

NUMERICAL SIMULATION OF WEAKLY IONIZED HYPERSONIC FLOW OVER REENTRY CAPSULES

by

Leonardo C. Scalabrin

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in The University of Michigan
2007

Doctoral Committee:

Professor Iain D. Boyd, Chairperson
Professor Bram van Leer
Professor Kenneth G. Powell
Associate Professor Hong G. Im

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.



UMI Microform 3287624

Copyright 2008 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

© Leonardo C. Scalabrin 2007
All Rights Reserved

To my loved wife, Ester.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Iain Boyd for his support and guidance through this project. I am extremely thankful for his positive attitude even when things were not going so well. I also would like to thank Dr. David Hash at NASA Ames Research Center who masterfully navigated through a bureaucratic maze so that I could have a very productive internship at Ames which also fulfilled a childhood dream of working at NASA.

I would like to thank the Hypersonic CFD community that provided me help and reading material through emails or personally: Prof. Bram van Leer at the University of Michigan, Dr. Michael Wright and Dr. Dinesh Prabhu at NASA Ames, Dr. Peter Gnoffo and Dr. Brian Hollis at NASA Langley, Prof. Graham Candler, Dr. Ioannis Nompelis and graduate students Mike Barnhardt and Travis Drayna at the University of Minnesota.

The friendship and encouragement from fellow graduate students at Michigan, including Jeff Sutton, Chad Rasmussen, Justin Koo, Lenny Talerico, Yoshi Suzuki, Shintaro Yamamoto, Sven Peth, Jesse Linnell, Zac Nagel, Sulabh Dhanuka, Andrew Lofthouse, Andrew Porwitzky, Jose Padilla, John Yim, Jon Burt, Nick Bisek, Peter Hamlington, Tim Deschenes and Tom Schwartzenruber are also deeply appreciated as well as the support from the staff in the Department of Aerospace Engineering, in particular Margaret Fillion, Denise Phelps, Suzanne Smith and David Mclean. I am also thankful for the friendship of Woody Kellum, who supervised me as a database

programmer in the Computer Aided Engineering Network (CAEN).

Finally, I would like to thank my loved wife Ester for her support during this long journey that also included the birth of our two kids, Isabella and Giovanni. I also would like to thank my parents Rosane and Sergio and my brother Marcelo for their unconditional support and love and my parents-in-law Ilza and Osiris and brother-in-law Davi who always helped Ester and I when things were difficult. The help from family friends Sam and Carla Williamson, Mark and Mary Jane Vanderput and Jose and Denise Vasconcelos was very important too.

This work was primarily supported by the Francois-Xavier Bagnoud Foundation and by the Rackham School of Graduate Studies which provided fellowships and travel grants. Additional funding was provided by the Space Vehicle Technology Institute, under NASA grant NCC3-989 with joint sponsorship from the Department of Defense, and by the Education Associates program at NASA Ames.

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	vii
LIST OF TABLES	xi
LIST OF APPENDICES	xii
CHAPTER	
I. INTRODUCTION	1
1.1 Motivation	1
1.2 Challenges of Numerical Simulations of Entry Flows	4
1.3 Review of Related Work	9
1.4 Scope of Present Work	12
II. MATHEMATICAL FORMULATION	14
2.1 Introduction	14
2.2 Conservation Equations	14
2.3 Viscous Terms	17
2.4 Thermodynamic Properties	18
2.5 Transport Properties	21
2.5.1 Wilke/Blottner/Eucken	21
2.5.2 Gupta/Collision cross section data	22
2.6 Source terms	26
2.6.1 Chemical Model	26
2.6.2 Relaxation Model	29
III. NUMERICAL METHOD	33
3.1 Introduction	33
3.2 Spatial Integration and Flux Calculations	34

3.3	Time Integration	37
3.3.1	Point Solver	40
3.3.2	Line Solver	43
3.4	Boundary Conditions	45
3.4.1	Inviscid Flux Boundary Condition	46
3.4.2	Viscous Boundary Condition	48
3.5	Implicit Boundary Conditions	51
3.5.1	Inviscid Matrices	52
3.5.2	Viscous Matrices	53
3.6	Higher order extension of inviscid fluxes	54
	IV. IMPLEMENTATION ASPECTS	59
4.1	Introduction	59
4.2	Code Overview	59
4.3	Line Finding Algorithm	62
4.4	Mesh Partitioning	68
4.4.1	Inter-processor Boundary Conditions	72
4.4.2	Speed-Up Measurements	75
	V. RESULTS	78
5.1	Introduction	78
5.2	Apollo Experimental Model	79
5.3	Mars Entry Spacecraft Model	86
5.3.1	Air Flow	87
5.3.2	Carbon Dioxide Flow	95
5.4	RAM-C Spacecraft	100
5.5	FIRE-II Spacecraft	107
5.5.1	Convective Heating	109
5.5.2	Radiative Heating Sensitivity Analysis	115
	VI. SUMMARY AND CONCLUSIONS	128
6.1	Summary	128
6.2	Conclusions	131
6.3	Future Research	134
	APPENDICES	139
	BIBLIOGRAPHY	174

LIST OF FIGURES

1.1	Some successful designs of space capsules.	3
1.2	Re-entry trajectory of the Apollo-6 mission.	6
1.3	Schematic of the fluid flow over a reentry capsule.	7
1.4	Typical failure of numerical flux functions in the simulation of hypersonic flow using unstructured grids.	9
3.1	Unstructured grid example.	39
3.2	Typical sparse linear system for unstructured grids.	40
3.3	Iterative solution of linear system using point implicit method. . . .	40
3.4	Face orientation.	41
3.5	Unstructured grid renumbering.	44
3.6	Full linear system.	44
3.7	Iterative solution of linear system using line implicit method. . . .	46
3.8	2nd order stencil for a structured grid.	55
3.9	Stencil search approach for 2nd order calculations.	57
3.10	Second-order stencil search.	58
4.1	Lines creation process.	64
4.2	Lines creation process - cont.	64
4.3	Lines creation process - cont.	65
4.4	Lines creation process - cont.	65
4.5	Lines creation process - concluded.	65
4.6	Lines created in a hybrid two-dimensional grid composed of triangles and quadrilaterals.	66
4.7	Detail of the lines created in a hybrid two-dimensional grid composed of triangles and quadrilaterals.	67
4.8	Lines created in a two-dimensional mesh composed of quadrilaterals and treated as an unstructured grid.	67
4.9	Convergence to steady state for a $M = 20$ flow over a blunt cone. .	68
4.10	Unstructured grid and corresponding graph used in METIS.	70
4.11	Mesh partitions.	70

4.12	Modified logical grid and corresponding graph.	71
4.13	Mesh partitions.	71
4.14	Mesh partitioning.	73
4.15	Cells used in inter-processor boundary conditions in different parts of the code.	74
4.16	Property contours unaffected by mesh partitioning.	75
4.17	Computational speed-up of the numerical code in different parallel machines.	76
5.1	Apollo experimental model geometry.	80
5.2	Mesh used for angle-of-attacks from 180 to 165 degrees.	81
5.3	Mesh used for angle-of-attacks from 160 to 150 degrees.	81
5.4	Detail of the surface mesh close to the symmetry point.	82
5.5	X-velocity component contours for $\alpha = 150$ degrees.	83
5.6	Temperature contours for $\alpha = 180$ degrees.	83
5.7	Axial force coefficient.	84
5.8	Normal force coefficient.	85
5.9	Pitching moment coefficient.	85
5.10	Mars entry spacecraft model geometry.	87
5.11	Block structure used for generating the grid over the Mars entry spacecraft model.	89
5.12	Mesh used in the simulations of a Mars entry spacecraft model.	89
5.13	Convective heat transfer rate along a Mars entry spacecraft model.	90
5.14	X-velocity contours and streamlines over the model.	91
5.15	Ratio of translational-rotational to vibrational temperatures.	91
5.16	Translational and vibrational temperatures along the stagnation streamline of the Mars entry experimental model.	92
5.17	Mass fractions along the stagnation streamline of the Mars entry experimental model.	93
5.18	Heat transfer on the model forebody.	94
5.19	Heat transfer on the model afterbody.	95
5.20	Temperature countours and streamlines over the model afterbody.	96
5.21	Computational grid used for CO ₂ flow over the Mars entry experimental model forebody.	98
5.22	X-velocity contours over the Mars entry experimental model forebody for CO ₂ flow.	99
5.23	Temperature profiles along the stagnation streamline for CO ₂ flow.	99
5.24	Species mass fractions profiles along the stagnation streamline for CO ₂ flow.	100

5.25	Heat transfer along the Mars entry experimental model forebody for CO ₂ flow.	101
5.26	RAM-C geometry.	102
5.27	Computational grid used for the RAM-C II spacecraft at 71 km altitude simulations.	103
5.28	Convective heat transfer rate along the RAM-C II spacecraft.	104
5.29	Electron number density comparisons along the body for RAM-C II flight test at 71 km altitude.	105
5.30	Translational-rotational temperature over the RAM-C II spacecraft at 71 km altitude using Park's 1990 chemistry set.	106
5.31	Influence of Park's high temperature correction on temperature profiles along the stagnation streamline for the RAM-C II flight test at 71 km altitude.	107
5.32	FIRE-II geometry.	109
5.33	Mesh used by DPLR and LeMANS for the 1636 seconds, 11-species cases.	111
5.34	Temperature contours around the FIRE-II spacecraft for the 1636 seconds, 11-species, non-catalytic wall case.	112
5.35	Pressure contours around the FIRE-II spacecraft for the 1636 seconds, 11-species, non-catalytic wall case.	112
5.36	Species mass fractions along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case.	113
5.37	Convective heat transfer along the wall for the 1636 seconds, 11-species, non-catalytic wall case.	114
5.38	Translational temperature profiles along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case.	115
5.39	Mesh used by LAURA for the 1636 seconds, 11-species cases.	116
5.40	Vibrational temperature profiles along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case.	116
5.41	Electron number density along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case calculated using different codes.	119
5.42	Atomic nitrogen number density along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case calculated using different codes.	119
5.43	Atomic oxygen number density along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case calculated using different codes.	120
5.44	Translational temperature profiles along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case.	122

5.45	Vibrational temperature profiles along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case.	123
5.46	Electron number density along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case calculated using different thermal non-equilibrium models.	123
5.47	Atomic nitrogen number density along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case calculated using different thermal non-equilibrium models.	124
5.48	Atomic oxygen number density along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case calculated using different thermal non-equilibrium models.	124

LIST OF TABLES

5.1	Freestream Conditions.	79
5.2	Conditions for the Mars entry spacecraft experimental model air-flow case.	88
5.3	Conditions for the Mars entry spacecraft experimental model carbon dioxide-flow case.	96
5.4	Vibrational energy data for CO ₂	97
5.5	Freestream Conditions.	102
5.6	Freestream Conditions for the FIRE-II cases.	108
5.7	FIRE-II Convective Heating.	110
5.8	Stagnation point radiative heating calculations using different codes.	118
5.9	Stagnation point radiative heating sensitivity analysis for Non-Catalytic wall.	121
A.1	Species Chemistry Data	140
A.2	Electronic Energy Levels.	141
A.3	Electronic Energy Levels - continued.	142
A.4	Electronic energy levels - continued.	143
A.5	Lewis Research Center Curve fits.	144
A.6	$\Omega_{0,0}$ integrals for collisions involving neutrals only.	145
A.7	$\Omega_{0,0}$ integrals for collisions involving neutrals only - continued. . . .	146
A.8	$\Omega_{1,1}$ integrals for collisions involving neutrals only.	146
A.9	$\Omega_{1,1}$ integrals for collisions involving neutrals only - continued. . . .	147
A.10	Curve fits for shielded coulomb potential.	147
A.11	Reaction data used in the DPLR code.	148
A.12	Reaction data used in the LAURA code.	149
A.13	Equilibrium Constants Curve Fits for Park's 90 model.	150
A.14	Equilibrium Constants Curve Fits for Park's 90 model - continued. .	151
A.15	Equilibrium Constants Curve Fits for Park's 90 model - continued. .	152
A.16	Modified Millikan and White's Parameters.	153
A.17	Modified Millikan and White's Parameters - continued.	154

LIST OF APPENDICES

A.	Chemistry Sets	140
A.1	Species data	140
A.2	Electronic energy data	140
A.3	Species thermodynamic curve fits	143
A.4	Collision cross section data	145
A.5	Reactions data	148
A.6	Equilibrium constant data	150
A.7	Modified Millikan and White coefficients	153
B.	Jacobian Matrices	155
B.1	Inviscid Jacobians	155
B.2	True Inviscid Jacobians	156
B.3	Viscous Jacobians	162
B.3.1	Implicit Boundary Conditions	165
B.4	Chemistry Jacobians	166
B.5	Non-Equilibrium Jacobians	168
C.	Axisymmetric Formulation and Axisymmetric Jacobian	170

CHAPTER I

INTRODUCTION

1.1 Motivation

The aging of the Space Shuttle prompted significant research on possible designs for its replacement through the years. For a long time it was believed that the replacement for this reusable vehicle would be a spacecraft more advanced, more efficient and of course, more reusable. One of the paradigms that received a lot of attention in the past years was air-breathing reusable vehicles. Such vehicles would take off from a runway and accelerate to orbital velocities using a series of different engines. The technology necessary for these spacecraft is still being developed and an operational vehicle is still many years away.

The tragic accident of the Space Shuttle Columbia in 2003 made it very clear that the US Space Program could not wait for the development of air-breathing vehicles to substitute the Space Shuttle. Due to a decision to retire the Space Shuttle in a short time, a space capsule design was selected as an immediate replacement to transport astronauts between the Earth and the International Space Station and maybe beyond. The selection of space capsules is based on their historic safety as demonstrated by the US, Russia and China, their low cost of maintenance and production and also because the technology is immediately available.

The US Space Program used manned space capsules for the Mercury, Gemini and Apollo programs and smaller scale unmanned space capsules for several missions to other planets. Russia still uses the Soyuz capsule for manned missions and China uses a capsule for its manned flights that is very similar to the Soyuz. Some typical capsules are shown in Fig. 1.1. In the US, the focus on the Space Shuttle program did not allow resources to be used in the design of new space capsules for manned flight. Therefore, there is room for improvement of existing designs and for the development of shapes that would be optimal in some design sense such as minimizing heating, minimizing communication blackout or increasing aerodynamic stability.

Numerical simulation of fluid flows is an integral part in the design of any vehicle, including entry space capsules. The extreme conditions experienced by the spacecraft during reentry are difficult to reproduce in an experimental facility and flight tests on other planets are not possible. The cost of experimental studies of hypersonic flows is higher than low speed flows because of the immense amount of energy that has to be used to reproduce flight conditions and because the number of facilities equipped to perform such experiments is very limited. A flight test in Earth's atmosphere is even more expensive. The application of numerical tools such as Computational Fluid Dynamics (CFD) can reduce the number of costly experiments by helping engineers achieve preliminary designs that are much closer to the final vehicle. Indeed, CFD has been used extensively in the past two decades as a design tool of entry capsules such as Galileo[64], Mars Pathfinder[19, 27], Stardust[68, 69, 70] and others. The increase in computing power allowed the simulation of more physical processes involved. In the ideal case, experimental and flight tests in the Earth's atmosphere can become a certification phase of the design developed using numerical tools. Currently, CFD is being used in the design of the next US manned capsule, Orion.

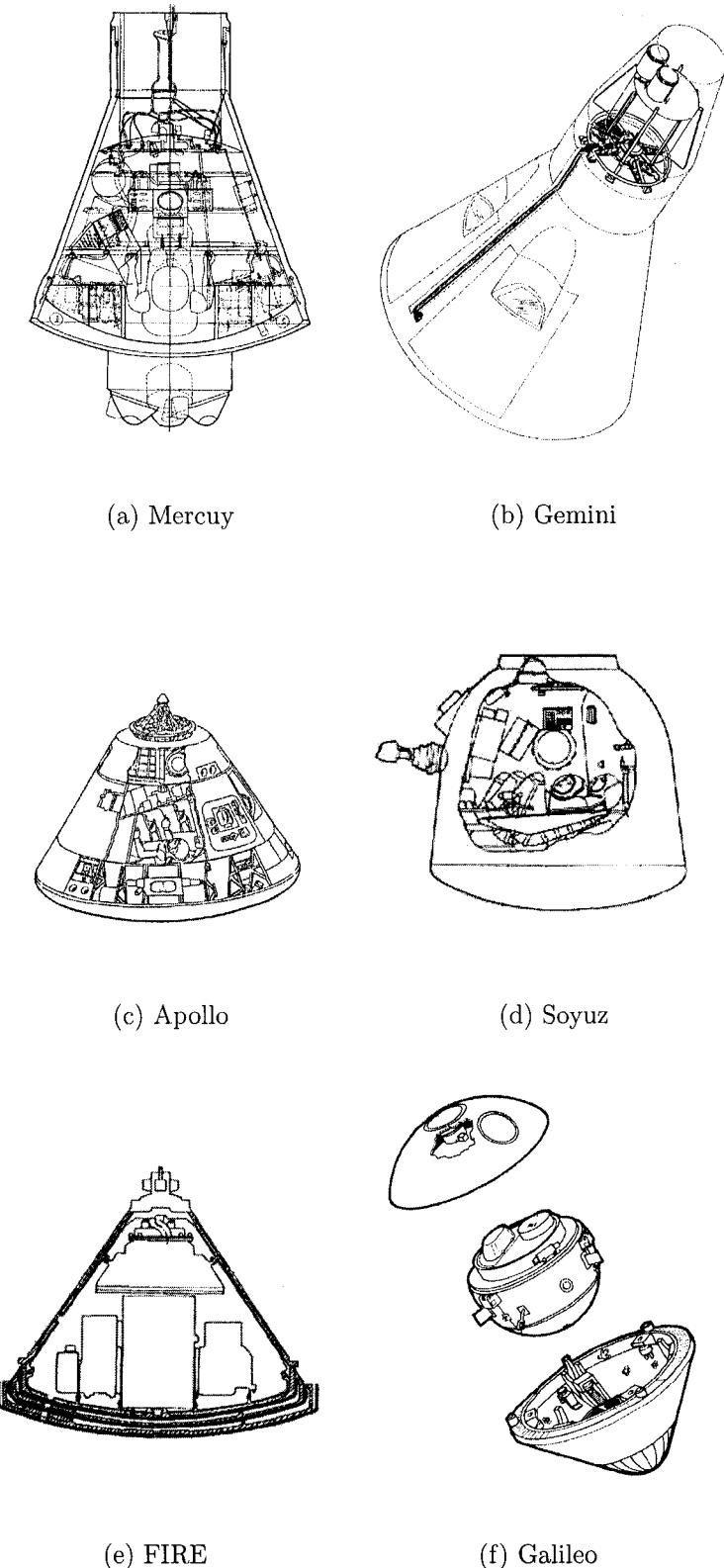


Figure 1.1: Some successful designs of space capsules.

