2514/3.61386.
- [181] Daniel P. Aeschliman and William L. Oberkampf. “Experimental methodology for computational fluid dynamics code validation”. In: *AIAA Journal* 36.5 (1998), pp. 733–741. DOI: 10.2514/2.461.
- [182] William L. Oberkampf and Daniel P. Aeschliman. “Joint computational/experimental aerodynamics research on a hypersonic vehicle. I-Experimental results”. In: *AIAA Journal* 30.8 (1992), pp. 2000–2009. DOI: 10.2514/3.11172.
- [183] Mary M. Walker and William L. Oberkampf. “Joint computational/experimental aerodynamics research on hypersonic vehicle. II-Computational results”. In: *AIAA Journal* 30.8 (1992), pp. 2010–2016. DOI: 10.2514/3.11173.

- [184] William Oberkampf, Daniel Aeschliman, John Henfling, and Donald Larson. “Surface pressure measurements for CFD code validation in hypersonic flow”. In: *Fluid Dynamics Conference*. AIAA paper 1995-2273. San Diego, California, 1995. DOI: 10.2514/6.1995-2273.
- [185] W. Oberkampf, D. Aeschliman, J. Henfling, D. Larson, and J. Payne. “Surface pressure measurements on a hypersonic vehicle”. In: *34th Aerospace Sciences Meeting and Exhibit*. AIAA paper 1996-669. Reno, Nevada, 1996. DOI: 10.2514/6.1996-669.
- [186] Jeffrey Payne and Mary Walker. “Verification of computational aerodynamic predictions for complex hypersonic vehicles using the INCA code”. In: *33rd Aerospace Sciences Meeting and Exhibit*. AIAA paper 1995-762. Reno, Nevada, 1995. DOI: 10.2514/6.1995-762.
- [187] Christopher J. Roy, Mary A. McWherter-Payne, and William L. Oberkampf. “Verification and validation for laminar hypersonic flowfields, part 1: verification”. In: *AIAA Journal* 41.10 (2003), pp. 1934–1943. DOI: 10.2514/2.1909.
- [188] Christopher J. Roy, William L. Oberkampf, and Mary A. McWherter-Payne. “Verification and validation for laminar hypersonic flowfields, Part 2: Validation”. In: *AIAA Journal* 41.10 (2003), pp. 1944–1954. DOI: 10.2514/2.1884.
- [189] Youcef Saad. *Krylov subspace methods in distributed computing environments*. Technical Report TR-92-126. Army High Performance Computing Center, 1992.
- [190] M. Rizk. “The use of finite-differenced Jacobians for solving the Euler equations and for evaluating sensitivity derivatives”. In: *Fluid Dynamics Conference*. AIAA paper 1994-2213. Colorado Spring, Colorado, 1994. DOI: 10.2514/6.1994-2213.
- [191] G.D. Van Albada, Bram Van Leer, and WWjun Roberts. “A comparative study of computational methods in cosmic gas dynamics”. In: *Upwind and High-Resolution Schemes*. Springer, 1997, pp. 95–103. DOI: 10.1007/978-3-642-60543-7_6.
- [192] Bram Van Leer. “Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method”. In: *Journal of computational Physics* 32.1 (1979), pp. 101–136. URL: [https://doi.org/10.1016/0021-9991\(79\)90145-1](https://doi.org/10.1016/0021-9991(79)90145-1).
- [193] Marian Nemec, David W. Zingg, and Thomas H. Pulliam. “Multipoint and multi-objective aerodynamic shape optimization”. In: *AIAA Journal* 42.6 (2004), pp. 1057–1065. DOI: 10.2514/1.10415.
- [194] Joaquim R.R.A. Martins, Peter Sturdza, and Juan J. Alonso. “The complex-step derivative approximation”. In: *ACM Transactions on Mathematical Software (TOMS)* 29.3 (2003), pp. 245–262. DOI: 10.1145/838250.838251.

REFERENCES

- [195] G. Simeonides, W. Haase, and M. Manna. “Experimental, analytical, and computational methods applied to hypersonic compression ramp flows”. In: *AIAA Journal* 32.2 (1994), pp. 301–310. DOI: 10.2514/3.11985.
- [196] Giampietro Carpentieri. “An adjoint-based shape-optimization method for aerodynamic design”. PhD thesis. TU Delft, 2009.
- [197] L. Hascoët and V. Pascual. “The Tapenade Automatic Differentiation tool: Principles, Model, and Specification”. In: *ACM Transactions On Mathematical Software* 39.3 (2013). DOI: 10.1145/2450153.2450158.
- [198] Jason E. Hicken and David W. Zingg. “Aerodynamic optimization algorithm with integrated geometry parameterization and mesh movement”. In: *AIAA Journal* 48.2 (2010), pp. 400–413. DOI: 10.2514/1.44033.
- [199] Jamshid A. Samareh. “Status and future of geometry modeling and grid generation for design and optimization”. In: *Journal of Aircraft* 36.1 (1999), pp. 97–104. DOI: doi.org/10.2514/2.2417.
- [200] O. Soto, R. Löhner, and C. Yang. “A stabilized pseudo-shell approach for surface parametrization in CFD design problems”. In: *Communications in Numerical Methods in Engineering* 18.4 (2002), pp. 251–258. DOI: 10.1002/cnm.490.
- [201] W. Kyle Anderson and V. Venkatakrishnan. “Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation”. In: *Computers & Fluids* 28.4 (1999), pp. 443–480. DOI: 10.1016/S0045-7930(98)00041-3.
- [202] Jamshid A. Samareh. “Aerodynamic shape optimization based on free-form deformation”. In: *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. AIAA paper 2004-4630. Albany, New York, 2004. DOI: 10.2514/6.2004-4630.
- [203] David F. Rogers. *An Introduction to NURBS: With Historical Perspective*. Elsevier, 2000.
- [204] Greg W. Burgreen, Oktay Baysal, and Mohamed E. Eleshaky. “Improving the efficiency of aerodynamic shape optimization”. In: *AIAA Journal* 32.1 (1994), pp. 69–76. DOI: 10.2514/3.11952.
- [205] V. Venkatakrishnan and D.J. Mavriplis. “Implicit method for the computation of unsteady flows on unstructured grids”. In: *Journal of Computational Physics* (1995). DOI: 10.1006/jcph.1996.0182.
- [206] John T. Batina. “Unsteady Euler airfoil solutions using unstructured dynamic meshes”. In: *AIAA Journal* 28.8 (1990), pp. 1381–1388. DOI: 10.2514/3.25229.