Due to extreme flow velocities, the numerical simulation of entry flows involves the modeling of various physical phenomena not routinely included in most numerical tools for flow analysis. It is common for entry flows to be in thermo-chemical non-equilibrium, ionized, and electronically excited. Some of the physical modeling is approximate and much of it has not changed since the 1980's. In addition, the numerical modeling of hypersonic flows presents distinct problems that have not been solved to date. Old models can be improved or replaced to allow more accurate simulation of entry flows used in the design of new space capsules and other hypersonic spacecraft including air breathing vehicles. The influence of the physical and numerical models can be tested against available experimental and flight data in a simulation that includes all the physical modeling required for meaningful comparisons. The Apollo program in the 1960's generated a large amount of experimental and flight test data that can be used for such studies. Because Orion will likely be similar to the Apollo capsule, there is a lot of interest in knowing how the current physical and numerical modeling perform in relation to these data.

1.2 Challenges of Numerical Simulations of Entry Flows

Typical values for the entry velocity of a spacecraft in a planetary atmosphere range between 7 and 12 km/s and the associated Mach numbers range between 20 and 50. Figure 1.2 shows the reentry trajectory of the Apollo 6 capsule in the Earth's atmosphere. As the spacecraft crosses the atmosphere, different flow regimes ensue. For altitudes larger than 120 km, the Knudsen number, which measures the ratio of the distance between gas particles or mean free path to the spacecraft characteristic length (around 4 meters for the Apollo capsule), can reach values above 10. In this flow regime, called free-molecular, collisions between gas particles that

are reflected at the spacecraft wall with the incoming particles are so rare that they can be neglected and no shock wave is observed. Between 120 and 90 kilometers of altitude, the Knudsen number is between 1 and 0.01, as can be seen in Fig. 1.2(a), and the flow regime is called transitional. The collisions between the reflected particles at the wall and the incoming particles are frequent leading to the formation of a diffuse bow shock wave in front of the body. During the free molecular and the transitional regime, the well known Navier-Stokes equations for fluid flow are not valid and the flow has to be modeled using different sets of equations such as the Burnett equations or the Boltzmann equation. The Boltzmann equation can be simulated numerically using the Direct Simulation Monte Carlo (DSMC) method[20]. The atmosphere is so rarefied during these regimes that almost no reduction in speed is observed and the heating rates are small as shown in Fig. 1.2(b). In this figure, the convective heating rates are estimated using the Fay and Riddell expression[11] in which the stagnation point heating rate is proportional to $\rho_{\infty}^{1/2} u_{\infty}^3$, where ρ_{∞} is the freestream density and u_{∞} is the freestream speed. The plotted results are normalized by the maximum heating rate calculated.

For altitudes below 90 km, the Knudsen number is smaller than 0.01 and the flow regime is called continuum. Collisions between the reflected particles and the incoming flow are so frequent that the bow shock wave becomes thin. The number density of particles in the gas is large enough so that the gas can be modeled as a continuum, except in localized regions such as inside the shock wave or in some regions of the rarefied wake. These regions are shown in Fig. 1.3 which presents a schematic view of continuum flow around the capsule. The local characteristic length of those regions can be of the same order of the mean free path thus invalidating the continuum approximation. These regions can be successfully simulated using the

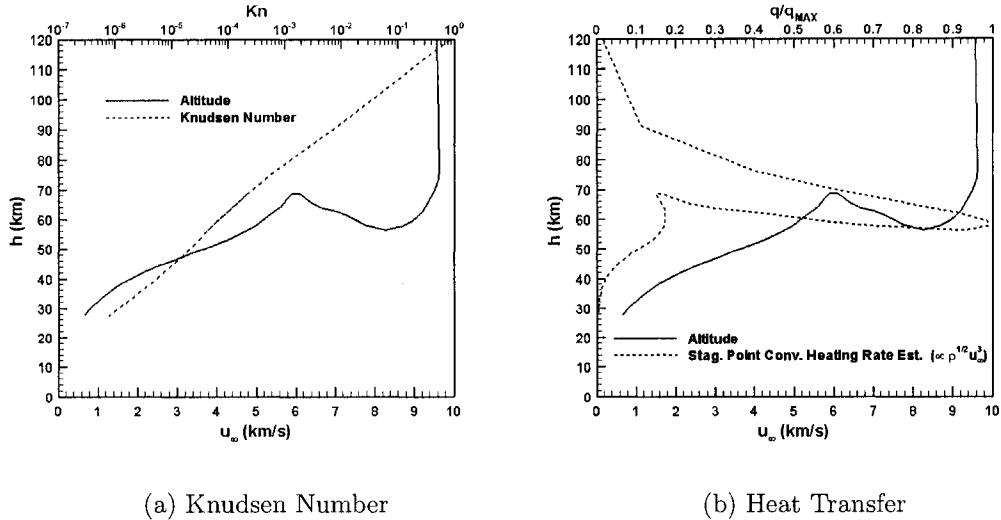


Figure 1.2: Re-entry trajectory of the Apollo-6 mission.

DSMC method. At an altitude around 80 km, the spacecraft is already experiencing significant reduction in speed. The immense amount of kinetic energy is transformed into thermal energy through the shock wave generating temperatures as high as 50,000 K immediately after the shock. Most of this thermal energy stays with the flow and is convected away but part of it is transferred to the spacecraft by convective and radiative heating. The maximum convective heating rate at the stagnation point usually occurs between 70 and 60 km altitude, well within the limits of continuum modeling, as can be seen in Fig. 1.2(b).

Due to the very high speeds and relatively low density of the flow, the shock stand-off distance at the end of the transitional regime and at the beginning of the continuum regime is small. During this period, the shock standoff distance and the boundary layer thickness can be approximately the same giving rise to what is called the shock layer. It is common in hypersonic flow to reference the region between the wall and the shock wave as the shock layer even when the shock wave is not at the boundary layer edge. The shock layer is indicated in Fig. 1.3. The high temperature

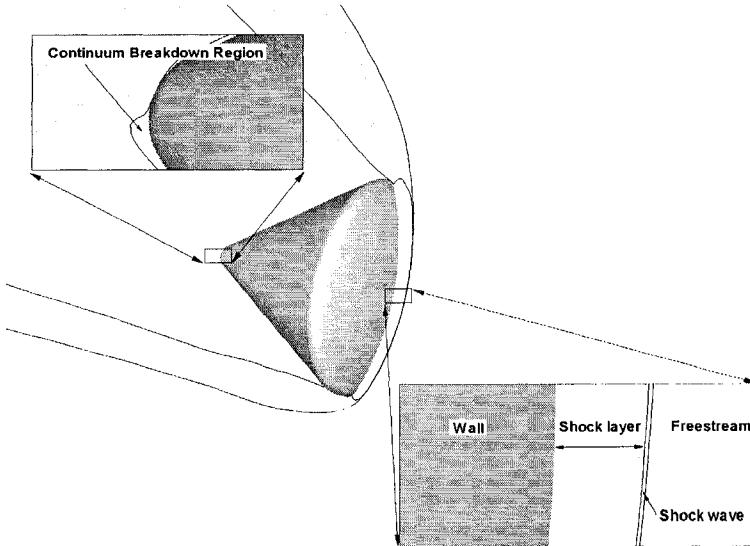


Figure 1.3: Schematic of the fluid flow over a reentry capsule.

in the shock layer due to the shock causes intense chemical reactions, vibrational and electronic energy excitation and gas ionization. The chemical reactions and the energy transfer processes between the different energy modes of the gas occur at finite rates that depend on intermolecular collisions which in turn depend on the gas density. Because of the associated high speeds and relatively low density, the flow characteristic time may be of the same order of the chemical and energy transfer processes characteristic times, leading to thermal and chemical non-equilibrium inside the shock layer. As the spacecraft descends in the atmosphere the rate of intermolecular collisions increases and the flow will eventually achieve thermal and chemical equilibrium at low altitudes.

The presence of an ionized and electronically excited gas causes the shock layer to emit radiation which is partly absorbed by the spacecraft as radiative heating. In addition, the spacecraft wall can be a catalyst to chemical reactions which leads to exothermic reactions close to the wall and additional heating. The large heating load in entry vehicles can be handled by ablation of material from the heat shield, which

will inject different chemical species in the flow. All these physical processes and the fluid flow are coupled together. The chemical non-equilibrium affects flow temperature, flow composition and flow properties such as viscosity and heat conduction. The thermal non-equilibrium affects the reaction rates. The emitted radiation affects the flow field by lowering the flow temperature. The amount of radiation depends on the thermal non-equilibrium of the flow field. The ablation of the heat shield leads to different chemical reactions close to the wall and can change the shape of the spacecraft. The injection of material due to ablation can make the boundary layer turbulent. The efficient simulation of hypersonic flows should take into account the interdependence of the physical phenomena.

Although the physical modeling difficulties are formidable, the numerical problems incurred by the numerical modeling can not be disregarded. The extremely high Mach numbers cause problems for most numerical flux functions and numerical tools. These problems are even more pronounced when using unstructured grids made of triangles or tetrahedra where the grid can not be aligned to the shock wave. One example of such a computation is shown in Fig. 1.4, which shows contours of x-velocity for a $M = 20$ flow over a cylinder obtained using a very fine mesh made of triangles. The staircase pattern emanating from the shock is generated by almost all flux functions currently used for compressible flow simulations. These problems at the shock wave can be minimized by using carefully generated grids of quadrilaterals and hexahedra that are aligned to the shock wave at the expense of grid generation time.

Moreover, the presence of chemical reactions requires very fine grids at the wall resulting in more expensive simulations. The potentially large number of chemical species also increases the computational cost. Most space capsules are flown at some

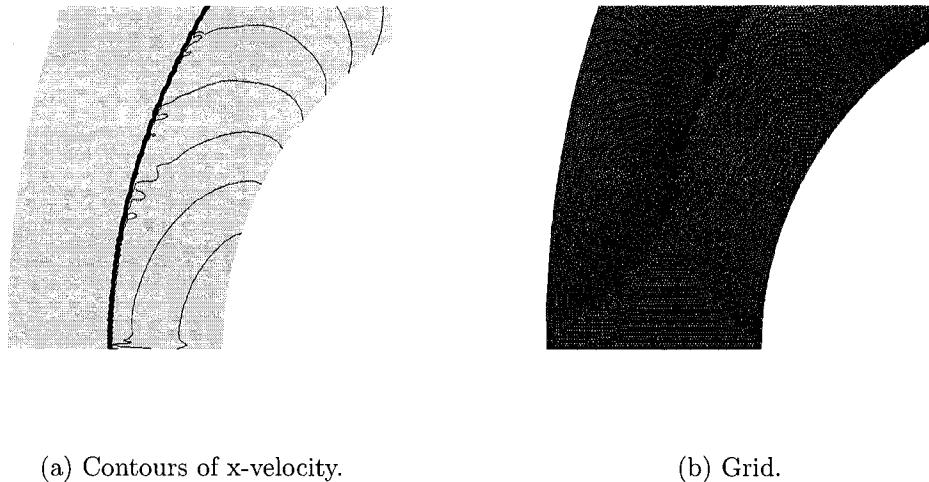


Figure 1.4: Typical failure of numerical flux functions in the simulation of hypersonic flow using unstructured grids.

angle-of-attack to create lift so that the maximum heating occurs at high altitudes which is beneficial because the maximum heating is proportional to the gas density. The simulation of such capsules requires expensive three-dimensional computations because of the angle-of-attack. In addition, most capsules have attachments such as antennas, windows and ports that may present localized heat load. Such small structures in the geometry result in difficult grid generation for detailed simulations.

1.3 Review of Related Work

The numerical simulation of fluid flows for aerospace applications has been performed for decades. It started with the numerical simulation of simplified forms of the Navier-Stokes equations, such as the linearized potential flow, the transonic small disturbance equations and the full potential equations[40]. The first simulations of hypersonic flows solved a simplification of the Navier-Stokes equations called the viscous shock-layer equations. Such simulations included analysis of the Galileo probe[64], of the Space Shuttle[47] and of aerobrake configurations[55]. Most of the

ideas that define modern Computational Fluid Dynamics (CFD) and that are currently used, such as flux vector splitting[86, 50], approximate Riemann solvers[79, 78] and methods for robust higher order approximations[51, 52] were developed around this time as well. These methods allow the simulation of the full Navier-Stokes equations for general geometries and more general flow conditions and are significantly more expensive than previous methods.

As the numerical modeling improved and computer power increased, more complex models for the physical phenomena were gradually included in the simulations of the Navier-Stokes equations. First attempts in the simulation of hypersonic flows using modern CFD loosely coupled chemistry source terms[25] or coupled a simplified chemistry model[22]. The first computation of a multi-dimensional flowfield including thermo chemical nonequilibrium and ionization was performed by Candler[18]. In his work, Candler solved using modern Computational Fluid Dynamics techniques a set of equations that coupled thermal and chemical non-equilibrium equations to the fluid dynamics equations. The set of equations solved is similar to the set developed by Lee[49] but considered a different vibrational temperature for each polyatomic species.

Around the same time, another CFD code for entry flows called LAURA (Langley Aerothermodynamic Upwind Relaxation Algorithm) was developed by Gnoffo[24] at NASA Langley Research Center. That code solves the set of equations developed by Lee[49] because the energy transfer between vibrational modes is usually fast. This numerical tool was used extensively in the design of several entry capsules[28] and for analysis of the Space Shuttle thermal protection system[26]. Both numerical codes used implicit methods, structured grids and took advantage of vector machines at the time.

The increased availability of parallel machines in the 1990's made the development of implicit algorithms amenable to parallelization necessary. Wright[95] showed that the linear system of equations originating from the implicit integration of fluid dynamics problems using structured grids can be efficiently solved by resolving the system of equations exactly in the direction normal to the body and relaxing it in the other directions. This approach is amenable to parallelization as long as the structured grid is partitioned along directions normal to the body. The code called DPLR (Data-Parallel Line Relaxation) is currently used at NASA Ames Research Center to design heat shields of reentry capsules[28].

On a different front, other researchers[88] developed implicit algorithms for CFD codes using unstructured grids. Unstructured grids are not limited to quadrilaterals or hexahedrons and offer more flexibility for grid generation. The algorithms developed at the time were mainly based on the solution of a sparse system of equations and did not use physical properties of the problem. The algorithms developed showed stability and convergence problems for ideal gas calculations. Nompelis[67] showed that the algorithm developed by Wright[95] is also very efficient for the simulation of hypersonic flows using unstructured grids requiring only that normal lines be assembled in the unstructured mesh, an approach that has been used extensively in the finite-element community[39, 61]. The algorithm developed was restricted to the unstructured treatment of three-dimensional meshes made of hexahedra but the extension to unstructured grids made of tetrahedra or two-dimensional grids is trivial.

Furthermore, the constant increase in computer power has allowed in recent years the use of more complex models for chemical kinetics and wall catalysis[74, 76, 75], transport properties[34, 72, 96] and diffusion models[87, 75, 29]. Coupled solutions

involving radiation[71] and ablation[38] have been performed but are still not common.

1.4 Scope of Present Work

The goal of this work is to develop and validate a multi-dimensional CFD code for the simulation of weakly ionized hypersonic flows in thermo-chemical non-equilibrium around entry configurations using unstructured grids. The code is to be used in the future by other researchers to analyze and improve the physical and numerical modeling of hypersonic flows to ultimately improve the design of space capsules. Current physical and numerical modeling are used in the code. While the physical and numerical modeling used is not new, the code itself is completely developed outside established research groups that perform entry flow simulations using CFD. The code is called “Le” Michigan Aerothermodynamic Navier-Stokes Solver (LeMANS). LeMANS is currently used as the CFD module in a hybrid DSMC-CFD code that is being developed[82, 84, 83] to simulate flows that include localized regions where the continuum approximation is not valid. The code uses unstructured grids to facilitate the coupling with the DSMC solver and to allow future investigation of the numerical problems caused by unstructured grids. In this work, only results with grids made of quadrilaterals and hexahedra are presented as part of the validation process of the code. LeMANS is also currently being used to perform evaluation of physical models employed in entry flow simulations[59, 58, 82, 84, 83], in the modeling of the flowfield typical of plasma guns[77] and in benchmark studies against other codes for entry flows[81] that aim at reducing uncertainties in aerothermodynamic calculations.

This dissertation is organized as follows: Chapter 2 presents the set of partial differential equations and associated physical models used to describe a weakly ion-

ized entry flow in thermo-chemical non-equilibrium. Chapter 3 details the numerical method used in the discretization of the set of partial differential equations using unstructured grids and the implicit algorithms used to solve the resulting sparse linear system. Chapter 4 shows unstructured grid partitioning strategies that are convenient for the implicit algorithm and for the higher order extension of inviscid fluxes. Chapter 5 presents code validation efforts using available experimental, numerical and flight data for space capsules. Conclusions and suggestions for further research are summarized in Chapter 6.

CHAPTER II

MATHEMATICAL FORMULATION

2.1 Introduction

The high speed of hypersonic flows causes many physical phenomena in the flow-field that are not normally modeled by the perfect gas form of the Navier-Stokes equations. Such physical phenomena include chemical non-equilibrium, vibrational and electronic excitation, thermal non-equilibrium and ionization. These phenomena require the use of a real gas model. This chapter presents the extended form of the Navier-Stokes equations used to describe the flow and the thermodynamic and transport property models used to describe the real gas. The finite-rate chemistry model and the energy relaxation model used to describe the chemical and thermal non-equilibrium of the flowfield are also presented.

2.2 Conservation Equations

The flow is modeled assuming that the continuum approximation is valid. It is assumed that the rotational and translational energy modes of all species can be described by a single temperature T_{tr} because the rotational energy equilibrates with the translational energy in just a few collisions. Furthermore, it is assumed that the vibrational and electronic energy modes of all species and the electron translational

energy mode can be described by a single temperature T_{ve} . That assumption is based on the fact that the energy transfer between the translational mode of electrons and the vibrational mode of molecules is very fast in air[24], that the vibrational temperature of different molecules are very similar[18] due to highly efficient vibrational-vibrational energy transfer[49] and that the electronic energy and the electron translational energy can be modeled by a single Maxwell-Boltzmann distribution because the electronic mode is excited by the translational mode of electrons[49]. These assumptions regarding energy transfer simplify the system of equations considerably by eliminating translational and vibrational energy equations for each polyatomic species and an energy equation for the electrons. This is considered a large simplification of the problem but the energy transfer models are not accurate enough to warrant extra complexity in the equations. While these simplifications may not be adequate for some problems, they provide accurate results for aerodynamic coefficients and convective heat transfer rates of reentry configurations[24]. With those approximations, the conservation equations for the three-dimensional system can be written as [49, 24, 74, 18]

$$\frac{\partial Q}{\partial t} + \frac{\partial(E - E_v)}{\partial x} + \frac{\partial(F - F_v)}{\partial y} + \frac{\partial(G - G_v)}{\partial z} = S_{cv} \quad , \quad (2.1)$$

where

$$Q = \begin{Bmatrix} \rho_1 \\ \vdots \\ \rho_{ns} \\ \rho u \\ \rho v \\ \rho w \\ E \\ E_{ve} \end{Bmatrix} \quad \text{and} \quad S_{cv} = \begin{Bmatrix} \dot{w}_1 \\ \vdots \\ \dot{w}_{ns} \\ 0 \\ 0 \\ 0 \\ 0 \\ \dot{w}_v \end{Bmatrix} \quad (2.2)$$

are the vector of conserved variables and the vector of source terms, respectively. In these expressions, $\rho_1 \cdots \rho_{ns}$ are the species densities, u , v and w are the bulk velocity components, E and E_{ve} are the total and the vibrational-electron-electronic energy per unit volume of mixture, respectively.

The inviscid and diffusive flux vector components in the x direction are given by

$$F = \begin{Bmatrix} \rho_1 u \\ \vdots \\ \rho_{ns} u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ (E + p)u \\ E_{ve} u \end{Bmatrix} \quad (2.3)$$

and

$$F_v = \left\{ \begin{array}{c} -J_{x,1} \\ \vdots \\ -J_{x,ns} \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{xx}u + \tau_{xy}v + \tau_{xz}w - (q_{tr,x} + q_{ve,x}) - \sum(J_{x,s}h_s) \\ -q_{ve,x} - \sum(J_{x,s}e_{ve,s}) \end{array} \right\}, \quad (2.4)$$

where p is the pressure, τ_{ij} are the viscous stress components and $q_{tr,i}$ and $q_{ve,i}$ are the translational-rotational and vibrational-electron-electronic heat fluxes in the i -th direction. Moreover, h_s is the species enthalpy and $J_{x,s}$ is the species diffusion flux in the x -direction. The flux vectors in the other directions are similar. The following sections provide more details on the modeling of the terms.

2.3 Viscous Terms

The viscous stresses are modeled assuming a Newtonian fluid for which, using Stokes' hypothesis,

$$\tau_{ij} = \mu \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) + \lambda \nabla \vec{u} \delta_{ij} \quad , \quad \lambda = -\frac{2}{3}\mu \quad , \quad (2.5)$$

where μ is the mixture coefficient of viscosity.

The heat fluxes are modeled according to Fourier's law as

$$\vec{q}_{tr,ve} = -\kappa_{tr,ve} \nabla T_{tr,ve} \quad , \quad (2.6)$$

where $\kappa_{tr,ve}$ are the mixture thermal conductivity for each energy mode. The species mass diffusion fluxes are modeled using Fick's law modified to enforce that the sum

of the diffusion fluxes is zero[87]

$$\vec{J}_{s \neq e} = \vec{I}_s - Y_s \sum_{r \neq e} \vec{I}_r , \quad (2.7)$$

where

$$\vec{I}_s = -\rho D_s \nabla Y_s , \quad (2.8)$$

D_s is the species diffusion coefficients and Y_s is the species mass fraction. This approach is computationally economical and a good approximation for the exact solution of the diffusion fluxes[87] given by the Stefan-Maxwell equations. The electron diffusion flux is not included in the summation in Eq. 2.7 because its value is assumed small due to the small molecular weight of electrons in relation to atoms and molecules. The diffusion flux of electrons is calculated assuming ambipolar diffusion to guarantee the charge neutrality of the flowfield by

$$\vec{J}_e = M_e \sum_{s \neq e} \frac{\vec{J}_s C_s}{M_s} , \quad (2.9)$$

where M_e is the electron molecular weight and C_s is the species charge. The species molecular weight and charge for the 11-species air model used in this work are listed in Appendix A.

2.4 Thermodynamic Properties

The mixture pressure p is obtained using Dalton's law of partial pressures and the perfect gas law for each species. Thus

$$p = \sum_{s \neq e} \rho_s \frac{R_u}{M_s} T_{tr} + \rho_e \frac{R_u}{M_e} T_{ve} , \quad (2.10)$$

where R_u is the universal gas constant. One should observe that the electron temperature is equilibrated with T_{ve} . The total energy per unit volume of mixture is

given by

$$E = \sum_{s \neq e} \rho_s Cv_{tr,s} T_{tr} + \frac{1}{2} \rho (u^2 + v^2 + w^2) + \sum_{s \neq e} \rho_s h_s^o + E_{ve} , \quad (2.11)$$

where h_s^o is the species enthalpy of formation listed in Appendix A for the species used in this work. The translational-rotational specific heat at constant volume, $Cv_{tr,s}$, is given by

$$Cv_{tr,s} = Cv_{t,s} + Cv_{r,s} . \quad (2.12)$$

This work assumes that the translational and the rotational specific heats at constant volume are constant and given by

$$Cv_{t,s} = \frac{3}{2} \frac{R_u}{M_s} \quad (2.13)$$

and

$$Cv_{r,s} = \begin{cases} \frac{R_u}{M_s} & \text{for molecules,} \\ 0 & \text{for atoms and electrons.} \end{cases} \quad (2.14)$$

This is a good approximation because the translational and the rotational energy modes are fully excited even at low temperatures. The species vibrational energy per unit mass is modeled using a harmonic oscillator as

$$e_{v,s} = \begin{cases} \frac{R_u}{M_s} \frac{\theta_{v,s}}{\exp(\theta_{v,s}/T_{ve}) - 1} & \text{for molecules,} \\ 0 & \text{for atoms and electrons,} \end{cases} \quad (2.15)$$

where $\theta_{v,s}$ is the species characteristic vibrational temperature. This harmonic oscillator model is adequate because it is accurate for the low vibrational energy levels and because the energy contribution of the higher levels, where the model loses accuracy, is negligible[49]. Values for $\theta_{v,s}$ are listed in Appendix A. The vibrational specific heat at constant volume is calculated by

$$Cv_{v,s} = \begin{cases} \frac{R_u}{M_s} \frac{(\theta_{v,s}/T_{ve})^2 \exp(\theta_{v,s}/T_{ve})}{[\exp(\theta_{v,s}/T_{ve}) - 1]^2} & \text{for molecules,} \\ 0 & \text{for atoms and electrons,} \end{cases} \quad (2.16)$$

The species electronic energy, $e_{el,s}$ is modeled by

$$e_{el,s} = \begin{cases} \frac{R_u}{M_s} \frac{\sum_{i=1}^{\infty} g_{i,s} \theta_{el,i,s} \exp(-\theta_{el,i,s}/T_{ve})}{\sum_{i=0}^{\infty} g_{i,s} \exp(-\theta_{el,i,s}/T_{ve})} & \text{for molecules and atoms} \\ 0 & \text{for electrons,} \end{cases} \quad (2.17)$$

where $\theta_{el,i,s}$ and $g_{i,s}$ are the characteristic electronic temperature and the degeneracy of the i -th energy level, respectively. This model is adequate because it is accurate for the low electronic energy levels and because the energy contribution of the higher levels, where the model loses accuracy, is negligible[49]. Numerical values for the model parameters are listed in Appendix A for the 11-species air model. The species electronic specific heat at constant volume is given by

$$Cv_{el,s} = \begin{cases} \frac{\partial e_{el,s}}{\partial T_{ve}} & \text{for molecules and atoms} \\ 0 & \text{for electrons,} \end{cases} \quad (2.18)$$

where

$$\frac{\partial e_{el,s}}{\partial T_{ve}} = \frac{R_u}{M_s} \left\{ \frac{[\sum_{i=1}^{\infty} g_{i,s} (\theta_{el,i,s}/T_{ve})^2 \exp(-\theta_{el,i,s}/T_{ve})]}{\sum_{i=0}^{\infty} g_{i,s} \exp(-\theta_{el,i,s}/T_{ve})} - \frac{[\sum_{i=1}^{\infty} g_{i,s} \theta_{el,i,s} \exp(-\theta_{el,i,s}/T_{ve})][\sum_{i=0}^{\infty} g_{i,s} (\theta_{el,i,s}/T_{ve}^2) \exp(-\theta_{el,i,s}/T_{ve})]}{[\sum_{i=0}^{\infty} g_{i,s} \exp(-\theta_{el,i,s}/T_{ve})]^2} \right\}. \quad (2.19)$$

The vibrational-electron-electronic energy per unit mass of the species is given by

$$e_{ve,s} = \begin{cases} e_{v,s} + e_{el,s} & \text{for molecules and atoms,} \\ Cv_{t,e} T_{ve} & \text{for electrons} \end{cases} \quad (2.20)$$

and the species vibrational-electron-electronic specific heat at constant volume, $Cv_{ve,s}$, is given by

$$Cv_{ve,s} = \begin{cases} Cv_{v,s} + Cv_{el,s} & \text{for molecules and atoms,} \\ Cv_{t,e} & \text{for electrons.} \end{cases} \quad (2.21)$$

The mixture vibrational-electron-electronic energy per unit volume is given by

$$E_{ve} = \sum_s \rho_s e_{ve,s} \quad (2.22)$$

and the internal energy per unit mass of the species is given by

$$e_s = \begin{cases} Cv_{tr,s}T_{tr} + h_s^o + e_{ve,s} & \text{for molecules and atoms,} \\ e_{ve,e} & \text{for electrons} \end{cases} \quad (2.23)$$

and the species enthalpies are given by

$$h_s = \begin{cases} R_s T_{tr} + e_s & \text{for molecules and atoms,} \\ R_e T_{ve} + e_e & \text{for electrons.} \end{cases} \quad (2.24)$$

2.5 Transport Properties

In this work, the transport properties of the mixture can be calculated using two different models. The first model uses Wilke's mixing rule[94], Blottner's curve fits for viscosity[14] and Eucken's[91] relation for thermal conductivities. This model is adequate for relatively small speeds for which the flowfield maximum temperature is around 10,000 K. This simple model is not designed for ionized flows even though it can be modified to handle a weakly ionized gas, an approach not used in this work because it lacks generality[72]. For larger speeds, on the order of 10 km/s, Gupta's mixing rule[34] with species viscosities and thermal conductivities calculated using non-coulombic/coulombic collision cross section data[96] is employed, as suggested in Ref. [72].

2.5.1 Wilke/Blottner/Eucken

In this model, the mixture transport properties are modeled using Wilke's semi-empirical mixing rule[94], for which

$$\mu = \sum_s \frac{X_s \mu_s}{\phi_s} \quad \text{and} \quad \kappa = \sum_s \frac{X_s \kappa_s}{\phi_s} \quad , \quad (2.25)$$

where X_s is the species molar fraction, μ_s is the species coefficient of viscosity and κ_s is the species thermal conductivity for each energy mode. The term ϕ_s is given

by

$$\phi_s = \sum_r X_r \left[1 + \sqrt{\frac{\mu_s}{\mu_r}} \left(\frac{M_r}{M_s} \right)^{1/4} \right]^2 \left[\sqrt{8 \left(1 + \frac{M_s}{M_r} \right)} \right]^{-1}. \quad (2.26)$$

The species viscosities are calculated using Blottner's curve fits[14] as

$$\mu_s = 0.1 \exp[(A_s \ln T + B_s) \ln T + C_s], \quad (2.27)$$

where A_s , B_s and C_s are constants determined for each species. Their values, for an 11-species air model, are listed in Appendix A. The species thermal conductivities are determined using Eucken's relation[91] as

$$\kappa_{tr,s} = \frac{5}{2} \mu_s C v_{t,s} + \mu_s C v_{r,s} \quad \text{and} \quad \kappa_{ve,s} = \mu_s C v_{ve,s}. \quad (2.28)$$

The mass diffusion coefficient for each species, D_s , is replaced by a single binary coefficient D as a simple way to ensure that the sum of diffusion fluxes is zero. This approach is not accurate for velocities above 10 km/s [29]. Its value can be obtained assuming a constant Lewis number, Le , by

$$D = \frac{Le\kappa}{\rho C_p}, \quad (2.29)$$

where κ is the mixture rotational-translational thermal conductivity and C_p is the mixture translational-rotational specific heat at constant pressure.

2.5.2 Gupta/Collision cross section data

In this formulation, the viscosity and the thermal conductivities for each energy mode are calculated using Gupta's mixing rule[34] which is a simplification of the first-order Chapman-Enskog approximation adequate for weakly ionized gases. The mixing rule is extended to a multi-temperature gas mixture by evaluating collision cross sections at the controlling temperature. For heavy-heavy collisions, the controlling temperature is the translational-rotational temperature. For collisions

involving electrons, the controlling temperature is the vibrational-electron-electronic temperature. Hence, the viscosity of the mixture is calculated by

$$\mu = \sum_{s \neq e} \frac{m_s \gamma_s}{\sum_{r \neq e} \gamma_r \Delta_{sr}^{(2)}(T_{tr}) + \gamma_e \Delta_{se}^{(2)}(T_{ve})} + \frac{m_e \gamma_e}{\sum_r \gamma_r \Delta_{er}^{(2)}(T_{ve})} , \quad (2.30)$$

where the collision terms, Δ_{sr} , are evaluated at the different controlling temperatures T_{tr} or T_{ve} . In the previous expression,

$$\gamma_s = \frac{\rho_s}{\rho M_s} \quad (2.31)$$

is the molar concentration of each species and

$$m_s = \frac{M_s}{N_{Avogadro}} \quad (2.32)$$

is the mass of each species molecule or atom.