- [207] John T. Batina. “Unsteady Euler algorithm with unstructured dynamic mesh for complex-aircraft aerodynamic analysis”. In: *AIAA Journal* 29.3 (1991), pp. 327–333. DOI: 10.2514/3.10583.
- [208] C.H. Farhat, C. Degand, B. Koobus, and M. Lesoinne. “Torsional springs for two-dimensional dynamic unstructured fluid meshes”. In: *Computer Methods in Applied Mechanics and Engineering* 163.1 (1998), pp. 231–245. DOI: 10.1016/S0045-7825(98)00016-4.
- [209] Clarence O.E. Burg. “A robust unstructured grid movement strategy using three-dimensional torsional springs”. In: *AIAA Journal* 2529 (2004). DOI: 10.2514/6.2004-2529.
- [210] Eric J. Nielsen and W. Kyle Anderson. “Recent improvements in aerodynamic design optimization on unstructured meshes”. In: *AIAA Journal* 40.6 (2002), pp. 1155–1163. DOI: 10.2514/2.1765.
- [211] Hyoung-Jin Kim, Daisuke Sasaki, Shigeru Obayashi, and Kazuhiro Nakahashi. “Aerodynamic optimization of supersonic transport wing using unstructured adjoint method”. In: *AIAA Journal* 39.6 (2001), pp. 1011–1020. DOI: 10.2514/2.1441.
- [212] Jeroen Witteveen. “Explicit and robust inverse distance weighting mesh deformation for CFD”. In: *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. AIAA paper 2010-165. 2010. DOI: 10.2514/6.2010-165.
- [213] B.M. Adams, L.E. Bauman, W.J. Bohnhoff, K.R. Dalbey, M.S. Ebeida, J.P. Eddy, M.S. Eldred, P.D. Hough, K.T. Hu, J.D. Jakeman, et al. *Dakota: a multilevel parallel object-oriented framework for design optimization, parameter estimation, etc: Version 6 User’s Manual*. Sandia Technical Report SAND2014-4633. Sandia National Laboratories, 2015.
- [214] Reeves Fletcher and Colin M. Reeves. “Function minimization by conjugate gradients”. In: *The Computer Journal* 7.2 (1964), pp. 149–154. DOI: 10.1093/comjnl/7.2.149.
- [215] W. O. Landsberg. “Improving performance of high Mach number scramjets: fuelling strategies and combustor design”. PhD thesis. University of Queensland, 2018.
- [216] Z. Denman. “Supersonic combustion of hydrocarbon fuels in a three-dimensional Mach 8 scramjet”. PhD thesis. University of Queensland, 2017.
- [217] Theodore Von Karman. “Supersonic aerodynamics: principles and applications”. In: *Journal of Aeronautical Sciences* (1947). DOI: <https://doi.org/10.2514/8.1394>.
- [218] Holt Ashley, Mårten Landahl, and Marten T. Landahl. *Aerodynamics of Wings and Bodies*. Chelmsford, Massachusetts: Courier Corporation, 1965.
- [219] J.D. Cole. “Newtonian flow theory for slender bodies”. In: *Journal of the Aeronautical Sciences* 24.6 (1957), pp. 448–455. DOI: 10.2514/8.3875.

REFERENCES

- [220] J.R. Eggers, Meyer M. Resnikoff, and David H. Dennis. *Bodies of revolution having minimum drag at high supersonic airspeeds*. National Advisory Committee for Aeronautics Technical Note NACA-TN-3666. 1956.
- [221] W.H. Mason and Jaewoo Lee. “Aerodynamically blunt and sharp bodies”. In: *Journal of Spacecraft and Rockets* 31.3 (1994), pp. 378–382. DOI: 10.2514/3.26449.
- [222] Martin R. Fink. “Hypersonic minimum-drag slender bodies of revolution.” In: *AIAA Journal* 4.10 (1966), pp. 1717–1724. DOI: 10.2514/3.3768.
- [223] W. Mason and Jaewoo Lee. “On optimal supersonic/hypersonic bodies”. In: *Flight Simulation Technologies Conference and Exhibit*. AIAA paper 1990-3072. Dayton, Ohio, 1992. DOI: 10.2514/6.1990-3072.
- [224] Amal Sahai, Bibin John, and Ganesh Natarajan. “Effect of fineness ratio on minimum-drag shapes in hypersonic flows”. In: *Journal of Spacecraft and Rockets* 51.3 (2014), pp. 900–907. DOI: 10.2514/1.A32583.
- [225] Wataru Yamazaki, Kisa Matsushima, and Kazuhiro Nakahashi. “Aerodynamic design optimization using the drag-decomposition method”. In: *AIAA Journal* 46.5 (2008), pp. 1096–1106. DOI: 10.2514/1.30342.
- [226] J. J. Thibert. *Drag Prediction and Analysis from Computational Fluid Dynamics*. Advisory Group for Aerospace Research and Development Report AGARD-AR-256. Nice, France: North Atlantic Treaty Organization, 1989.
- [227] N. Yu, H. Chen, and S. Samant. “Inviscid drag calculations for transonic flows”. In: *6th Computational Fluid Dynamics Conference Danvers*. AIAA paper 1983-1928. Danvers, Massachusetts, 1983. DOI: 10.2514/6.1983-1928.
- [228] Geoffrey Ingram Taylor and J.W. Maccoll. “The air pressure on a cone moving at high speeds II”. In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 139.838 (1933), pp. 298–311. DOI: 10.1098/rspa.1933.0018.
- [229] J.W. Maccoll. “The conical shock wave formed by a cone moving at a high speed”. In: *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences* 159.898 (1937), pp. 459–472. DOI: 10.1098/rspa.1937.0083.
- [230] Chad Eric Burdyslaw. “Achieving Automatic Concurrency Between Computational Field Solvers and Adjoint Sensitivity Codes”. PhD thesis. University of Tennessee at Chattanooga, Computational Engineering, 2006.

- [231] Will O. Landsberg, Vincent Wheatley, and Ananthanarayanan Veeraragavan. “Characteristics of cascaded fuel injectors within an accelerating scramjet combustor”. In: *AIAA Journal* (2016), pp. 3692–3700. DOI: 10.2514/1.J054815.
- [232] A. Vernon Gnos, Earl C. Watson, William R. Seebaugh, Robert J. Sanator, and Joseph P. DeCarlo. *Investigation of flow fields within large scale hypersonic inlet models*. NASA Technical Note D-7150. National Aeronautics and Space Administration, 1973.
- [233] M.K. Smart. “Design of three-dimensional hypersonic inlets with rectangular-to-elliptical shape transition”. In: *Journal of Propulsion and Power* 15.3 (1999), pp. 408–416. DOI: 10.2514/2.5459.
- [234] Rowan Gollan, Rowan Gollan, Paul Ferlemann, and Paul Ferlemann. “Investigation of REST-class hypersonic inlet designs”. In: *17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*. AIAA paper 2011-2254. San Francisco, California, 2011. DOI: 10.2514/6.2011-2254.
- [235] Andrew Gelsey, Doyle Knight, Song Gao, and Mark Schwabacher. “NPARC simulation and redesign of the NASA P2 hypersonic inlet”. In: *31st Joint Propulsion Conference and Exhibit*. AIAA paper 1995-2760. San Diego, California, 1995. DOI: 10.2514/6.1995-2760.
- [236] Vijay Shukla, Andrew Gelsey, Mark Schwabacher, Donald Smith, and Doyle Knight. “Automated redesign of the NASA P8 hypersonic inlet using numerical optimization”. In: *32nd Joint Propulsion Conference and Exhibit*. AIAA paper 1996-2549. Lake Buena Vista, Florida, 1996. DOI: 10.2514/6.1996-2549.
- [237] Vijay Shukla, Andrew Gelsey, Mark Schwabacher, Donald Smith, and Doyle D. Knight. “Automated design optimization for the P2 and P8 hypersonic inlets”. In: *Journal of Aircraft* 34.2 (1997), pp. 228–235. DOI: 10.2514/2.2161.
- [238] Susumu Hasegawa and Doyle Knight. “Numerical Analysis and Optimization of Two-Dimensional Hypersonic Inlets”. In: *42nd AIAA Aerospace Sciences Meeting and Exhibit*. AIAA paper 2004-856. Reno, Nevada, 2004. DOI: 10.2514/6.2004-856.
- [239] Susumu Hasegawa and Doyle Knight. “Application of optimization algorithms to scramjet inlet design”. In: *AIAA/CIRA 13th International Space Planes and Hypersonics Systems and Technologies Conference*. AIAA paper 2005-3207. Capua, Italy, 2005. DOI: 10.2514/6.2005-3207.
- [240] Travis William Drayna. “Design and Optimization of Hypersonic Inward-Turning Inlets”. PhD thesis. University of Minnesota, 2011.
- [241] Doyle D. Knight. “Numerical simulation of realistic high-speed inlets using the Navier-Stokes equations”. In: *AIAA Journal* 15.11 (1977), pp. 1583–1589. DOI: 10.2514/3.60825.

REFERENCES

- [242] Various. *Air Intakes for High Speeds Vehicles*. Advisory Group for Aerospace Research and Development Report AGARD-AG-270. Nice, France: North Atlantic Treaty Organization, 1991.
- [243] Rowan J. Gollan and Michael K. Smart. “Design of modular shape-transition inlets for a conical hypersonic vehicle”. In: *Journal of Propulsion and Power* 29.4 (2013), pp. 832–838. DOI: 10.2514/1.B34672.
- [244] J.O. Border and R.J. Gollan. “Investigating the use of Walsh functions for computing flows with unsteady shocks”. In: *20th Australasian Fluid Mechanics Conference*. Paper Number: 741. Adelaide, Australia, 2018, pp. 741.1–741.4.
- [245] Peter A. Gnoffo. “Global series solutions of nonlinear differential equations with shocks using Walsh functions”. In: *Journal of Computational Physics* 258 (2014), pp. 650–688. DOI: 10.1016/j.jcp.2013.10.054.
- [246] Alireza Mazaheri and Hiroaki Nishikawa. “Very efficient high-order hyperbolic schemes for time-dependent advection–diffusion problems: Third-, fourth-, and sixth-order”. In: *Computers & Fluids* 102 (2014), pp. 131–147. DOI: 10.1016/j.compfluid.2014.06.020.
- [247] Alireza Mazaheria and Hiroaki Nishikawa. “High-order shock-capturing hyperbolic residual-distribution schemes on irregular triangular grids”. In: *Computers & Fluids* 131 (2016), pp. 29–44. DOI: 10.1016/j.compfluid.2016.03.012.

APPENDIX A

METHOD OF MANUFACTURED SOLUTIONS CONSTANTS

A.1 2D Euler Equations

Table A.1: Constants for 2D supersonic Euler manufactured solution.

Equation, ϕ	ϕ_0	ϕ_x	ϕ_y	ϕ_{xy}	a_{ϕ_x}	a_{ϕ_y}	$a_{\phi_{xy}}$
ρ (kg/m ³)	1	0.15	-0.1	0	1	0.5	0
u (m/s)	800	50	-30	0	1.5	0.6	0
v (m/s)	800	-75	40	0	0.5	2/3	0
p (N/m ²)	1×10^5	0.2×10^5	0.5×10^5	0	2	1	0

Table A.2: $f_s(\cdot)$ functions for 2D Euler and 2D subsonic Navier-Stokes manufactured solution.

Equation, ϕ	f_{s_x}	f_{s_y}	$f_{s_{xy}}$
ρ (kg/m ³)	sin	cos	cos
u (m/s)	sin	cos	cos
v (m/s)	cos	sin	cos
p (N/m ²)	cos	sin	sin

A.2 2D Navier-Stokes Equations

Table A.3: Constants for 2D subsonic Navier-Stokes manufactured solution.

Equation, ϕ	ϕ_0	ϕ_x	ϕ_y	ϕ_{xy}	a_{ϕ_x}	a_{ϕ_y}	$a_{\phi_{xy}}$
ρ (kg/m ³)	1	0.1	0.15	0.08	0.75	1.0	1.25
u (m/s)	70	4	-12	7	5/3	1.5	0.6
v (m/s)	90	-20	4	-11	1.5	1.0	0.9
p (N/m ²)	1×10^5	-0.3×10^5	0.2×10^5	-0.25×10^5	1.0	1.25	0.75

Table A.4: $f_s(\cdot)$ functions for 2D Euler and 2D subsonic Navier-Stokes manufactured solution.

Equation, ϕ	f_{s_x}	f_{s_y}	$f_{s_{xy}}$
ρ (kg/m ³)	sin	cos	cos
u (m/s)	sin	cos	cos
v (m/s)	cos	sin	cos
p (N/m ²)	cos	sin	sin

A.3 2D RANS Equations

Table A.5: Constants for 2D $k - \omega$ Reynolds-Averaged Navier-Stokes manufactured solution.

Equation, ϕ	ϕ_0	ϕ_x	ϕ_y	ϕ_{xy}	a_{ϕ_x}	a_{ϕ_y}	$a_{\phi_{xy}}$
ρ (kg/m ³)	1	0.15	-0.1	0.08	0.75	1	1.25
u (m/s)	70	7	-8	5.5	1.5	1.5	0.6
v (m/s)	90	-5	10	-11	1.5	1	0.9
p (N/m ²)	1×10^5	0.2×10^5	0.175×10^5	-0.25×10^5	1	1.25	0.75
k (m ² /m ²)	780	160	-120	80	0.65	0.7	0.8
ω (1/s)	150	-30	22.5	40	0.75	0.875	0.6

Table A.6: $f_s(\cdot)$ functions for 2D RANS manufactured solution.

Equation, ϕ	f_{s_x}	f_{s_y}	$f_{s_{xy}}$
ρ (kg/m ³)	cos	sin	cos
u (m/s)	sin	cos	cos
v (m/s)	sin	cos	cos
p (N/m ²)	cos	sin	sin
k (m ² /m ²)	cos	sin	cos
ω (1/s)	cos	sin	cos

A.4 3D Euler Equations

Table A.7: Constants for 3D supersonic Euler manufactured solution.

Equation, ϕ	ϕ_0	ϕ_x	ϕ_y	ϕ_z	a_{ϕ_x}	a_{ϕ_y}	a_{ϕ_z}
ρ (kg/m ³)	1	0.15	-0.1	-0.12	1	0.5	1.5
u (m/s)	800	50	-30	-18	1.5	0.6	0.5
v (m/s)	800	-75	40	-30	0.5	2/3	1.25
w (m/s)	800	15	-25	35	1/3	1.5	1
p (N/m ²)	1×10^5	0.2×10^5	0.5×10^5	-0.35×10^5	2	1	1/3

Table A.8: $f_s(\cdot)$ functions for 3D Euler manufactured solution.