The translational thermal conductivity is given by

$$\kappa_t = \frac{15}{4} k_{B,SI} \sum_{s \neq e} \frac{\gamma_s}{\sum_{r \neq e} a_{sr} \gamma_r \Delta_{sr}^{(2)}(T_{tr}) + 3.54 \gamma_e \Delta_{se}^{(2)}(T_{ve})} , \quad (2.33)$$

where $k_{B,SI}$ is the Boltzmann constant in SI units and

$$a_{sr} = 1 + \frac{[1 - (m_s/m_r)][0.45 - 2.54(m_s/m_r)]}{[1 + (m_s/m_r)]^2} . \quad (2.34)$$

The expression for the rotational thermal conductivity is

$$\kappa_r = k_{B,SI} \sum_{s=mol} \frac{\gamma_s}{\sum_{r \neq e} \gamma_r \Delta_{sr}^{(1)}(T_{tr}) + \gamma_e \Delta_{se}^{(1)}(T_{te})} \quad (2.35)$$

and the vibrational-electronic thermal conductivity is given by

$$\kappa_{vel} = k_{B,SI} \frac{Cv_{ve}}{R} \sum_{s=mol} \frac{\gamma_s}{\sum_{r \neq e} \gamma_r \Delta_{sr}^{(1)}(T_{tr}) + \gamma_e \Delta_{se}^{(1)}(T_{te})} . \quad (2.36)$$

It should be observed that because the previous expression uses Cv_{ve} it includes both the vibrational and the electronic modes. The electron thermal conductivity is calculated by

$$\kappa_e = \frac{15}{4} k_{B,SI} \frac{\gamma_e}{\sum_r 1.45 \gamma_r \Delta_{er}^{(2)}(T_{ve})} . \quad (2.37)$$

The thermal conductivity for the translational-rotational mode is then given by the sum of each energy mode as

$$\kappa_{tr} = \kappa_t + \kappa_r . \quad (2.38)$$

The collision terms are given by

$$\Delta_{sr}^{(1)}(T) = \frac{8}{3} \left[\frac{2M_s M_r}{\pi R_u T (M_s + M_r)} \right]^{1/2} 10^{-20} \pi \Omega_{sr}^{(1,1)}(T) , \quad (2.39)$$

and

$$\Delta_{sr}^{(2)}(T) = \frac{16}{5} \left[\frac{2M_s M_r}{\pi R_u T (M_s + M_r)} \right]^{1/2} 10^{-20} \pi \Omega_{sr}^{(2,2)}(T) , \quad (2.40)$$

where the collision integrals $\pi \Omega_{sr}^{(l,j)}$ must be calculated according to the type of collision. The constant 10^{-20} converts to square meters from Angstroms which is the standard unit for collision integrals. For neutral-neutral, electron-neutral and neutral-ion collisions, the collision integrals are obtained by a variety of methods[96]. Such methods usually rely on modeling the interaction potential for a pair of species and integrating the differential cross section obtained from that potential over the entire solid angle space[34]. The approach for modeling of the potentials is diverse and based on theoretical or experimental data available for each interaction. The result of the integration still depends on the relative speed of the particles which is a function of the temperature of the species and the collision integral data is then tabulated as a function of temperature. Fortunately, the collision integrals for the interactions involving neutrals can be curve fit according to

$$\pi \Omega_{sr}^{(l,j)}(T) = D T^{[A(\ln T)^2 + B \ln T + C]} , \quad (2.41)$$

reducing the complexity of the calculations. The values for the coefficients A , B , C and D for all the collisions relevant to an 11-species air simulation are included in Appendix A.

For electron-ion, ion-ion and electron-electron collisions the interaction potential can be modeled using shielded Coulomb potentials[96]. The resulting collision integrals can be curve fit to

$$\pi\Omega^{(n,n)}(T) = 5.0 \times 10^{15} \pi (\lambda_D/T)^2 \ln \{ D_n T^* [1 - C_n \exp(-c_n T^*)] + 1 \} , \quad (2.42)$$

where

$$T^* = \frac{\lambda_D}{e_{CGS}^2 / (k_{B,CGS} T)} , \quad (2.43)$$

and

$$\lambda_D = \sqrt{\frac{k_{B,CGS} T}{4\pi n_{e,CGS} e_{CGS}^2}} \quad (2.44)$$

is the Debye length. In the last expressions, $k_{B,CGS}$ is the Boltzmann constant in CGS units, e_{CGS} is the fundamental charge in CGS units and $n_{e,CGS}$ is the electron number density also in CGS units. The values for the coefficients D_n , C_n , c_n are included in Appendix A.

The binary diffusion coefficients between heavy particles can be calculated by

$$D_{sr} = \frac{k_{B,SI} T_{tr}}{p \Delta_{sr}^{(1)}(T_{tr})} , \quad (2.45)$$

and for electrons as

$$D_{er} = \frac{k_{B,SI} T_{ve}}{p \Delta_{er}^{(1)}(T_{ve})} . \quad (2.46)$$

The species diffusion coefficient to be used in the conservation equations is given by

$$D_s = \frac{\gamma_t^2 M_s (1 - M_s \gamma_s)}{\sum_{r \neq s} (\gamma_r / D_{sr})} , \quad (2.47)$$

where

$$\gamma_t = \sum_s \gamma_s . \quad (2.48)$$

2.6 Source terms

The source term, $S_{c,v}$, is given by

$$S_{c,v} = \left\{ \begin{array}{ccccccc} \dot{w}_1 & \dots & \dot{w}_{ns} & 0 & 0 & 0 & \dot{w}_v \end{array} \right\}^T \quad (2.49)$$

where $\dot{w}_1 \dots \dot{w}_{ns}$ are the species mass production rates by chemical reactions and \dot{w}_v is the vibrational energy source term.

2.6.1 Chemical Model

This work assumes a standard finite-rate chemistry model for reacting air. In this work, an 11-species and a 5-species model for air and an 8-species model for the Martian atmosphere are used. The discussion in this section is based on the 11-species air model but it is immediately applicable to the other models. The species considered in the 11-species model are N₂, O₂, NO, N, O, N₂⁺, O₂⁺, NO⁺, N⁺, O⁺ and e. The reactions considered for the 11-species model are listed in the Appendix A. The reactions can be classified as dissociative, exchange, recombination ionization, charge exchange and impact ionization reactions. All reactions for the three models can be represented generically as

$$\sum \alpha_s [S] \rightleftharpoons \sum \beta_s [S] \quad , \quad (2.50)$$

where [S] represents one of the 11-species and α and β are the stoichiometric coefficients. The reactions are written such that the right arrow represents an exothermic reaction. The chemical production rate of species s in reaction k is given by[24]

$$\dot{w}_{sk} = (\beta_{sk} - \alpha_{sk}) \left[10^3 k_{fk} \prod_j \left(10^{-3} \frac{\rho_j}{M_j} \right)^{\alpha_{jk}} - 10^3 k_{bk} \prod_j \left(10^{-3} \frac{\rho_j}{M_j} \right)^{\beta_{jk}} \right] \quad (2.51)$$

where the factor 10^{-3} converts the concentration from kmol/m³ to mol/cm³ and the factor 10^3 converts from mol/(cm³s) to kmol/(m³s) because most of available

data for reactions are in CGS units. The source terms in the species conservation equations are given by

$$\dot{w}_s = M_s \sum_k \dot{w}_{sk} . \quad (2.52)$$

The forward and backward reaction rates are affected by the level of non-equilibrium in the flow. To account for that effect, Park's two-temperature model[74] is used. In that model the dissociation reactions are controlled by a combination of the translational-rotational and the vibrational-electron-electronic temperature

$$T_P = T_{tr}^a T_{ve}^b \quad (2.53)$$

to account for the fact that vibrationally excited molecules are more likely to dissociate. Two different set of values are typically used for a and b : $a = b = 0.5$ or $a = 0.7$ and $b = 0.3$. Exchange, charge exchange and recombination ionization reactions are controlled by the translational-rotational temperature and the impact ionization reactions are controlled by the electron temperature which in the current model is equilibrated with the vibrational-electron-electronic temperature.

The forward reaction rates are calculated using Arrhenius curve fits on the controlling temperature, T_c , as

$$k_{fk} = C_{fk} T_c^{\eta_k} \exp(-\theta_k/T_c) , \quad (2.54)$$

where T_c can be the translational-rotational (T_{tr}), the vibrational-electron-electronic (T_{ve}) or Park's temperature (T_P) and C_{fk} , η_k and θ_k are constants[74] which are listed in Appendix A. The backward reaction rate is obtained from

$$k_{bk}(T_{bc}) = \frac{k_{fbk}(T_{bc})}{K_{eq}(T_{bc})} , \quad (2.55)$$

where the backward controlling temperature, T_{bc} , is not necessarily the same as for the forward reaction. The backward controlling temperature for all reactions is T_{tr}

except for recombination ionization, impact ionization and impact dissociation of N₂ for which the controlling temperature is T_{ve} . The equilibrium constants can be given by curve fits[74] of the form

$$K_e(T) = e^{\left[A_1\left(\frac{T}{10000}\right) + A_2 + A_3 \ln\left(\frac{10000}{T}\right) + A_4\left(\frac{10000}{T}\right) + A_5\left(\frac{10000}{T}\right)^2\right]} , \quad (2.56)$$

where the coefficients A_i are a function of the local number density. The coefficients are listed in Appendix A. The values for A_i are interpolated for number densities within the range of the data. For number densities outside the range available, the tabulated values for the maximum and minimum number densities are used accordingly.

Another option to calculate the equilibrium constants is to use Gibb's free energy as

$$K_r = \left(\frac{p_o}{RT_{tr}}\right)^{\nu_r} \exp\left\{-\sum_s (\beta_{s,r} - \alpha_{s,r}) \left[\left(\frac{\hat{h}_s}{RT_{tr}} - \frac{\hat{s}_s}{R}\right)\right]\right\} , \quad (2.57)$$

where p_o is a reference pressure set to 1 bar and

$$\nu_r = \sum_s (\beta_{s,r} - \alpha_{s,r}) . \quad (2.58)$$

The normalized enthalpy and entropy are obtained from curve fits of the form

$$\frac{\hat{h}_s}{RT_{tr}} = -\frac{a_{1s}}{T^2} + a_{2s}\frac{\ln(T)}{T} + a_{3s} + a_{4s}\frac{T}{2} + a_{5s}\frac{T^2}{3} + a_{6s}\frac{T^3}{4} + a_{7s}\frac{T^4}{5} + a_{8s} + \frac{a_{9s}}{T} , \quad (2.59)$$

$$\frac{\hat{s}_s}{R} = -\frac{a_{1s}}{2T^2} - \frac{a_{2s}}{T} + a_{3s} \ln(T) + a_{4s}T + a_{5s}\frac{T^2}{2} + a_{6s}\frac{T^3}{3} + a_{7s}\frac{T^4}{4} + a_{8s} \ln(T) + a_{10s} , \quad (2.60)$$

where the values for the coefficients a_{1s} to a_{10s} for an 11-species air model are included in Appendix A.

2.6.2 Relaxation Model

The vibrational energy source term is given by

$$\dot{w}_v = S_{epg} + S_{c-v} + S_{t-v} + S_{h-e} + S_{e-i} \quad , \quad (2.61)$$

where $S_{epg} = -p_e \nabla \cdot \vec{u}$ is an approximation to the work done on electrons by the electric field induced by the electron pressure gradient[18, 24], S_{c-v} is the vibrational-electron-electronic energy added or removed by chemical reactions, S_{t-v} is the energy transferred between the translational-rotational and the vibrational-electron-electronic modes, S_{h-e} is the energy transfer between heavy particles and electrons, and S_{e-i} is the energy removed from free electrons during impact ionization reactions.

The vibrational energy added or removed by reactions can be modeled using a preferential or non-preferential model by

$$S_{c-v} = \sum \dot{w}_s \left(D'_s + e_{el,s} \right) \quad . \quad (2.62)$$

In the non-preferential model it is assumed that molecules are created or destroyed at the average vibrational energy, hence

$$D'_s = e_{v,s} \quad (2.63)$$

while in the preferential model it is assumed that the molecules are created or destroyed at higher vibrational energy levels. The value of energy added or removed is typically set as a fraction of the dissociation energy of the molecule or

$$D'_s = \alpha D_s \quad , \quad (2.64)$$

where α is typically set to 0.3 and D_s is the dissociation potential of the molecule listed in Appendix A for the 11-species air model. Both preferential and non-preferential models are phenomenological simplifications to a complicated physical process for which there are no definitive models.

The energy transfer rate between heavy particles and electrons, S_{h-e} , is given by

$$S_{h-e} = 3R_u\rho_e(T - T_v)\sqrt{\frac{8R_uT_v}{\pi M_e}} \sum_{r \neq e} \frac{\rho_r N_a}{M_r^2} \sigma_{er} , \quad (2.65)$$

where σ_{er} are the collision cross sections between electrons and heavy particles[49, 18] given by

$$\sigma_{er} = 10^{-20} \text{ m}^2 \quad (2.66)$$

for electron-neutral collisions and

$$\sigma_{er} = \frac{8\pi}{27} \frac{e^4}{k^2 T_e^2} \ln \left[1 + \frac{9k^3 T_e^3}{4\pi N_e e^6} \right] \quad (2.67)$$

for electron-ion collisions.

The term S_{e-i} models the energy removed from free electrons during impact ionization reactions. That energy is used to ionize a neutral atom. For the 11-species air model, there are only two impact ionization reactions involving N and O. The source term in this case is calculated by

$$S_{e-i} = M_{N+}\dot{w}_{N+,iir}\hat{I}_N + M_{O+}\dot{w}_{O+,iir}\hat{I}_O , \quad (2.68)$$

where \hat{I}_s is the energy used to ionize the species and the subscript *iir* stands for the impaction ionization reactions. The parameter \hat{I}_s can be set to the first ionization energy of the species, which overestimates the amount of electronic energy removed by the impact ionization reactions because it assumes that all energy necessary to ionize the neutrals is coming from the electrons. The energy removed in this case can lead to negative vibrational-electron-electronic energy in calculations. In this work, the terms \hat{I}_s are set to be around 1/3 of the first energy of ionization. The values employed are listed in Appendix A.

The term S_{t-v} is the energy exchange rate between the vibrational-electronic and the translational-rotational energy modes. It is assumed that these multi energy

mode exchange rates can be modeled by a single exchange rate due to the fast energy transfer between translational-rotational modes and between the vibrational-electronic modes. The single energy exchange rate is modeled using the Landau-Teller model[91] as

$$S_{t-v} = \sum_s \rho_s \frac{e_{vs}^* - e_{vs}}{\tau_s} , \quad (2.69)$$

where

$$\tau_s = <\tau_s> + \tau_{ps} \quad (2.70)$$

and where the molar averaged Landau-Teller relaxation time[49] is written as

$$<\tau_s> = \frac{\sum_r X_r}{\sum_r X_r / \tau_{sr}} , \quad (2.71)$$

where X_r is the molar fraction of species r . The Landau-Teller inter-species relaxation time, τ_{sr} , is modeled using curve fits developed by Millikan and White[62] for vibrational relaxation as

$$\tau_{sr} = \frac{101325}{p} \exp[A_{sr}(T^{-1/3} - B_{sr}) - 18.42] , \quad (2.72)$$

where

$$\begin{aligned} A_{sr} &= 1.16 \cdot 10^{-3} \mu_{sr}^{1/2} \theta_{vs}^{4/3} , \\ B_{sr} &= 0.015 \mu_{sr}^{1/4} , \\ \mu_{sr} &= \frac{M_s M_r}{M_s + M_r} . \end{aligned} \quad (2.73)$$

The values for A_{sr} and B_{sr} can be calculated using the previous generic expressions or using tabulated data from Ref. [37]. It is known that Millikan and White curve fits under predict the relaxation time at high temperatures. Park[74] corrected this problem by limiting the collision cross section and adding the corresponding limiting relaxation time to Eq. 2.70 given by

$$\tau_{ps} = \frac{1}{\sigma_s c_s N} , \quad (2.74)$$

where N is the number density of the mixture, c_s is the average molecular speed of the species given by

$$c_s = \sqrt{\frac{8R_u T}{\pi M_s}} \quad (2.75)$$

and σ_s is the limiting cross section given by

$$\sigma_s = 10^{-20} \left(\frac{50,000}{T} \right)^2 \text{ m}^2 \quad (2.76)$$

Equation 2.69 is simplified according to Ref. [24] as

$$S_{t-v} = \sum_s \rho_s \frac{e_{vs}^* - e_{vs}}{\tau_s} = \frac{\rho C v_{ve}}{\tau} (T - T_v) \quad , \quad (2.77)$$

where

$$\tau = \frac{\sum_{mol} \rho_s / M_s}{\sum_{mol} \rho_s / (M_s \tau_s)} \quad . \quad (2.78)$$

CHAPTER III

NUMERICAL METHOD

3.1 Introduction

In this chapter, the set of differential equations is integrated over a computational grid and over time to yield a linear system of equations to be solved. The spatial integration uses the finite-volume method and it requires the calculation of fluxes across cell faces. In this work, a Flux Vector Splitting (FVS) scheme is used to calculate the inviscid component of such fluxes and a centered scheme to calculate the viscous component. The time integration is performed using an implicit method to avoid the numerical stiffness caused by the chemistry source terms and because only steady state solutions are sought. The time integration requires the linearization of terms. The resulting linear system of equations obtained has to be solved iteratively due to the linearization performed and due to its size. The system is further simplified to facilitate its solution in a parallel machine by using either a point or a line solver[95]. This chapter also details the boundary conditions and their implementation in an implicit method and how the properties at cells are reconstructed to obtain 2nd order accuracy of the inviscid fluxes.

3.2 Spatial Integration and Flux Calculations

The finite-volume method[40] is obtained by integrating the set of equations in a generic mesh cell. It follows that

$$V_{cl} \frac{\partial Q_{cl}}{\partial t} = - \sum_{j \in cl} (\vec{F}_j - \vec{F}_{d_j}) \cdot \vec{n}_j s_j + V_{cl} S_{cv,cl} = R_{cl} \quad . \quad (3.1)$$

where $\vec{F} = E\vec{i} + F\vec{j} + G\vec{k}$ is the inviscid flux at the face, $\vec{F}_d = E_d\vec{i} + F_d\vec{j} + G_d\vec{k}$ is the diffusive flux at the face, $S_{cv,cl}$ is the chemistry and non-equilibrium source term, V_{cl} is the area of the cl -th cell, \vec{n} is the normal vector to the j -th face pointing outwards of the cl -th cell and s_j is the area of the j -th face. The summation is taken only over the faces that define the cl -th cell.