Equation, ϕ	f_{s_x}	f_{s_y}	$f_{s_{xy}}$
ρ (kg/m ³)	sin	cos	sin
u (m/s)	sin	cos	cos
v (m/s)	cos	sin	sin
w (m/s)	sin	sin	cos
p (N/m ²)	cos	sin	cos

A.5 3D Navier-Stokes Equations

Table A.9: Constants for 3D subsonic Navier-Stokes manufactured solution.

Equation, ϕ	ϕ_0	ϕ_x	ϕ_y	ϕ_z	ϕ_{xy}	ϕ_{yz}	ϕ_{zx}
ρ (kg/m ³)	1	0.1	-0.1	0.1	0.08	0.05	0.12
u (m/s)	70	7	-15	-10	7	4	-4
v (m/s)	90	-5	10	5	-11	-5	5
w (m/s)	80	-10	10	12	-12	11	5
p (N/m ²)	1×10^5	0.2×10^5	0.5×10^5	0.2×10^5	0.25×10^5	-0.1×10^5	0.1×10^5
		a_x	a_y	a_z	a_{xy}	a_{yz}	a_{zx}
ρ (kg/m ³)		0.75	0.45	0.8	0.65	0.75	0.5
u (m/s)		0.5	0.85	0.4	0.6	0.8	0.9
v (m/s)		0.8	0.8	0.5	0.9	0.4	0.6
w (m/s)		0.85	0.9	0.5	0.4	0.8	0.75
p (N/m ²)		0.4	0.45	0.85	0.75	0.7	0.8

Table A.10: $f_s(\cdot)$ functions for 3D subsonic Navier-Stokes manufactured solution.

Equation, ϕ	f_{s_x}	f_{s_y}	f_{s_z}	$f_{s_{xy}}$	$f_{s_{yz}}$	$f_{s_{zx}}$
ρ (kg/m ³)	cos	sin	sin	cos	sin	cos
u (m/s)	sin	cos	cos	cos	sin	cos
v (m/s)	sin	cos	cos	cos	sin	cos
w (m/s)	cos	sin	cos	sin	sin	cos
p (N/m ²)	cos	cos	sin	cos	sin	cos

A.6 3D RANS Equations

Table A.11: Constants for 3D subsonic Navier-Stokes manufactured solution.

Equation, ϕ	ϕ_0	ϕ_x	ϕ_y	ϕ_z	ϕ_{xy}	ϕ_{yz}	ϕ_{zx}
ρ (kg/m ³)	1	0.1	-0.1	0.1	0.08	0.05	0.12
u (m/s)	70	7	-15	-10	7	4	-4
v (m/s)	90	-5	10	5	-11	-5	5
w (m/s)	80	-10	10	12	-12	11	5
p (N/m ²)	1×10^5	0.2×10^5	0.5×10^5	0.2×10^5	0.25×10^5	-0.1×10^5	0.1×10^5
k (m ² /m ²)	780	160	-120	80	80	60	-70
ω (1/s)	150	-30	22.5	20	40	-15	25
		a_x	a_y	a_z	a_{xy}	a_{yz}	a_{zx}
ρ (kg/m ³)		0.75	0.45	0.8	0.65	0.75	0.5
u (m/s)		0.5	0.85	0.4	0.6	0.8	0.9
v (m/s)		0.8	0.8	0.5	0.9	0.4	0.6
w (m/s)		0.85	0.9	0.5	0.4	0.8	0.75
p (N/m ²)		0.4	0.45	0.85	0.75	0.7	0.8
k (m ² /m ²)		0.65	0.7	0.8	0.8	0.85	0.6
ω (1/s)		0.75	0.875	0.65	0.6	0.75	0.8

 Table A.12: $f_s(\cdot)$ functions for 3D subsonic Navier-Stokes manufactured solution.

Equation, ϕ	f_{s_x}	f_{s_y}	f_{s_z}	$f_{s_{xy}}$	$f_{s_{yz}}$	$f_{s_{zx}}$
ρ (kg/m ³)	cos	sin	sin	cos	sin	cos
u (m/s)	sin	cos	cos	cos	sin	cos
v (m/s)	sin	cos	cos	cos	sin	cos
w (m/s)	cos	sin	cos	sin	sin	cos
p (N/m ²)	cos	cos	sin	cos	sin	cos
k (m ² /m ²)	cos	cos	sin	cos	cos	sin
ω (1/s)	cos	cos	sin	cos	cos	sin

ADDITIONAL FLOW SOLVER TEST CASES

Chapter 4 presented results from several test cases used throughout the development of the flow solver. The details of these simulations are presented here.

B.1 Supersonic ramp

This test case was taken from Marques et al. [133]. The geometry was extracted from Figure 1 on p. 461 of the cited paper, the extracted dimensions are presented in Figure B.1. The fine mesh presented by Marques et al. [133] was recreated as closely as possible using triangular elements with the Pointwise™ orthonormal advancing front algorithm. The generated grid is also shown in Figure B.1, and consisted of 9897 cells. The inflow conditions for the simulation are presented in Table B.1. A colour map of the pressure field is presented in Figure B.2.

Table B.1: Freestream inflow conditions for supersonic ramp simulation.

Mach number	2.0
Velocity, m·s ⁻¹	694.377
Pressure, Pa	10,000.0
Temperature, K	300.0

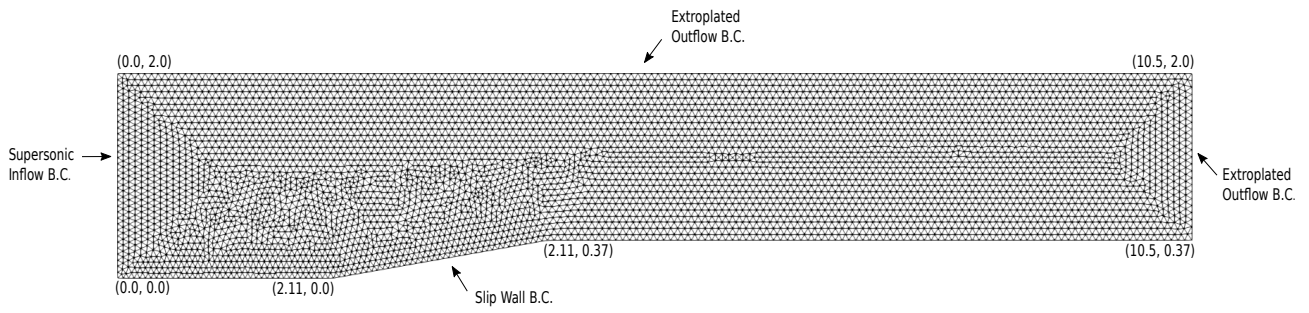


Figure B.1: Supersonic ramp simulation schematic.