The inviscid flux vector at the j -th face is discretized using the Steger-Warming FVS scheme[86]. It uses the homogeneous property of the inviscid flux vector,

$$\vec{F} \cdot \vec{n} = F_n = \frac{dF_n}{dQ} Q = A Q \quad , \quad (3.2)$$

where F_n is the normal flux at the j -th face. The matrix A is called the Jacobian of the inviscid flux. It can be diagonalized by the matrices of its eigenvectors L and R as

$$A = L \Lambda R \quad (3.3)$$

and then separated into positive and negative parts

$$A^+ = L \Lambda^+ R \quad \text{and} \quad A^- = L \Lambda^- R \quad , \quad (3.4)$$

where the matrices Λ are the diagonal matrices of the eigenvalues of the Jacobian. The Steger-Warming scheme uses this split of the Jacobian to separate the flux on a downstream and on an upstream flux in relation to the face orientation as

$$\vec{F}_j \cdot \vec{n}_j = F_j^+ + F_j^- = (A_{cl}^+ Q_{cl} + A_{cr}^- Q_{cr}) \quad , \quad (3.5)$$

where cl and cr indicate the cells on the left and right sides of the face, respectively. The expressions for the matrices of the particular set of equations solved in this work can be found in Appendix B.1. The eigenvalues of the Jacobian are calculated by[16]

$$\lambda^{\pm} = \frac{1}{2} \left(\lambda \pm \sqrt{\lambda^2 + \epsilon^2} \right) , \quad (3.6)$$

where ϵ is supposed to be a small number used only to correct the sonic glitch problem[16]. However, it is very common practice to use it to add numerical dissipation to the scheme.

The Steger-Warming FVS is too dissipative and it cannot calculate boundary layer profiles accurately[60]. Therefore, a modified Steger-Warming FVS scheme[60] is used at boundary layers. The modification simply involves calculating the matrices A^+ and A^- using an average of the states on the left and right sides of the face,

$$\vec{F}_j \cdot \vec{n}_j = F_j^+ + F_j^- = (A_{j+}^+ Q_{cl} + A_{j-}^- Q_{cr}) , \quad (3.7)$$

where the subscripts $j+$ and $j-$ indicate that the Jacobians are no longer evaluated at cl or cr , but at average states

$$Q_{j+} = \frac{Q_{cl} + Q_{cr}}{2} \quad \text{and} \quad Q_{j-} = \frac{Q_{cl} + Q_{cr}}{2} . \quad (3.8)$$

This approach is very efficient and gives good results for boundary layers. However, it is not suitable in the vicinity of strong shock waves. Close to shock waves, the method needs to be switched back to the original Steger-Warming scheme which is more dissipative. A popular approach is to use a pressure switch of the form[21]

$$Q_{j+} = (1 - w)Q_{cl} + wQ_{cr} \quad \text{and} \quad Q_{j-} = (1 - w)Q_{cr} + wQ_{cl} , \quad (3.9)$$

where

$$w = \frac{1}{2} \frac{1}{(\alpha \nabla p)^2 + 1} \quad \text{and} \quad \nabla p = \frac{|p_{cl} - p_{cr}|}{\min(p_{cl}, p_{cr})} . \quad (3.10)$$

The value of α is usually set to 6 but some problems may require larger values. The modified Steger-Warming FVS scheme corresponds to $w = (1 - w) = 0.5$, which happens for small values of ∇p . The original Steger-Warming FVS scheme is recovered for $w = 0$ and $(1 - w) = 1$ which happens for large values of ∇p . In this work, extra dissipation is added using ϵ . The extra dissipation deteriorates results inside the boundary layer and, therefore, it has to be reduced there. The expression for ϵ is then

$$\epsilon_j = \begin{cases} 0.3(a_j + |\vec{u}_j|) & d_j > d_0 \\ 0.3(1 - |\vec{n}_j \cdot \vec{w}_j|)(a_j + |\vec{u}_j|) & d_j < d_0 \end{cases}, \quad (3.11)$$

where d_j is the face distance to the nearest wall boundary, d_0 is a constant set by the user that should be larger than the boundary layer thickness but smaller than the shock stand-off distance, \vec{w}_j is the normal vector of the nearest wall boundary and \vec{n}_j is the normal vector to the face. The term $(1 - |\vec{n}_j \cdot \vec{w}_j|)$ reduces the value of ϵ at faces that are parallel to the wall inside the boundary layer. For the calculations performed in this work d_0 is set to 5×10^{-6} meters.

The viscous terms are calculated using the values of properties at the cell centers and at the nodes [44]. The property values at the nodes are calculated using a simple average of the values of the cells that share that node. Use of this method increases the stencil used in the derivative calculations which avoids loss of accuracy when using unstructured meshes[44].

The source terms are evaluated as

$$S_{cv,cl} = S_{cv}(Q_{cl}) . \quad (3.12)$$

Caution must be used when including chemistry in a numerical code. The forward and backward rate of reactions can be very large numbers depending on the temperature in the cell. In this work, for the conditions simulated, the backward rate

of reaction for small temperatures sometimes became too large due to very small values for the equilibrium constant Kc . The forward and backward reaction rates are computed using a modified temperature

$$\begin{aligned} T'_f &= \frac{1}{2} \left((T_f + T_{min}) + \sqrt{(T_f - T_{min})^2 + \epsilon^2} \right), \\ T''_f &= \frac{1}{2} \left((T_f + T_{min}) - \sqrt{(T_f - T_{min})^2 + \epsilon^2} \right), \end{aligned} \quad (3.13)$$

where T_{min} is set to 800 K and $\epsilon = 80$ K in this work. Another important modification that needs to be addressed in a numerical code is that species densities need to be positive at all times, even for very small numbers. A negative value for species density, even a very small negative number, changes the sign of the chemical source terms causing the code to diverge.

3.3 Time Integration

The system of equations used to numerically simulate hypersonic, chemically reacting viscous flows is very stiff[40]. One of the options to overcome the time-step limitations in such a case is using implicit time integration. Starting from the backward Euler method one obtains

$$\frac{V_{cl}}{\Delta t} \Delta Q_{cl}^n = \left[- \sum_{j \in cl} (F_{n,j} - F_{dn,j}) s_j + V_{cl} S_{cv,cl} \right]^{n+1} = R_{cl}^{n+1}, \quad (3.14)$$

where R_{cl}^{n+1} is called the residue at the cl -th cell. Linearizing the residue at time $n+1$ as a function of the residue at time n it follows that

$$\begin{aligned} \frac{V_{cl}}{\Delta t} \Delta Q_{cl}^n &= R_{cl}^n + \left[- \sum_{j \in cl} \left(\left(\frac{\partial F_n}{\partial Q} \right)_j \Delta Q - \left(\frac{\partial F_{dn}}{\partial Q} \right)_j \Delta Q \right) s_j + \right. \\ &\quad \left. V_{cl} \left(\frac{\partial S_{cv}}{\partial Q} \right)_{cl} \Delta Q_{cl} \right]^n. \end{aligned} \quad (3.15)$$

Now, from the spatial discretization of the inviscid terms,

$$\left(\frac{\partial F_n}{\partial Q} \right)_j \Delta Q = \left(\frac{\partial F_n^+}{\partial Q} \right)_j \Delta Q + \left(\frac{\partial F_n^-}{\partial Q} \right)_j \Delta Q. \quad (3.16)$$

It is common practice to use

$$\begin{aligned} \left(\frac{\partial F_n^+}{\partial Q} \right)_j \Delta Q &= A_{j+}^+ \Delta Q_{cl} \\ \left(\frac{\partial F_n^-}{\partial Q} \right)_j \Delta Q &= A_{j-}^- \Delta Q_{cr} \end{aligned} , \quad (3.17)$$

which assumes that

$$\frac{\partial F_n^\pm}{\partial Q} = A^\pm \quad (3.18)$$

which is not true and can deteriorate the stability of the numerical method. In this work, the true Jacobians of the split fluxes are used in the place of A^\pm when calculating the implicit operator. Those matrices are provided in Appendix B.2. The diffusive terms can be approximated by[66]

$$\left(\frac{\partial F_{dn}}{\partial Q} \right)_j \Delta Q = B_{j-}^- \Delta Q_{cr} - B_{j+}^+ \Delta Q_{cl} . \quad (3.19)$$

The matrices B are presented in Appendix B.3. The Jacobian for the source term S_{cv} is given by

$$\left(\frac{\partial S_{cv}}{\partial Q} \right)_{cl} = C_{cl} + D_{cl} \quad (3.20)$$

where C_{cl} is the chemistry part and D_{cl} is the thermal non-equilibrium part of the Jacobian. These matrices are presented in Appendices B.4 and B.5 respectively. Hence, a line in the system of equations to be solved can be written as

$$\begin{aligned} \left[\frac{V_{cl}}{\Delta t} + \sum_{j \in cl} (A_{j+}^+ + B_{j+}^+) s_j - V_{cl} (C_{cl} + D_{cl}) \right] \Delta Q_{cl} + \\ \sum_{j \in cl} (A_{j-}^- - B_{j-}^-) s_j \Delta Q_{cr,j} = R_{cl}^n \end{aligned} , \quad (3.21)$$

or, in a simpler form, as

$$M_{cl} \Delta Q_{cl}^n + \sum_{j \in cl} N_j^- \Delta Q_{cr,j}^n = R_{cl}^n , \quad (3.22)$$

where

$$N_j^- = (A_{j-}^- - B_{j-}^-) s_j , \quad (3.23)$$

$$N_j^+ = (A_{j+}^+ + B_{j+}^+) s_j \quad (3.24)$$

and

$$M_{cl} = \frac{V_{cl}}{\Delta t} + \sum_{j \in cl} N_j^+ - V_{cl} (C_{cl} + D_{cl}) \quad . \quad (3.25)$$

The term $Q_{cr,j}^n$ stands for the values at the cell to the right of the j -th face. When using unstructured grids, Eq. 3.22 is a sparse matrix without any structure. Considering the mesh shown in Fig. 3.1, the corresponding matrix form of Eq. 3.22 for the entire mesh is shown in Fig. 3.2. In that figure, each square corresponds to a block matrix. The size of each block matrix is given by the sum of the number of species conservation equations plus three momentum equations plus two energy equations. The solution of this system of algebraic equations using a direct matrix inversion is not feasible. In this work, a point or a line implicit[95, 23, 89] approach modified for unstructured grids is used to simplify and solve this system.

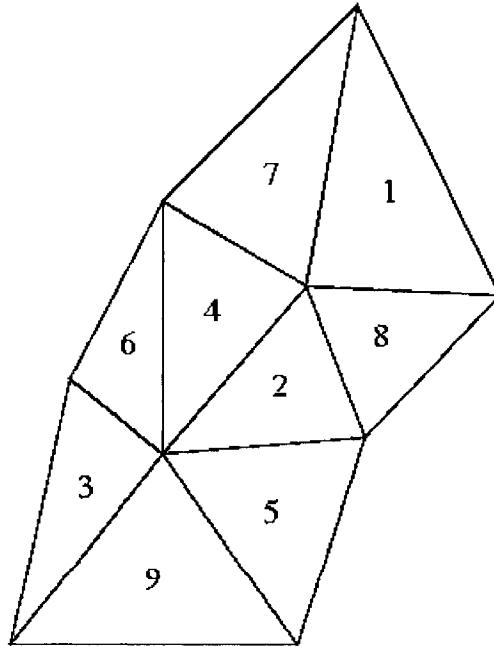


Figure 3.1: Unstructured grid example.

$$\begin{bmatrix} 1 & 2 & 3 & + & 5 & 6 & 7 & 8 & 9 \\ 1 & \square & & & & & & & \\ 2 & & \square & & & & & & \\ 3 & & & \square & & & & & \\ 4 & & & & \square & & & & \\ 5 & & & & & \square & & & \\ 6 & & & & & & \square & & \\ 7 & & & & & & & \square & \\ 8 & & & & & & & & \square \\ 9 & & & & & & & & & \end{bmatrix} = \begin{Bmatrix} \Delta Q^{(n+1)} \\ R^{(n)} \end{Bmatrix}$$

Figure 3.2: Typical sparse linear system for unstructured grids.

3.3.1 Point Solver

The point method is obtained by moving all the off-diagonal block matrices to the right hand side and then solving the resulting system of equations iteratively by

$$M_{cl} \Delta Q_{cl}^{n+1,k+1} = - \sum_{j \in cl} N_j^- \Delta Q_{cr,j}^{n+1,k} + R_{cl}^n . \quad (3.26)$$

The corresponding matrix form of equation 3.26 for the entire mesh is shown in Fig. 3.3. The system of equations is clearly block diagonal. Assuming that $\Delta Q^{(n+1,0)} = 0$,

$$\begin{bmatrix} 1 & 2 & 3 & + & 5 & 6 & 7 & 8 & 9 \\ 1 & \square & & & & & & & \\ 2 & & \square & & & & & & \\ 3 & & & \square & & & & & \\ 4 & & & & \square & & & & \\ 5 & & & & & \square & & & \\ 6 & & & & & & \square & & \\ 7 & & & & & & & \square & \\ 8 & & & & & & & & \square \\ 9 & & & & & & & & & \end{bmatrix} = \begin{Bmatrix} \Delta Q^{(n+1,p)} \\ R^{(n)} \end{Bmatrix} - \begin{bmatrix} 1 & 2 & 3 & + & 5 & 6 & 7 & 8 & 9 \\ 1 & \square & & & & & & & \\ 2 & & \square & & & & & & \\ 3 & & & \square & & & & & \\ 4 & & & & \square & & & & \\ 5 & & & & & \square & & & \\ 6 & & & & & & \square & & \\ 7 & & & & & & & \square & \\ 8 & & & & & & & & \square \\ 9 & & & & & & & & & \end{bmatrix} \begin{Bmatrix} \Delta Q^{(n+1,p-1)} \end{Bmatrix}$$

Figure 3.3: Iterative solution of linear system using point implicit method.

$\Delta Q^{(n+1,1)}$ is obtained by solving a simple block diagonal system of equations. That system is solved simply by

$$\Delta Q_{cl}^{n+1,1} = M_{cl}^{-1} R_{cl}^n . \quad (3.27)$$

The next step requires a multiplication of the sparse matrix on the right hand side by the newly calculated $\Delta Q^{(n+1,1)}$ or

$$\tilde{R}_{cl}^{n+1,k} = - \sum_{j \in cl} N_j^- \Delta Q_{cr,j}^{n+1,k} . \quad (3.28)$$

That multiplication is straightforward when using the face-based data structure of the mesh. In the face-based data structure, faces are stored with a defined orientation such that a face has a cell on its left side and a cell on its right side and the normal vector to the face points from left to right, as depicted in Fig. 3.4. In this work, only

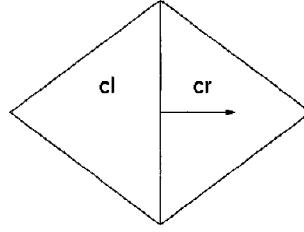


Figure 3.4: Face orientation.

the matrices N_j^+ and N_j^- , calculated using the defined face orientation, are stored at each face. It should be noted that the finite-volume integration assumes that the vector \vec{n} points outwards of the cell. For clarity, a subscript *cl* is added in the notation to $N_{j,cl}^+$ and $N_{j,cl}^-$ to indicate that those matrices have the correct normal vector for the *cl*-th cell only. Due to its properties, it can be shown that the matrices

$$N_{j,cr}^+ = -N_{j,cl}^- \quad (3.29)$$

and

$$N_{j,cr}^- = -N_{j,cl}^+ , \quad (3.30)$$

where the subscript *cr* indicates that those matrices have the correct normal vector for the *cr*-th cell. Hence, for the *j*-th face and its two adjacent cells *cl* and *cr*, the contributions to the sparse matrix multiplication, $\tilde{R}_{cl,j}^{n+1,k}$ and $\tilde{R}_{cr,j}^{n+1,k}$, are given by

$$\tilde{R}_{cl,j}^{n+1,k} = -N_{j,cl}^- \Delta Q_{cr,j}^{n+1,k} \quad (3.31)$$

for the cl -th cell and

$$\tilde{R}_{cr,j}^{n+1,k} = -(-N_{j,cl}^+) \Delta Q_{cl,j}^{n+1,k} \quad (3.32)$$

for the cr -th cell. Thus, instead of performing a loop over cells and then a loop over the faces that define that cell, the present work performs only one loop over all the faces according to the simple algorithm

SPARSE MATRIX MULTIPLICATION

```

1  for  $j \leftarrow 1$  to  $NFM$ 
2      do
3           $cl \leftarrow cl[j]$   $\triangleright$  Find face's left cell.
4           $cr \leftarrow cr[j]$   $\triangleright$  Find face's right cell.
5           $\tilde{R}[cl] \leftarrow -N[j][-] \cdot Q[cr]$ 
6           $\tilde{R}[cr] \leftarrow +N[j][+] \cdot Q[cl]$ 
7      end
```

and the contributions are distributed to the cells to left and right of each face.

Following Ref. [95], four iterations are taken in the iterative process.

The time step used in the simulations is set by

$$\Delta t = \text{CFL} \frac{h}{\|\vec{v}\| + a} \quad , \quad (3.33)$$

where h is a measure of the cell size, $\|\vec{v}\| + a$ is an estimate of the largest wave speed in the cell and CFL is a parameter set to ensure stability of the time integration method. The maximum CFL number of this method varies according to the freestream conditions[95]. The code starts running with CFL set to one and increases it exponentially with the number of iterations.

3.3.2 Line Solver

The line method has better convergence performance but it comes at the expenses of a more complicated algorithm. There are many algorithms available for efficient solution of sparse linear systems. Such algorithms rely heavily on general mathematical properties of matrices to solve the system. These algorithms, however, do not take into account some flow characteristics that can lead to a faster solution of the system. For instance, flow gradients will be stronger in the direction normal to the body according to boundary layer theory. It was shown[95] that solving the system of equations exactly in the normal direction for structured grids and relaxing the system in the other directions was efficient in terms of convergence and also facilitated the parallelization of an implicit code. The same approach is used in this work for an unstructured grid code.

The idea of solving the linear system originated by an implicit method using special directions in unstructured grids is not new[39, 61, 89, 67]. The first step in such a method is to define the special directions that will be used. In this work, such directions are called lines that are grown from the body surface by a simple algorithm that chooses the next cell such that the line stays nearly normal to the body. Cells that are “leftover” are grouped together and handled by a point implicit algorithm. The lines formed from the mesh in Fig. 3.1 are indicated in Fig. 3.5. That figure also shows a renumbering of the cells that will be shown to facilitate the solution of the linear system. The renumbering of the cells is never carried out in the code. Instead, an array keeps track of the relation between the two cell numbering systems.

Using the numbering for the cells provided in Fig. 3.5, the sparse system is greatly simplified as can be observed in Fig. 3.6. The matrix is still sparse, but one can see a tridiagonal structure embedded in the matrix. In order to take advantage of

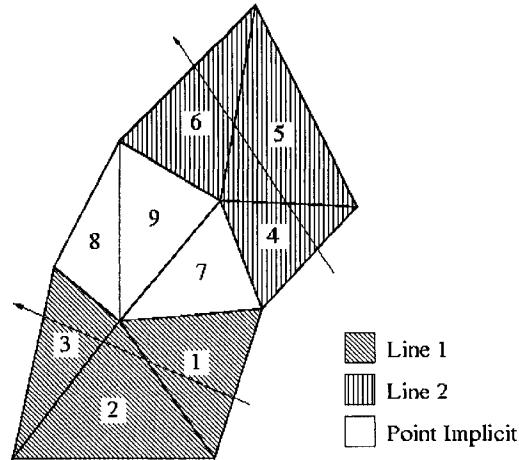


Figure 3.5: Unstructured grid renumbering.

the tridiagonal structure, an iterative process is assembled. All the terms off the tridiagonal part of the matrix are moved to the right hand side of the equation. By doing that, the iterative process is now of the form depicted in Fig. 3.7. The last three lines of the system represent the three cells “leftover” from the line finding procedure. Those cells corrections are calculated using a point implicit method which is naturally integrated in the tridiagonal solver.

$$\begin{bmatrix} 1 & 2 & 3 & + & 5 & 6 & 7 & 8 & 9 \\ 1 & \begin{matrix} \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \end{matrix} & & & & & & & \\ 2 & & \blacksquare & & & & & & \\ 3 & & & \blacksquare & & & & & \\ 4 & & & & \blacksquare & \blacksquare & \blacksquare & & \\ 5 & & & & & \blacksquare & \blacksquare & \blacksquare & \\ 6 & \blacksquare & & & & & \blacksquare & & \\ 7 & & \blacksquare & & & & & \blacksquare & \\ 8 & & & \blacksquare & & & & & \blacksquare \\ 9 & & & & \blacksquare & & \blacksquare & & \blacksquare \end{bmatrix} \left\{ \Delta Q^{(n+1)} \right\} = \left\{ R^{(n)} \right\}$$

Figure 3.6: Full linear system.

The implicit equation corresponding to Fig. 3.7 is given by

$$O_i \Delta Q_{i-1}^{n+1,k+1} + M_i \Delta Q_i^{n+1,k+1} + P_i \Delta Q_{i+1}^{n+1,k+1} = \tilde{R}_i^{n+1,k} + R_i^n . \quad (3.34)$$

where the subscript i reflect the new numbering used in the lines. The block matrices O_i and P_i form the off-diagonals and the block matrices M_i compose the diagonal. The sparse matrix contribution, $\tilde{R}_i^{n+1,k}$, is actually calculated using the original mesh numbering and then mapped to the new numbering as

$$\tilde{R}_{i(cl)}^{n+1,k} = - \sum_{j \in cl, j \ni line} N_j^- \Delta Q_{cr,j}^{n+1,k} , \quad (3.35)$$

where the function $i(cl)$ represents the mapping from the original numbering to the line solver numbering. The summation in the previous equation does not include the faces that are between cells $i-1$ and i and between i and $i+1$. Those contributions are actually accounted for by the matrices O_i and P_i respectively. Due to its diagonal characteristic, the system is very simple to solve. Assuming that $\Delta Q^{(n+1,0)} = 0$, $\Delta Q^{(n+1,1)}$ is obtained by solving a simple tridiagonal system of equations. The next step requires a multiplication of the sparse matrix on the right hand side by the newly calculated $\Delta Q^{(n+1,1)}$. That multiplication is straightforward when using the face-based data structure of the mesh and the original mesh numbering. The result of that multiplication is mapped to the linear system numbering and a new right hand side is fed to the tridiagonal solver. Following Ref. [95], four iterations are taken in the iterative process. As for the point implicit method, the code starts running with CFL set to one and increases it exponentially.

3.4 Boundary Conditions

The implementation of boundary conditions in this work is performed using “ghost” cells. Ghost cells are not created by the grid generator. Instead, they are logically created by the numerical code and are associated with a boundary face. The values of its properties are set so that the flux calculation on the boundary face yields the correct flux. For instance, for a wall boundary face, the correct inviscid

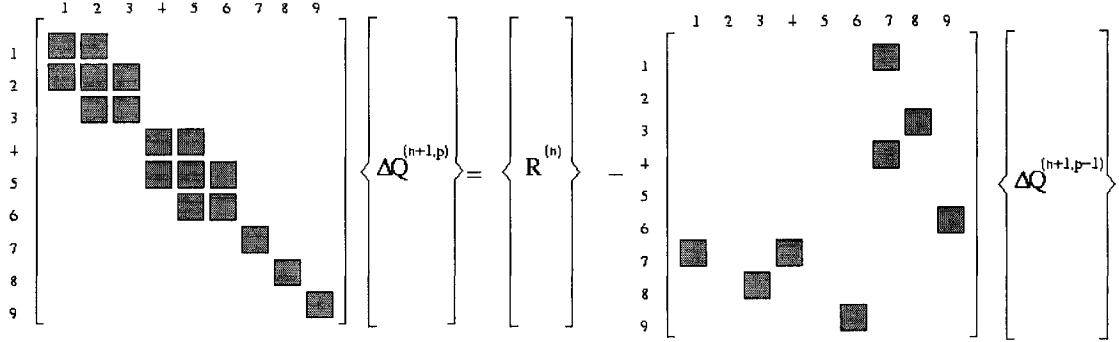


Figure 3.7: Iterative solution of linear system using line implicit method.

flux calculations should yield zero mass and energy fluxes and a momentum flux equal to the pressure. The viscous flux for the wall boundary face should yield a prescribed diffusion flux, viscous forces and heat fluxes. Only one set of properties is not enough to satisfy all the constraints. In this work, the properties of the ghost cell change according to the flux calculation being performed. When the inviscid fluxes are being calculated, the ghost cells hold properties such that the inviscid fluxes are correct at the boundaries. The same happens at the viscous flux calculation.

3.4.1 Inviscid Flux Boundary Condition

This section details the values used as inviscid boundary conditions necessary for the problems modeled in this work.

Wall, Symmetry

The inviscid flux for a symmetry face and a wall are the same. No mass and energy fluxes and momentum flux given by the pressure only. That is accomplished by setting the normal velocity to the wall to zero. This is easier to achieve by working in the face coordinate frame. The properties at the face reference frame are given by

$$Q'_{cl} = RQ_{cl} \quad (3.36)$$

where R is the rotation matrix, given in Appendix B.3. Species densities and energies are unchanged by the rotation matrix. The properties at the ghost cell are set to

$$\begin{aligned} \rho_{cr,s} &= \rho_{cl,s} \\ \rho_{cr} u'_{cr,n} &= -\rho_{cl} u'_{cl,n} \\ \rho_{cr} u'_{cr,t} &= \rho_{cl} u'_{cl,t} \\ \rho_{cr} u'_{cr,r} &= \rho_{cl} u'_{cl,r} \\ E_{cl} &= E_{cr} \\ E_{v,cl} &= E_{v,cr} \end{aligned} \quad . \quad (3.37)$$

It can be observed that taking the face value as the average between the values at the cr -th and cl -th face yields that $u_{w,n} = 0$ which satisfies the condition for no mass or energy flux through the wall.

The boundary condition can be written in matrix form as

$$Q'_{cr} = W Q'_{cl} \quad (3.38)$$

where

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.39)$$

and then

$$Q_{cr} = R^{-1} W R Q_{cl} \quad , \quad (3.40)$$

where the rotation matrix R^{-1} is used to return the velocities back to the Cartesian coordinate frame.

Exit

In this case, zeroth order extrapolation is used because this work focuses on hypersonic external flows, for which the exit plane is always at supersonic speeds. The properties at the ghost cell are given by

$$Q_{cr} = Q_{cl} \quad . \quad (3.41)$$

Freestream

This work assumes that the free stream boundaries are not influenced by any signal coming from the computational domain such as sound or shock waves. That is accomplished by guaranteeing that no shock waves reach the boundary by careful grid generation. The properties at the ghost cell are given by

$$Q_{cr} = Q_{\infty} \quad . \quad (3.42)$$

where Q_{∞} is constant and known.

3.4.2 Viscous Boundary Condition

This section details the values used as viscous boundary conditions necessary for the problems modeled in this work.

Wall

The values of properties at the wall for a non-catalytic wall are set as

$$\begin{aligned}
 p_w &= p_{cl} \\
 Y_{w,s} &= Y_{cl} \\
 u_w &= 0 \\
 v_w &= 0 \quad , \\
 w_w &= 0 \\
 T_w &= T_w^* \\
 T_{v,w} &= T_{v,w}^*
 \end{aligned} \tag{3.43}$$

where the values for T_w^* and $T_{v,w}^*$ are known. For a catalytic wall the properties are set to

$$\begin{aligned}
 p_w &= p_{cl} \\
 Y_{w,s} &= Y_{w,s}^* \\
 u_w &= 0 \\
 v_w &= 0 \quad , \\
 w_w &= 0 \\
 T_w &= T_w^* \\
 T_{v,w} &= T_{v,w}^*
 \end{aligned} \tag{3.44}$$

where the values for $Y_{w,s}^*$, T_w^* and $T_{v,w}^*$ are known. The values at the ghost cell should be set so that the gradients of properties calculated using data at cr -th and cl -th cells are the same as if using data at cl -th cell and at the wall. Thus, the properties at the ghost cell are set to

$$x_{cr} = 2x_w - x_{cl} \quad , \tag{3.45}$$

where x can be the pressure, mass fractions, velocities and temperatures. The density at the ghost must be set to calculate the values of conserved variables. This is

obtained using

$$\rho_{cr} = \frac{p_{cr}}{\sum_{s \neq e} Y_{cr,s} R_s T_{tr,cr} + Y_{cr,e} R_e T_{ve,cr}} \quad (3.46)$$

and

$$\rho_{cr,s} = \rho_{cr} Y_{cr,s} \quad . \quad (3.47)$$

This work also can use a non-catalytic wall in radiative equilibrium. That is accomplished by setting the radiative flux given by $q_w^r = \sigma T_w^4$, where σ is the Stefan-Boltzmann constant, equal to the convective heating crossing the wall, $q_{w,n}^c$. The convective heating entering the wall can be approximated by

$$q_{w,n}^c = (\kappa_{tr} + \kappa_{ve}) \frac{T_{tr,cl} - T_w}{\Delta n} \quad , \quad (3.48)$$

where Δn is the normal distance between the centroid of the cl -th cell and the centroid of the boundary face. The previous expression assumes that T_{ve} is equilibrated with T_{tr} immediately over the wall. Using that approximation, the wall temperature is set by solving the equation

$$\sigma T_w^4 - (\kappa_{tr} + \kappa_{ve}) \frac{T_{tr,cl} - T_w}{\Delta n} = 0 \quad (3.49)$$

iteratively at each wall boundary face. Both translational-rotational and vibrational-electron-electronic temperatures are set at the solution of the equation.

Symmetry, Exit, Freestream

For these boundary conditions, it is assumed that there are no property gradients at the boundary. This is easily enforced by setting

$$Q_{cr} = Q_{cl} \quad . \quad (3.50)$$

3.5 Implicit Boundary Conditions

The implementation of implicit boundary conditions in this work uses “folding” matrices. The implicit equation

$$\left[\frac{V_{cl}}{\Delta t} + (A_{j+}^+ + B_{j+}^+) s_j - V_{cl} (C_{cl} + D_{cl}) \right] \Delta Q_{cl} + \\ (A_{j-}^- - B_{j-}^-) s_j \Delta Q_{cr,j} = R_{cl}^n \quad , \quad (3.51)$$

is repeated here without the summation to focus only on the relation between an internal cell, cl , and a boundary cell (cr, j) . The full equation for the cl -th cell certainly has contributions from other faces, that may or may not be boundary faces, that need to be considered. They are not included in the above equation just for clarity. As stated before, the ghost cell holds different values for inviscid and viscous flux calculations. Hence, the above equation needs to be expanded to

$$\left[\frac{V_{cl}}{\Delta t} + (A_{j+}^+ + B_{j+}^+) s_j - V_{cl} (C_{cl} + D_{cl}) \right] \Delta Q_{cl} + \\ A_{j-}^- s_j \Delta Q_{cr,j,inv} - B_{j-}^- s_j \Delta Q_{cr,j,vis} = R_{cl}^n \quad (3.52)$$

to describe the contribution of a boundary face to the system. The folding matrices relate the property values at the ghost cell, cr , with the property values at the internal cell, cl as

$$\Delta Q_{cr,j,inv} = E_{j,inv} \Delta Q_{cl} \quad (3.53)$$

$$\Delta Q_{cr,j,vis} = E_{j,vis} \Delta Q_{cl} \quad . \quad (3.54)$$

Using those expressions, the contribution to the system of equations of the boundary face is given by

$$\left[\frac{V_{cl}}{\Delta t} + (A_{j+}^+ + E_{j,inv} A_{j-}^- - B_{j-}^- E_{j,vis} + B_{j+}^+) s_j - V_{cl} (C_{cl} + D_{cl}) \right] \Delta Q_{cl} = R_{cl}^n \quad . \quad (3.55)$$

It should be observed that the cl -th cell has contributions from other faces that should be included when writing its full implicit equation. Because the viscous

Jacobians are easier to calculate using primitive variables as shown in Appendix B.3, the viscous folding matrices are set for the primitive variables and applied directly at the calculation of the Jacobians so that B_{j+}^+ already includes the contribution from the boundary. To reflect that fact, the previous equation is modified to

$$\left[\frac{V_{cl}}{\Delta t} + (A_{j+}^+ + E_{j,inv} A_{j-}^- + B_{j+}^{+,*}) s_j - V_{cl} (C_{cl} + D_{cl}) \right] \Delta Q_{cl} = R_{cl}^n . \quad (3.56)$$

where the matrix $B_{j+}^{+,*}$ accounts for the contribution from the boundary. The calculation of that matrix is shown in Appendix B.3.

3.5.1 Inviscid Matrices

The folding matrix E for an inviscid wall is given simply by differentiating Eq. 3.40 to

$$\Delta Q_{cr} = R^{-1} W R \Delta Q_{cl} , \quad (3.57)$$

and the folding matrix is then given by

$$E = R^{-1} W R . \quad (3.58)$$

Similarly, the folding matrix for an exit is given by

$$E = I , \quad (3.59)$$

where I is the identity matrix. For the free stream boundary condition, there is no relation between the internal value and the free stream value that is a constant. Therefore, it is set that

$$E = 0 , \quad (3.60)$$

where 0 represents a matrix of zeroes.