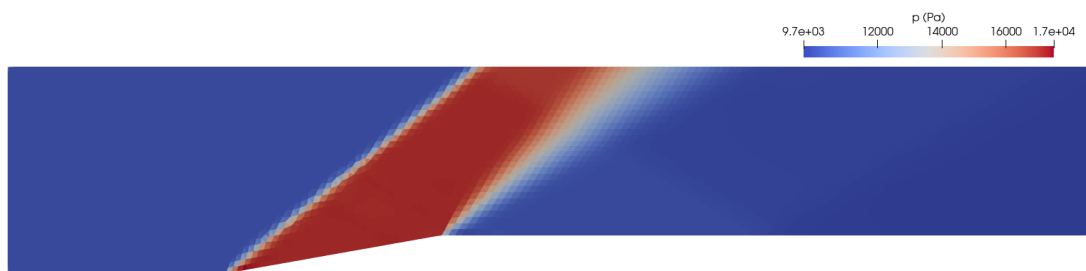


Figure B.2: Colour map of pressure for supersonic ramp simulation.

B.2 Mach 10 blunted wedge

The geometry for this test case is a blunted wedge. The radius of curvature of blunted nose is 0.005 m, the wedge angle is 10° , and the length of the model is 0.3 m. A schematic for the simulation is shown in Figure B.3. The grid, also shown in Figure B.3, consisted of 150×90 cells, with a first cell width off the wall of 1.0×10^{-5} m. The inflow conditions for the simulation are presented in Table B.2. The wall temperature was set to 287 K. A colour map of the pressure field is presented in Figure B.4.

Table B.2: Freestream inflow conditions for blunted wedge simulation.

Mach number	9.59
Velocity, $\text{m}\cdot\text{s}^{-1}$	1386.2
Pressure, Pa	74.62
Temperature, K	52.0

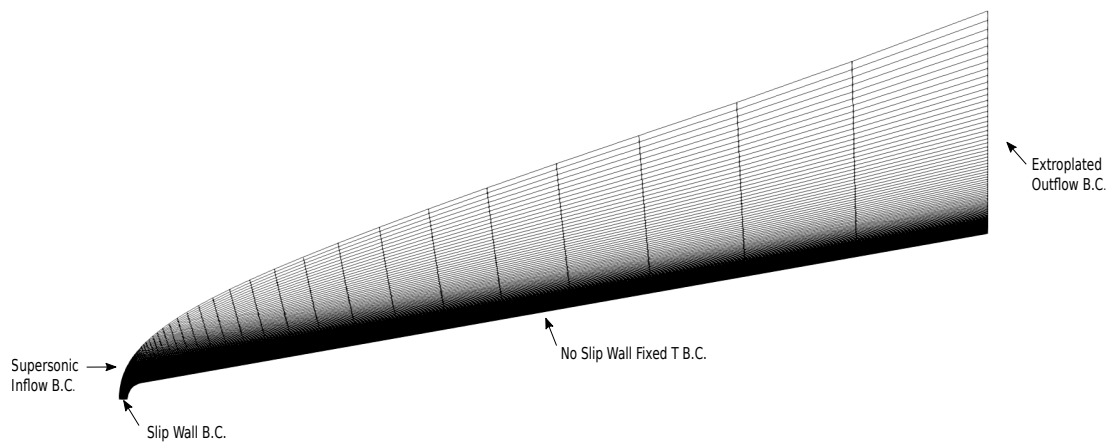


Figure B.3: Blunted wedge simulation schematic.

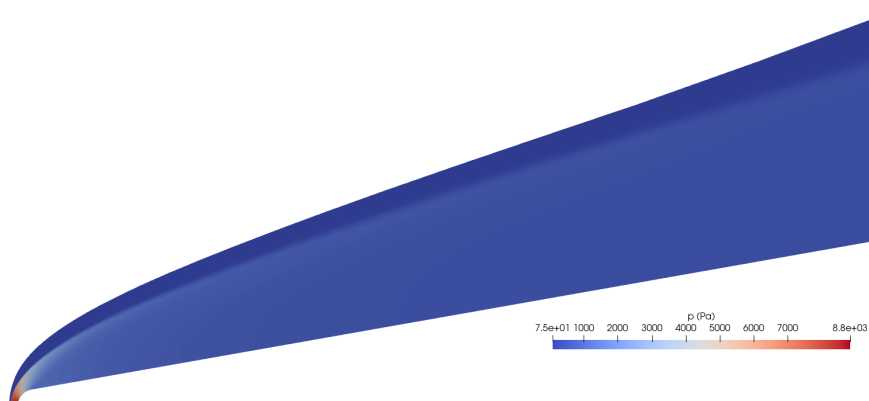


Figure B.4: Colour map of pressure for blunted-wedge simulation.

B.3 Mach 8 cylinder

The geometry for this test case is a cylinder with a radius of 0.005 m. A schematic for the simulation is shown in Figure B.5(a). The grid, also shown in Figure B.5(a), consisted of 100×100 cells. The inflow conditions for the simulation are presented in Table B.3. A colour map of the pressure field is presented in Figure B.5(b).

Table B.3: Freestream inflow conditions for cylinder simulation.

Mach number	8.0
Velocity, $\text{m}\cdot\text{s}^{-1}$	2777.51
Pressure, Pa	10,000.0
Temperature, K	300.0