3.5.2 Viscous Matrices

Because the viscous Jacobians are easier to calculate using primitive variables, the viscous folding matrices are set for the primitive variables and applied directly at the calculation of the Jacobians as shown in Appendix B.3. The primitive variables are

$$V = \left\{ Y_s \ u \ v \ w \ T_{tr} \ T_{ve} \right\}^T . \quad (3.61)$$

The folding matrix for a viscous wall is obtained by differentiating Eq. 3.45 for each property. Therefore, for a non-catalytic wall where $Y_w = Y_{cl}$ and the other wall properties are constant, the folding matrix becomes

$$F = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} . \quad (3.62)$$

and for a catalytic wall, where Y_w is a prescribed constant, the folding matrix becomes

$$F = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} . \quad (3.63)$$

For the radiative equilibrium wall, Eq. 3.49 can be differentiated yielding

$$\sigma T_w^3 \Delta T_w - (\kappa_{tr} + \kappa_{ve}) \frac{\Delta T_{tr,cl} - \Delta T_w}{\Delta n} = 0 . \quad (3.64)$$

Now, assuming that typical SI values for the wall temperature are around 10^3 K, that the Stefan-Boltzmann constant is around 10^{-8} J/(K 4 m 2 s), κ is around 10^{-2} J/(K m s) and Δn is around 10^{-7} m, the previous equation is simplified to

$$\Delta T_w = \Delta T_{tr,cl} \quad . \quad (3.65)$$

Using the differentiated form of Eq. 3.45 gives that

$$\Delta T_{tr,cr} = \Delta T_{tr,cl} \quad , \quad (3.66)$$

and

$$\Delta T_{ve,cr} = 2\Delta T_{tr,cl} - \Delta T_{ve,cl} \quad . \quad (3.67)$$

Hence, the folding matrix for a non-catalytic wall in radiative equilibrium is given by

$$F = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 2 & -1 \end{bmatrix} . \quad (3.68)$$

Finally, for exit, symmetry and free stream boundary conditions, the folding matrix is simply given by the identity matrix

$$F = I \quad . \quad (3.69)$$

3.6 Higher order extension of inviscid fluxes

Higher-order of accuracy extensions of finite volume codes using Flux Vector Splitting (FVS) or Riemann solvers to discretize the inviscid fluxes are straightforward

in structured grids because of the implicit connectivity information. The MUSCL variable extrapolation[40] originally developed for 1-D calculations is readily extrapolated for 2 dimensions when using a structured grid. For the faces aligned to the vertical direction in the structured grid shown in Fig. 3.8, the values to the left (u_L) and to the right (u_R) are obtained according to

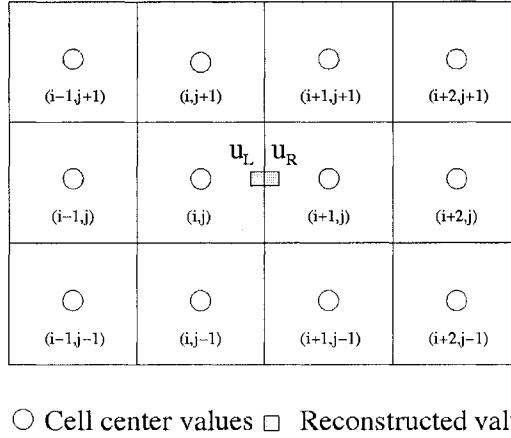


Figure 3.8: 2nd order stencil for a structured grid.

$$\begin{cases} u_L = u_{(i,j)} + \frac{1}{2}\Psi(r_L)(u_{(i,j)} - u_{(i-1,j)}) \\ u_R = u_{(i+1,j)} - \frac{1}{2}\Psi(r_R)(u_{(i+2,j)} - u_{(i+1,j)}) \end{cases}, \quad (3.70)$$

where

$$r_L = \frac{u_{(i+1,j)} - u_{(i,j)}}{u_{(i,j)} - u_{(i-1,j)}} \quad , \quad r_R = \frac{u_{(i+1,j)} - u_{(i,j)}}{u_{(i+2,j)} - u_{(i+1,j)}} \quad (3.71)$$

and $\psi(r)$ is a limiter function[40]. Even though all the meshes used in this work are “structured like” grids, the code handles them internally as unstructured grids. Hence, the implicit connectivity information - the information of each cell neighbors - characteristic of structured codes is not available making the above extension to higher order of accuracy more complicated.

Many approaches are available in the literature to achieve high order of accuracy using unstructured grids. The most well known and widely used is the one developed

by Barth and Jespersen[9]. Many authors modified the ideas of Barth and Jespersen to create their own extension to high order of accuracy[48, 88]. Those approaches are highly effective at transonic and low supersonic flows but they seem to fail at hypersonic flows due to the presence of stronger shocks.

A different approach to higher order extension using unstructured grids was developed by Batina[10] in which a structured stencil is assembled locally so Eq. 3.70 can be used. One example of such a stencil is shown in Fig. 3.9(a). In that figure, the points $i - 1$ and $i + 2$ are the opposite vertices of the triangles that share the face for which the fluxes are being computed. The values at the vertices are obtained by averaging the cell center values of the cells that share the vertices. A similar idea is also used by Bibb et al[12] in the FELISA cell-vertex code. Though FELISA is a 3D code, the idea is exemplified for a 2D case in Fig. 3.9(b). The coordinates of point $i - 1$ in that figure are obtained by a reflection of the vector $i, i + 1$ in relation to the point i . The property values at the point $i - 1$ are obtained from interpolation of the values at vertices that define the cell in which the point lies. A problem with these two approaches is that the values of variables in some points of the localized stencil are obtained by interpolation or averaging. Close to shock waves, the interpolated or averaged quantities can be too large hindering the limiting process.

The approach proposed here follows the idea of Batina[10] and Bibb et al[12] closely. However, the interpolation step is eliminated. Instead, the stencil is constructed such that only cell centered values are used. The search for the stencil is depicted in Fig. 3.10 for “structured like” and unstructured grids. This figure shows the search for the cell $i + 2$. The search for the cell $i + 2$ is limited to cells that share at least one node with cell $i + 1$. The cells that meet such a restriction are shaded in Fig. 3.10. The selected cell will be the one such that the dot product between the

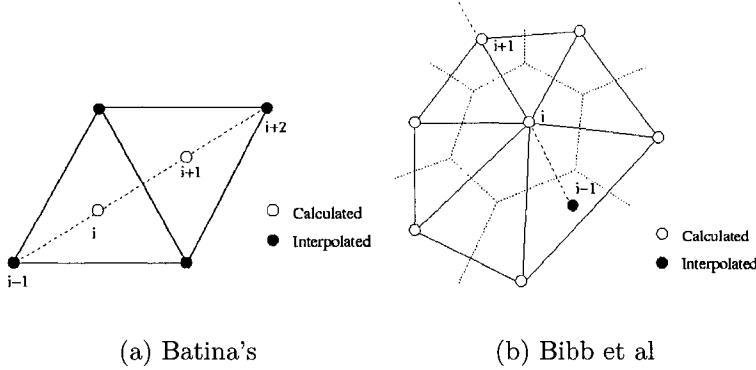


Figure 3.9: Stencil search approach for 2nd order calculations.

face normal and the normalized vector joining the face centroid to the cell centroid has the maximum positive value. Figure 3.10 shows some of the face centroid-cell centroid vectors and the selected cells. For the search of the cell $i - 1$, the dot product must be the maximum negative value.

It should be noted that the distance between the cells $i + 1$ and $i + 2$ centroids is different from the distance between i and $i + 1$. That is clearly the case in Fig. 3.10(b). Therefore, the undivided differences used to calculate the limiter must be corrected according to

$$r = \frac{u_{i+1} - u_i}{u_{i+2} - u_{i+1}} \frac{b}{a} \quad (3.72)$$

where $\|a\| = \|\vec{r}_{i+1} - \vec{r}_i\|$ and $\|b\| = \|\vec{r}_{i+2} - \vec{r}_{i+1}\|$. It is important to note that the search for the stencil is performed only once at the start of the code run. This is in contrast to ENO[35, 85]/WENO[57] schemes that search for stencils at every iteration. The cost for the stencil search is negligible in relation to the total run.

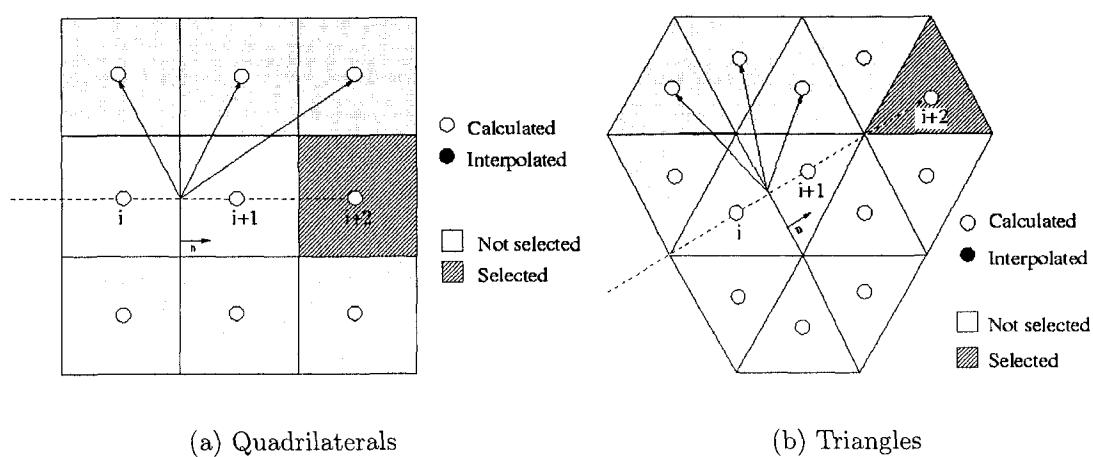


Figure 3.10: Second-order stencil search.

CHAPTER IV

IMPLEMENTATION ASPECTS

4.1 Introduction

This chapter addresses some aspects of the implementation of the numerical code LeMANS. While it is not feasible to detail all the routines in a parallel, implicit, unstructured, multi-dimensional CFD code, it is worthwhile to detail the general structure of the code and some of the more complex operations that are performed during its execution. In particular, the line-finding algorithm and the parallelization strategy are described in this chapter. It should be noted that some of the other important operations, such as the 2nd order stencil finding algorithm and the implicit integration in an unstructured code are described in Chapter III because they aid the comprehension of the numerical formulation.

4.2 Code Overview

LeMANS is coded using the C language to have access to dynamic memory allocation and to facilitate the creation of a hybrid code[82, 84, 83] using the DSMC code MONACO[20] also available at the University of Michigan. An overview of the execution of LeMANS is shown below followed by a brief explanation of how the algorithm connects to the numerical and mathematical models presented in Chapters

II and III.

“LE” MICHIGAN AEROTHERMODYNAMICS NAVIER-STOKES SOLVER

```

1 Read problem setup file
2 Read chemistry and thermodynamic data files
3 Read mesh
4 Process mesh
5     Create face based data structure
6     Create ghost cells
7     Assemble 2nd order stencils for each face
8     Assemble lines normal to the wall
9     Calculate distance of each face and cell to the wall
10    Partition mesh
11   Assign initial condition to each cell
12   Calculate thermodynamic and transport properties at each cell
13 While residue norm < tolerance do
14   Calculate time step in each cell
15   Assign inviscid boundary conditions to ghost cells
16   Calculate inviscid fluxes at each face
17   Calculate inviscid residue at each cell
18   Calculate inviscid Jacobians at each face
19   Assign implicit inviscid boundary conditions (“Folding”) at boundary faces
20   Assign viscous boundary conditions to ghost cells
21   Calculate primitive variable derivatives at each face
22   Calculate viscous fluxes at each face
23   Calculate viscous residue at each cell
24   Calculate viscous Jacobians at each face
25   Assign implicit viscous boundary conditions (“Folding”) at boundary faces
26   Calculate chemistry source terms at each cell
27   Calculate non-equilibrium source terms at each cell
28   Integrate
29     Calculate chemistry Jacobians at each cell
30     Calculate non-equilibrium Jacobians at each cell
31     Assemble tri-diagonal linear system with Jacobians and residue
32     Loop
33       Solve linear system for conserved property corrections at each cell
34       Communicate corrections at inter-processor boundary
35       Perform sparse matrix multiplication and update linear system
36     End Loop
37     Update conserved properties at each cell
38     Calculate thermodynamic and transport properties at each cell
39 End While
40 Output results.
41 End

```

LeMANS starts by reading a problem setup file which has several options regarding the physical models to be employed in the simulation. Next, LeMANS reads all

the species database files necessary to calculate thermodynamic and transport properties. The database data is included in Appendix A. LeMANS then reads the mesh, which should be written using FLUENT format[5] because most of the commercial grid generators can create FLUENT files. The mesh is processed to create the face based data structure in which each face structure stores the cells on its left and right sides. For the 2nd order scheme, the face based data structure also stores the cells to the far left and far right, as described in Section 3.6. The mesh processing also creates the ghost cells to be used to assign the boundary conditions, calculates the distance of the face to the nearest wall and creates the lines used in the line solver. Before starting the time march loop, LeMANS assigns the initial condition to the domain.

In the time march loop, LeMANS calculates the time step by using Eq. 3.33 in all the cells and apply the minimum time step value to all the cells. The inviscid boundary conditions are applied according to the type of boundary, as presented in Section 3.4. The inviscid fluxes are calculated using a Flux Vector Splitting scheme and taking advantage of the face based data structure in which the fluxes are calculated on a per-face basis. The inviscid residue is calculated by adding/subtracting the face fluxes to the cells sharing the face, as shown in Eq. 3.1. The inviscid residue is added to what is called the cell residue, as in Eq. 3.14, which is a measure of how far the cell is from steady state. The inviscid Jacobians, shown in Appendix B, are then calculated. The inviscid Jacobians are two matrices that are stored at the face: one for the left cell and one for the right cell. The inviscid implicit boundary conditions are then applied, as described in Section 3.5. It basically consists of adding the boundary face inviscid Jacobian associated with the ghost cell to the face inviscid Jacobian associated with the internal cell. The viscous boundary conditions are then

assigned as detailed in Section 3.4. The viscous fluxes are calculated at each face following Eq. 3.1 and the viscous residue is calculated by adding/subtracting the fluxes to the residue of the cells sharing the face, Eq. 3.14, similarly for the inviscid residue calculation. The viscous Jacobians, shown in Appendix B, are also calculated on a per-face basis and added to the already stored inviscid Jacobians at the faces. The viscous implicit boundary conditions, described in Section 3.5, are assigned similarly as for the inviscid Jacobians.

The chemistry and non-equilibrium source terms are calculated by evaluating the expressions in Section 2.6 at the cells and adding the result to the cell residue, Eq. 3.14. The chemistry and non-equilibrium Jacobians are calculated according to Appendix B and stored on a per-cell basis. The linear system of equations is then assembled using the per-cell chemistry and non-equilibrium Jacobians and the per-face inviscid and viscous Jacobians as in Eqs. 3.26 or 3.34 according to the type of solver selected by the user. The linear system of equations is then solved and the corrections to be added to the cell properties are calculated. The properties at the cell are updated and thermodynamic and transport properties are calculated using the models described in Section 2.5. The time march continues until convergence to steady state is reached.

4.3 Line Finding Algorithm

The simplification of the linear system created by the implicit integration of the differential equations uses lines of cells as close to perpendicular to the wall as possible as mentioned in Chapter III. In a structured code, this is trivial. The implicit connectivity information in the structured grid already provides the line structure. Assuming that the cells in a two-dimensional structured grid are numbered as (i, j) ,

the lines are defined as the cells for which the “i” index is the same. For unstructured grids that information is not available. The cells are not numbered as a matrix but instead as a vector. Hence, the list of cells that compose each line must be searched and the information stored somehow.

The line-finding algorithm in an unstructured code must be general enough to use grids made of any mix of elements such as triangles and quadrilaterals in two-dimensions and hexahedra, tetrahedra, pyramids and prisms in three-dimensions. The algorithm developed in this work has three objectives: generality for two-dimensional and three-dimensional unstructured grids, maximization of the length of the lines and maintaining the line as normal to the wall as possible. Figures 4.1 to 4.5 illustrate the line-finding algorithm.

First, the lines are created starting at the walls only, as depicted in Fig. 4.1 which shows three lines, (A, B and C), the normal vector that defines the line direction and the first cell that compose each line. It should be noted in that figure that cell number 6 is shared by two-boundaries: the wall at the bottom and another boundary at the right that is not a wall, such as a symmetry or exit boundary. For the sake of clarity, only part of a two-dimensional mesh made of triangles is shown and each line has a distinct shading pattern for its cells. The choice of the first cell that composes each line is trivial for convex walls because cells at the wall boundary have just one wall boundary face associated with it. For concave walls, one internal cell may be shared by two or more wall faces.

Next, a list of possible options of cells to be added to each line is assembled, as shown in tabular form in Fig. 4.2. The list is limited to cells that are not already assigned to any lines and to cells that have at least one face in common with the last cell added to the line. The options are ranked according to the dot product of

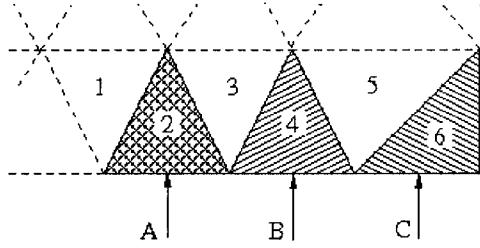


Figure 4.1: Lines creation process.

the normal vector that defines the line and the vector connecting the possible cell to the last cell in the line. Those vectors are also indicated in Fig. 4.2. This ranking intends to keep the line as normal to the wall as possible. Finally, the list of lines is ranked according to the number of options for growth that each line has. Lines that have fewer options will have preference in selecting its next cell. This ranking intends to maximize the length of the lines by avoiding that a line with more than one option for growing selects a cell that is the only option for another line.