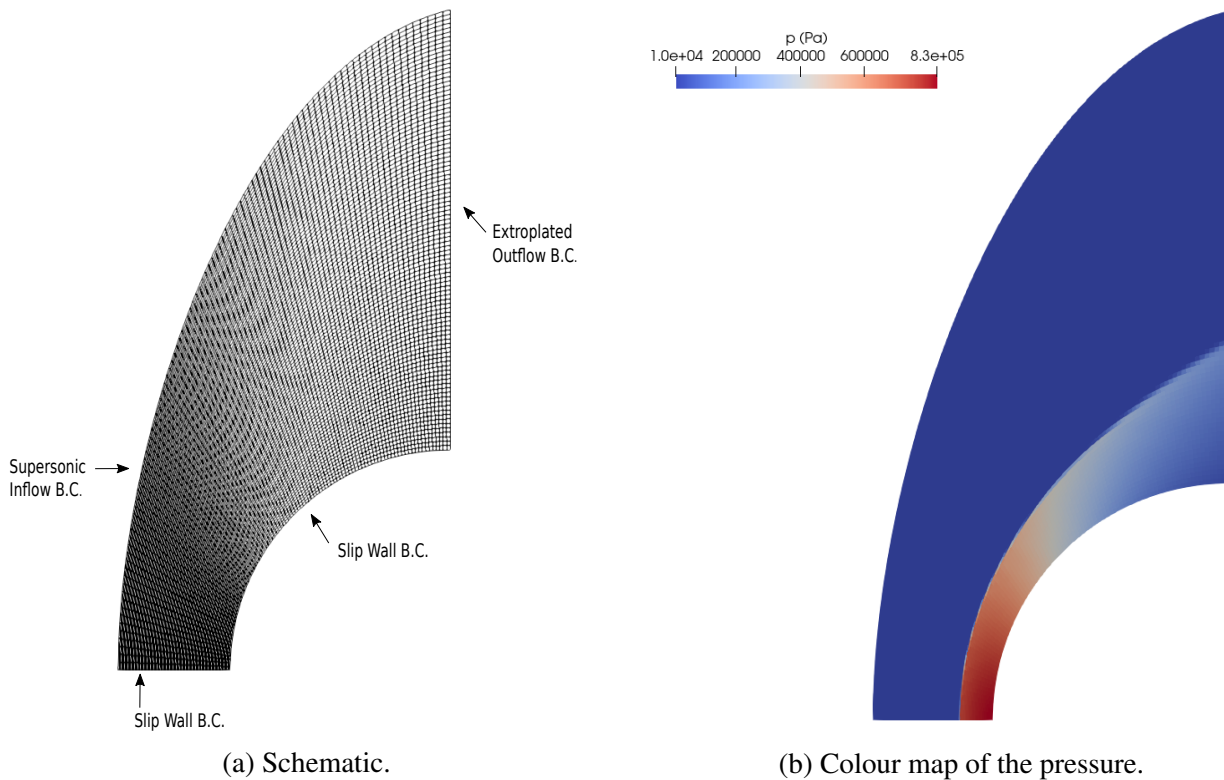


Figure B.5: Cylinder simulation

APPENDIX C

DERIVATION OF TRANSFORM JACOBIAN

The adjoint solver in Eilmer operates on the primitive flow state variables. Consequently, the adjoint operator is of the form $\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}$. The approximate low-order flow Jacobian used as the precondition matrix in the flow solver, however, requires $\frac{\partial \mathbf{R}}{\partial \mathbf{U}}$, i.e. the flow Jacobian with respect to the conserved flow state variables. Consequently, a transform matrix is needed during the construction of the precondition matrix. This transform matrix is defined such that

$$\frac{\partial \mathbf{R}}{\partial \mathbf{U}} = \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \cdot \frac{\partial \mathbf{Q}}{\partial \mathbf{U}},$$

where $\frac{\partial \mathbf{Q}}{\partial \mathbf{U}}$ is the transform matrix. For the RANS equations coupled with the $k - \omega$ turbulence model, the primitive and conservative flow state variables are given as,

$$\mathbf{Q} = \begin{bmatrix} \rho \\ u \\ v \\ w \\ p \\ k \\ \omega \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \\ U_6 \\ U_7 \end{bmatrix} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \\ \rho k \\ \rho \omega \end{bmatrix} \quad (\text{C.1})$$

where the total energy is defined as

$$E = \frac{p}{\gamma - 1} + \frac{\rho(u^2 + v^2 + w^2)}{2} + k. \quad (\text{C.2})$$

Note that the Reynolds and Favre averaging superscripts have been omitted for simplicity. Also note that the velocity components are now represented by u, v, w for clarity. To derive the transform Jacobian, \mathbf{Q} is first written in terms of \mathbf{U}

$$\mathbf{Q} = \begin{bmatrix} \rho \\ u \\ v \\ w \\ p \\ k \\ \omega \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2/U_1 \\ U_3/U_1 \\ U_4/U_1 \\ (\gamma-1)\left(U_5 - \frac{U_2^2+U_3^2+U_4^2}{2U_1} - U_6\right) \\ U_6/U_1 \\ U_7/U_1 \end{bmatrix}. \quad (\text{C.3})$$

The transform Jacobian is then given as

$$\frac{\partial \mathbf{Q}}{\partial \mathbf{U}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{U_2}{U_1^2} & \frac{1}{U_1} & 0 & 0 & 0 & 0 & 0 \\ -\frac{U_3}{U_1^2} & 0 & \frac{1}{U_1} & 0 & 0 & 0 & 0 \\ -\frac{U_4}{U_1^2} & 0 & 0 & \frac{1}{U_1} & 0 & 0 & 0 \\ (\gamma-1)\frac{(U_2^2+U_3^2+U_4^2)}{2U_1^2} & -\frac{U_2}{U_1}(\gamma-1) & -\frac{U_3}{U_1}(\gamma-1) & -\frac{U_4}{U_1}(\gamma-1) & \gamma-1 & -(\gamma-1) & 0 \\ -\frac{U_6}{U_1^2} & 0 & 0 & 0 & 0 & \frac{1}{U_1} & 0 \\ -\frac{U_7}{U_1^2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{U_1} \end{bmatrix} \quad (\text{C.4})$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 \\ -\frac{v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 & 0 & 0 \\ -\frac{w}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 & 0 & 0 \\ \frac{\gamma-1}{2}(u^2+v^2+w^2) & -u(\gamma-1) & -v(\gamma-1) & -w(\gamma-1) & \gamma-1 & -(\gamma-1) & 0 \\ -\frac{k}{\rho} & 0 & 0 & 0 & 0 & \frac{1}{\rho} & 0 \\ -\frac{\omega}{\rho} & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho} \end{bmatrix} \quad (\text{C.5})$$

APPENDIX D

MINIMUM-DRAG SLENDER BODY OF REVOLUTION BASELINE AND OPTIMISED GEOMETRIES

Presented below is an example result from the minimum-drag slender body of revolution optimisation application from Chapter 8. The geometric details for the $L/D = 5$ case, parameterised using a 20-point Bézier curve and a 40-point Bézier curve, are provided.

APPENDIX D. MINIMUM-DRAG SLENDER BODY OF REVOLUTION BASELINE AND OPTIMISED GEOMETRIES

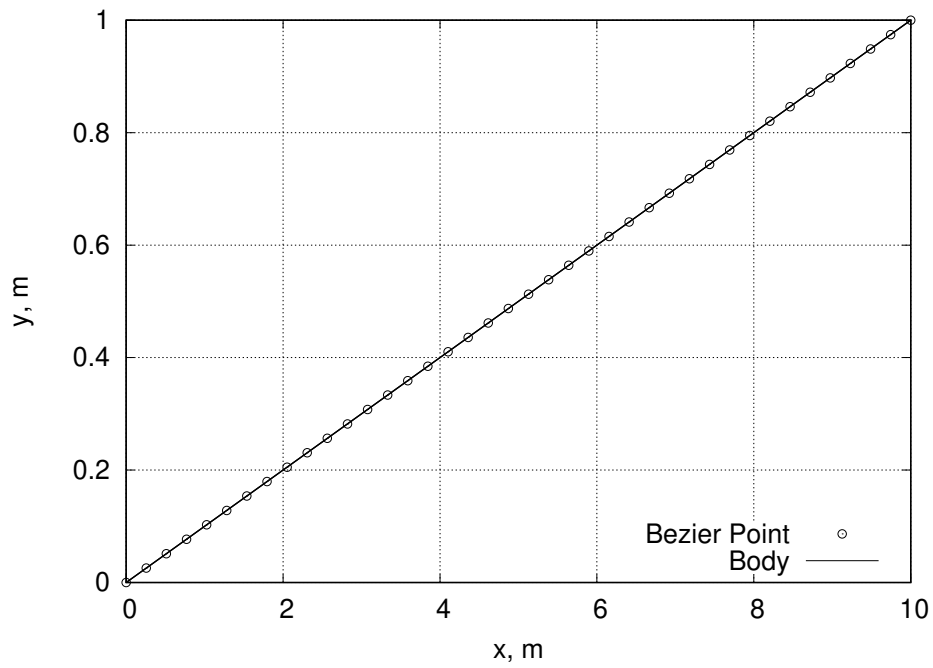
Table D.1: 20-point Bézier coordinates for minimum-drag slender body of revolution (in meters).

Baseline		Optimised	
x-coordinate	y-coordinate	x-coordinate	y-coordinate
0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
5.2631e-01	5.2631e-02	5.2631e-01	2.0565e-01
1.0526e+00	1.0526e-01	1.0526e+00	1.2788e-01
1.5789e+00	1.5789e-01	1.5789e+00	3.3311e-01
2.1052e+00	2.1052e-01	2.1052e+00	3.6057e-01
2.6315e+00	2.6315e-01	2.6315e+00	3.7454e-01
3.1578e+00	3.1578e-01	3.1578e+00	4.3411e-01
3.6842e+00	3.6842e-01	3.6842e+00	5.0854e-01
4.2105e+00	4.2105e-01	4.2105e+00	5.6567e-01
4.7368e+00	4.7368e-01	4.7368e+00	6.0317e-01
5.2631e+00	5.2631e-01	5.2631e+00	6.3688e-01
5.7894e+00	5.7894e-01	5.7894e+00	6.7956e-01
6.3157e+00	6.3157e-01	6.3157e+00	7.3045e-01
6.8421e+00	6.8421e-01	6.8421e+00	7.7947e-01
7.3684e+00	7.3684e-01	7.3684e+00	8.1852e-01
7.8947e+00	7.8947e-01	7.8947e+00	8.4977e-01
8.4210e+00	8.4210e-01	8.4210e+00	8.8479e-01
8.9473e+00	8.9473e-01	8.9473e+00	9.3339e-01
9.4736e+00	9.4736e-01	9.4736e+00	9.6301e-01
1.0000e+01	1.0000e+00	1.0000e+01	1.0000e+00

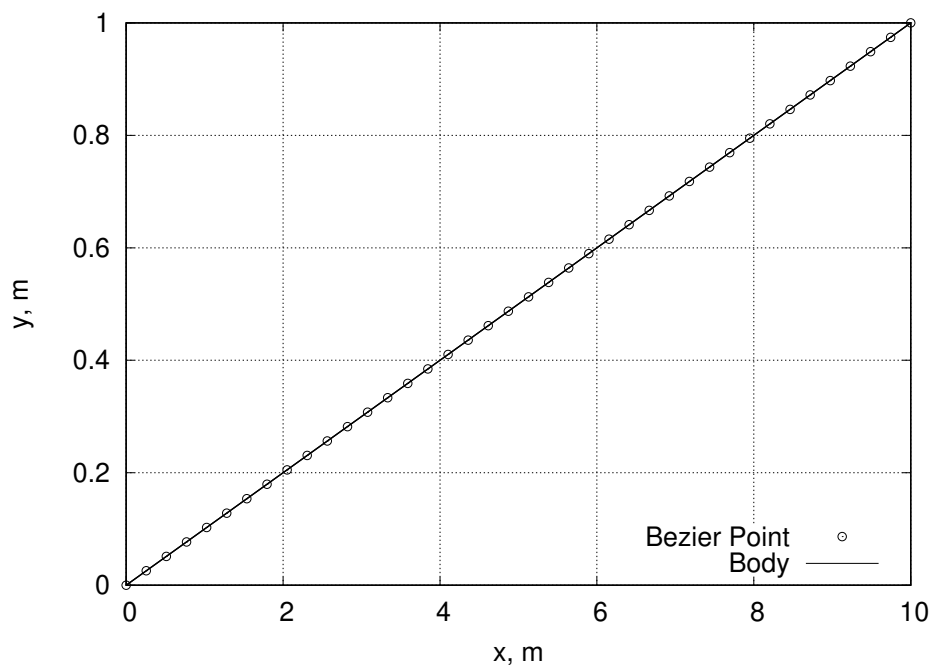
Table D.2: 40-point Bézier coordinates for minimum-drag slender body of revolution (in meters).

Baseline		Optimised	
x-coordinate	y-coordinate	x-coordinate	y-coordinate
0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
2.5641e-01	2.5641e-02	2.5641e-01	1.2073e-01
5.1282e-01	5.1282e-02	5.1282e-01	9.7987e-02
7.6923e-01	7.6923e-02	7.6923e-01	1.7574e-01
1.0256e+00	1.0256e-01	1.0256e+00	2.1818e-01
1.2820e+00	1.2820e-01	1.2820e+00	2.4211e-01
1.5384e+00	1.5384e-01	1.5384e+00	2.6684e-01
1.7948e+00	1.7948e-01	1.7948e+00	2.9669e-01
2.0512e+00	2.0512e-01	2.0512e+00	3.2890e-01
2.3076e+00	2.3076e-01	2.3076e+00	3.6012e-01
2.5641e+00	2.5641e-01	2.5641e+00	3.8878e-01
2.8205e+00	2.8205e-01	2.8205e+00	4.1500e-01
3.0769e+00	3.0769e-01	3.0769e+00	4.3967e-01
3.3333e+00	3.3333e-01	3.3333e+00	4.6377e-01
3.5897e+00	3.5897e-01	3.5897e+00	4.8790e-01
3.8461e+00	3.8461e-01	3.8461e+00	5.1222e-01
4.1025e+00	4.1025e-01	4.1025e+00	5.3661e-01
4.3589e+00	4.3589e-01	4.3589e+00	5.6077e-01
4.6153e+00	4.6153e-01	4.6153e+00	5.8447e-01
4.8717e+00	4.8717e-01	4.8717e+00	6.0756e-01
5.1282e+00	5.1282e-01	5.1282e+00	6.3001e-01
5.3846e+00	5.3846e-01	5.3846e+00	6.5193e-01
5.6410e+00	5.6410e-01	5.6410e+00	6.7343e-01
5.8974e+00	5.8974e-01	5.8974e+00	6.9466e-01
6.1538e+00	6.1538e-01	6.1538e+00	7.1570e-01
6.4102e+00	6.4102e-01	6.4102e+00	7.3657e-01
6.6666e+00	6.6666e-01	6.6666e+00	7.5725e-01
6.9230e+00	6.9230e-01	6.9230e+00	7.7765e-01
7.1794e+00	7.1794e-01	7.1794e+00	7.9771e-01
7.4358e+00	7.4358e-01	7.4358e+00	8.1740e-01
7.6923e+00	7.6923e-01	7.6923e+00	8.3673e-01
7.9487e+00	7.9487e-01	7.9487e+00	8.5575e-01
8.2051e+00	8.2051e-01	8.2051e+00	8.7452e-01
8.4615e+00	8.4615e-01	8.4615e+00	8.9302e-01
8.7179e+00	8.7179e-01	8.7179e+00	9.1119e-01
8.9743e+00	8.9743e-01	8.9743e+00	9.2907e-01
9.2307e+00	9.2307e-01	9.2307e+00	9.4701e-01
9.4871e+00	9.4871e-01	9.4871e+00	9.6518e-01
9.7435e+00	9.7435e-01	9.7435e+00	9.8201e-01
1.0000e+01	1.0000e+00	1.0000e+01	1.0000e+00

APPENDIX D. MINIMUM-DRAG SLENDER BODY OF REVOLUTION BASELINE AND OPTIMISED GEOMETRIES



(a)



(b)

Figure D.1: Minimum-drag slender body of revolution Bézier parameterisation (a) 20-point curve (b) 40-point curve.

APPENDIX E

P2 INLET BASELINE AND OPTIMISED GEOMETRIES

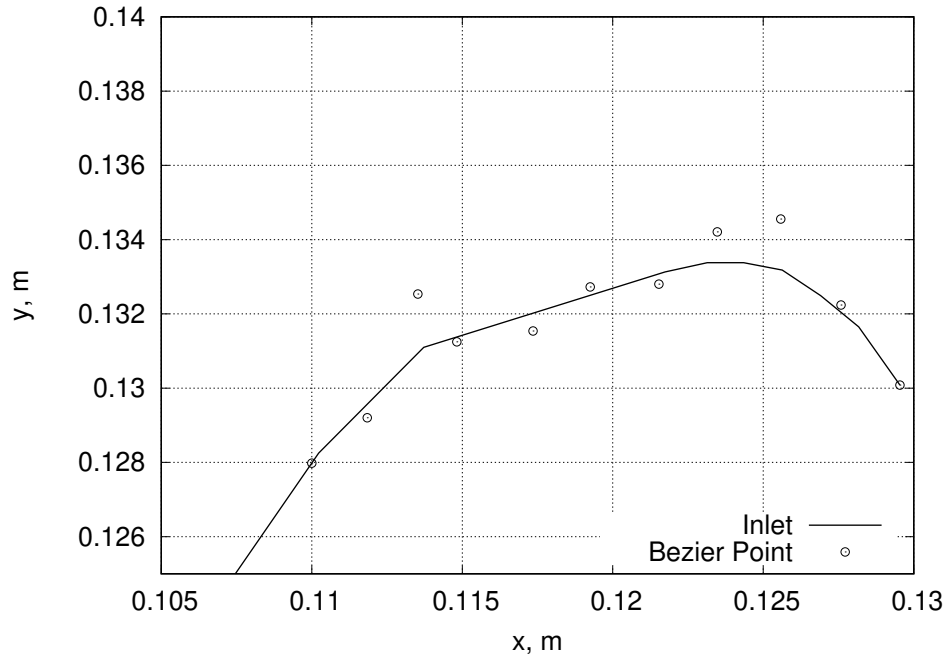
Geometric details for the baseline and optimised P2 inlet are provided below.

Table E.1: 11-point Bézier coordinates for P2 inlet (in meters).

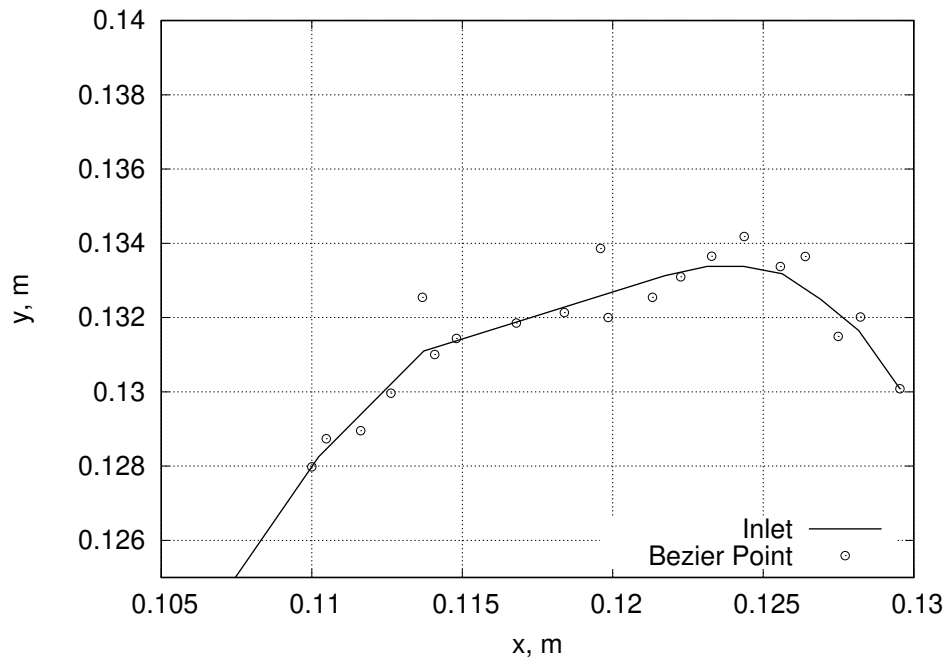
Baseline		Optimised	
x-coordinate	y-coordinate	x-coordinate	y-coordinate
1.1000e+00	1.2798e-01	1.1000e+00	1.279e-01
1.1184e+00	1.2920e-01	1.1184e+00	1.292e-01
1.1352e+00	1.3253e-01	1.1352e+00	1.3637e-01
1.1482e+00	1.3124e-01	1.1482e+00	1.3543e-01
1.1734e+00	1.3153e-01	1.1734e+00	1.3240e-01
1.1925e+00	1.3272e-01	1.1925e+00	1.3307e-01
1.2153e+00	1.3280e-01	1.2153e+00	1.3485e-01
1.2347e+00	1.3420e-01	1.2347e+00	1.3665e-01
1.2558e+00	1.3455e-01	1.2558e+00	1.3589e-01
1.2758e+00	1.3223e-01	1.2758e+00	1.3261e-01
1.2954e+00	1.3008e-01	1.2954e+00	1.3008e-01

Table E.2: 20-point Bézier coordinates for P2 inlet (in meters).

Baseline		Optimised	
x-coordinate	y-coordinate	x-coordinate	y-coordinate
1.1000e+00	1.2798e-01	1.1000e+00	1.2798e-01
1.1048e+00	1.2873e-01	1.1048e+00	1.2873e-01
1.1162e+00	1.2895e-01	1.1162e+00	1.2919e-01
1.1263e+00	1.2996e-01	1.1263e+00	1.3171e-01
1.1367e+00	1.3254e-01	1.1367e+00	1.3560e-01
1.1480e+00	1.3143e-01	1.1480e+00	1.3484e-01
1.1408e+00	1.3100e-01	1.1408e+00	1.3373e-01
1.1679e+00	1.3180e-01	1.1679e+00	1.3357e-01
1.1839e+00	1.3213e-01	1.1839e+00	1.3334e-01
1.1984e+00	1.3200e-01	1.1984e+00	1.3339e-01
1.1959e+00	1.3386e-01	1.1959e+00	1.3565e-01
1.2132e+00	1.3254e-01	1.2132e+00	1.3442e-01
1.2225e+00	1.3309e-01	1.2225e+00	1.3459e-01
1.2328e+00	1.3365e-01	1.2328e+00	1.3456e-01
1.2436e+00	1.3418e-01	1.2436e+00	1.3461e-01
1.2556e+00	1.3337e-01	1.2556e+00	1.3352e-01
1.2639e+00	1.3364e-01	1.2639e+00	1.3368e-01
1.2748e+00	1.3149e-01	1.2748e+00	1.3149e-01
1.2822e+00	1.3201e-01	1.2822e+00	1.3201e-01
1.2954e+00	1.3008e-01	1.2954e+00	1.3008e-01



(a)



(b)

Figure E.1: P2 inlet Bézier parameterisation (a) 11-point curve (b) 20-point curve.