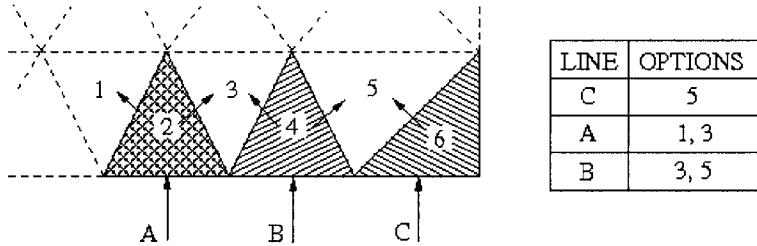


Figure 4.2: Lines creation process - cont.

Figure 4.3 shows what happens next. Cell number 5 is assigned to line C and then it is removed from the growth options of line B which now has less options than line A. Hence, the line ranking is modified and line B will have the priority in the next growth round, which is shown in Fig. 4.4. In this round, cell number 3 is assigned to line B and removed from the growth options of line A. Finally, with just one option left, cell number 1 is assigned to line A, as shown in Fig. 4.5. The process would then repeat itself for a third row addition to the lines and so on. The growth

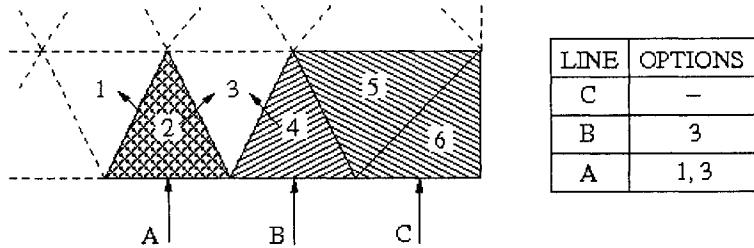


Figure 4.3: Lines creation process - cont.

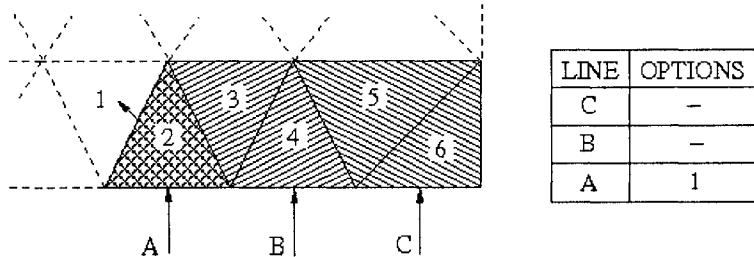


Figure 4.4: Lines creation process - cont.

options of each line are populated by looping over the faces that define the last cell in the line and, therefore, is general for any type of cell used in the unstructured code. In order to illustrate the generality, this algorithm is employed to create lines in a

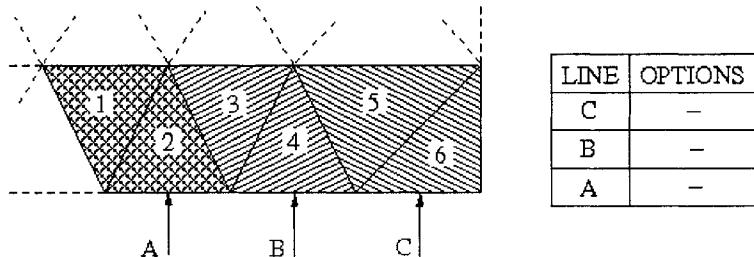


Figure 4.5: Lines creation process - concluded.

two-dimensional grid composed of triangles and layers of quadrilaterals close to the wall, a common practice in viscous simulations using unstructured grids. A general overview of the lines created is presented in Fig. 4.6 where the lines are represented by using different colors to fill their cells. The algorithm works remarkably well: most of the lines are able to reach the freestream boundary and the lines are approximately normal to the wall. A detailed view of the mesh close to wall in Fig. 4.7 shows that



Figure 4.6: Lines created in a hybrid two-dimensional grid composed of triangles and quadrilaterals.

the change in cell type from quadrilaterals to triangles does not impact the line-finding algorithm. Figure 4.8 shows that the algorithm is able to recover the lines in a grid made of quadrilaterals treated as an unstructured grid. All the lines reach the freestream boundary and are as normal to the body as possible.

The justification for the added complexity of using the line solver is shown in Fig. 4.9, which shows the convergence history for a numerical simulation of a $M = 20$ viscous flow over a blunt-cone. The fluid is N_2 and the freestream conditions for this flow are $u_\infty = 2680.47 \text{ m/s}$, $T_\infty = 43.210 \text{ K}$, $\rho_\infty = 7.817 \times 10^{-4} \text{ kg/m}^3$, $T_w = 300 \text{ K}$. The nose radius for the blunt cone is 0.1524 meters and the cone angle is 9 degrees. The simulation is run with a constant CFL number equal to 100 and the initial condition is of freestream flow in all the computational domain. It can be observed that both methods present similar behavior for the beginning of the simulation in which the blunt shock moves from the wall to its final stand-off distance due to the

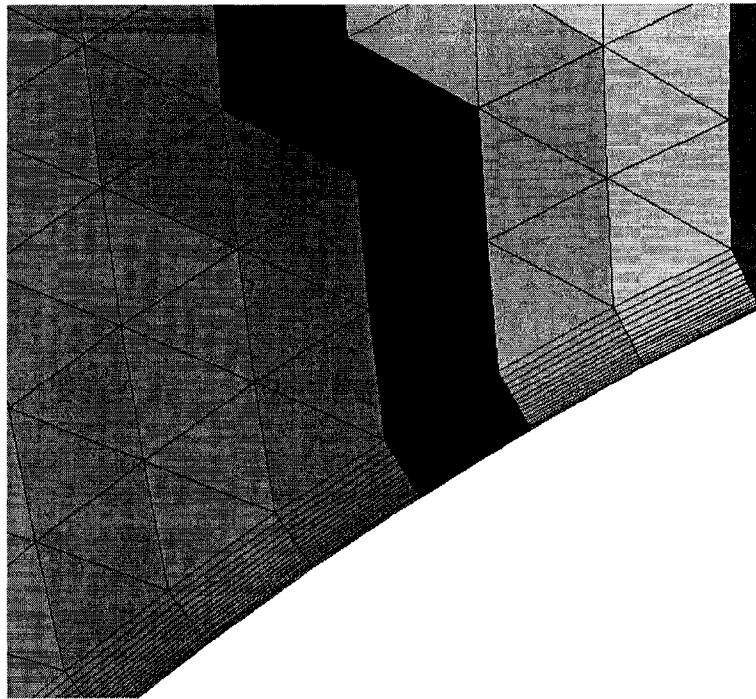


Figure 4.7: Detail of the lines created in a hybrid two-dimensional grid composed of triangles and quadrilaterals.

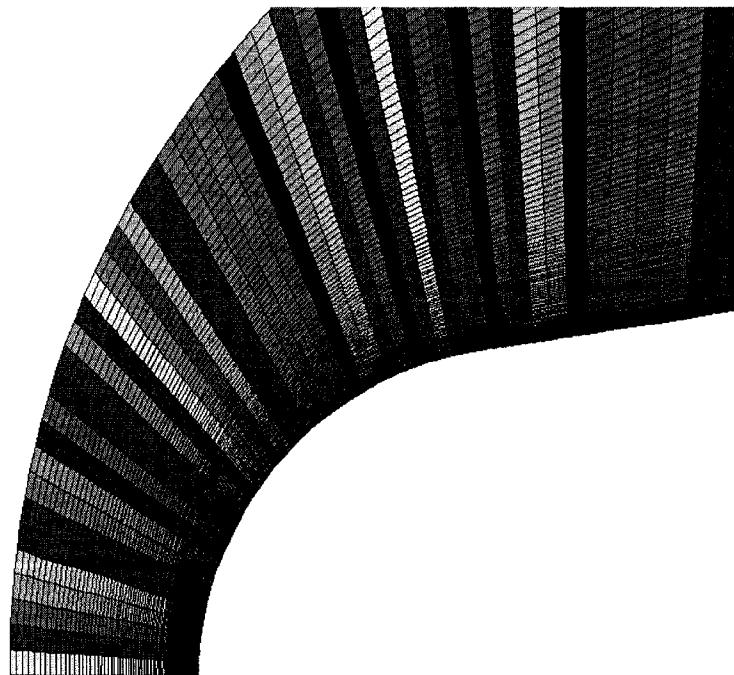


Figure 4.8: Lines created in a two-dimensional mesh composed of quadrilaterals and treated as an unstructured grid.

initial condition chosen. After the shock reaches its final position the line implicit solver leads to a much faster convergence to machine zero.

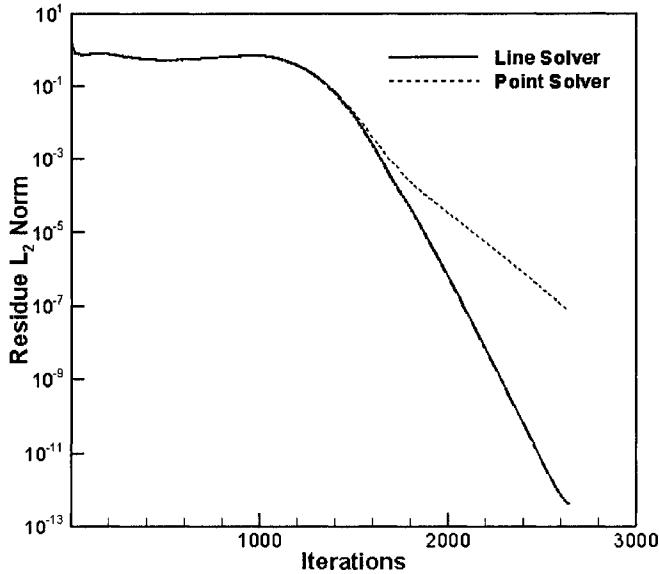


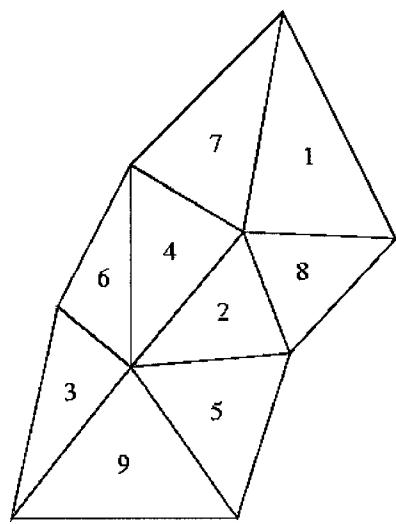
Figure 4.9: Convergence to steady state for a $M = 20$ flow over a blunt cone.

4.4 Mesh Partitioning

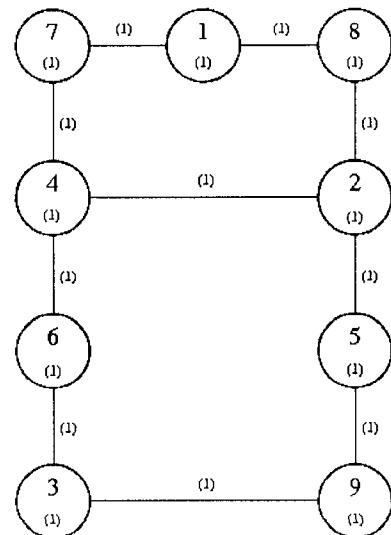
The code developed in this work takes full advantage of today's parallel machines by dividing the computational work among many processors. This task is accomplished using METIS[46] to partition the mesh between the processors and the Message Passage Interface protocol (MPI) to communicate necessary information from one processor to another. METIS uses graph theory to partition unstructured grids minimizing the amount of inter-processor communication. The numerical code assembles a graph that logically represents the mesh which is then partitioned by METIS. The result provided by METIS is a partition number for each node where each partition in fact corresponds to a processor. The numerical code uses that information to distribute the work among the processors. Two different ways to assemble

the graph are implemented in LeMANS, one for the point implicit solver and another for the line implicit solver.

In the most simple case, each node in the graph corresponds to a cell in the mesh and each edge in the graph connecting two nodes corresponds to the face shared by the cells represented by the two nodes. Figure 4.10 shows a pictorial model of a grid made of triangles over a cylinder and its graph. The node numbering in the graph is the same numbering used in the mesh. The numbers in parenthesis are node or edge weights. The node weight represents the amount of computational work to be executed in the corresponding cell. The edge weight represents the amount of communication to be performed at that edge if that edge is split in the partitioning. For a CFD application, the amount of computational work per cell is approximately constant. The amount of communication per edge is related to the amount of data sent to the ghost cell. It is also approximately constant in a CFD application. Hence, the code uses a weight of 1 per cell and a weight of 1 per edge. One possible partitioning for this mesh is presented in Fig. 4.11. A partition generated this way usually is not optimal for a line implicit solver because most of the lines are significantly shortened. The partitioning provides load balance but the wall boundary is entirely contained in one processor and the maximum length of the lines is reduced to just one cell instead of a possible maximum of three cells. In order to avoid such problems, the graph is modified so that each node in fact corresponds to a line. The node weight is given by the number of cells that make the line. Each edge corresponds to the face connecting one line to a neighboring cell. In that case, the edge weight is set to one. For two adjacent lines, an edge in the graph corresponds to all the faces connecting the adjacent lines and its weight is set to the number of faces shared by the lines. Such modification is pictured in Fig. 4.12 which shows the logical grid in which the lines



(a) Grid



(b) Graph

Figure 4.10: Unstructured grid and corresponding graph used in METIS.

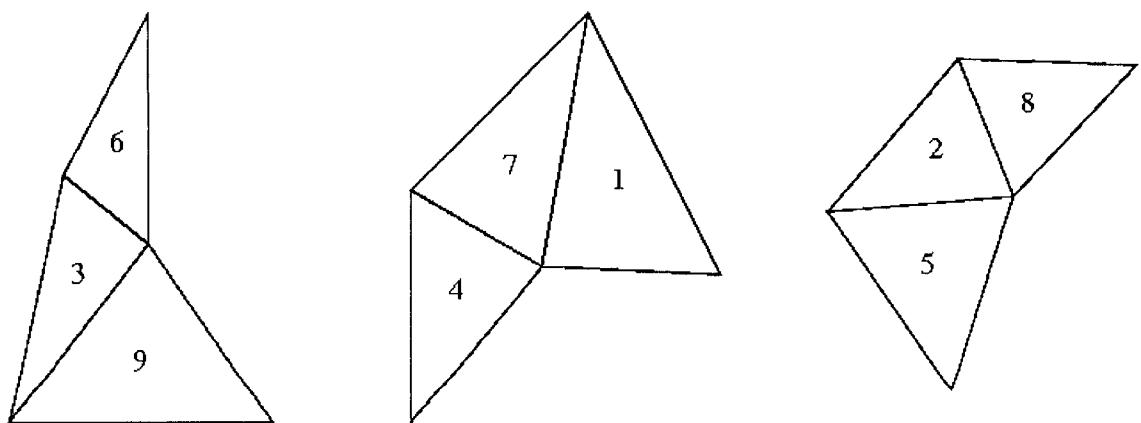


Figure 4.11: Mesh partitions.

are seen as extended cells. In this case, the two lines are not adjacent and the edge weights are all one. The weights for the two nodes that represent the lines are set to three because both lines are composed of three cells in this case. The partitioning for

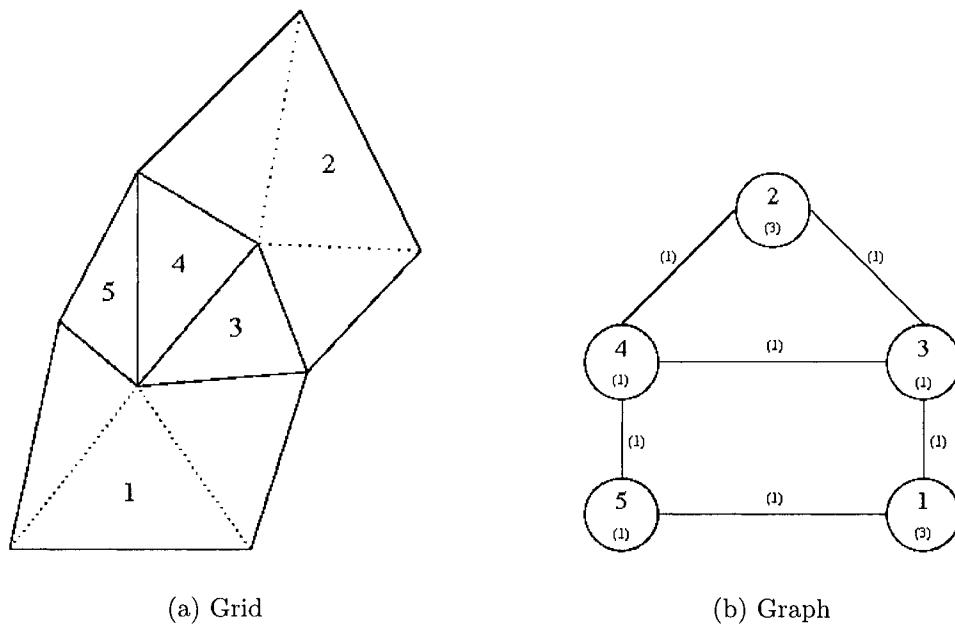


Figure 4.12: Modified logical grid and corresponding graph.

this grid is presented in Fig. 4.13. This partitioning allows the creation of the same lines as in the serial problem. The implemented algorithm is highly effective as can

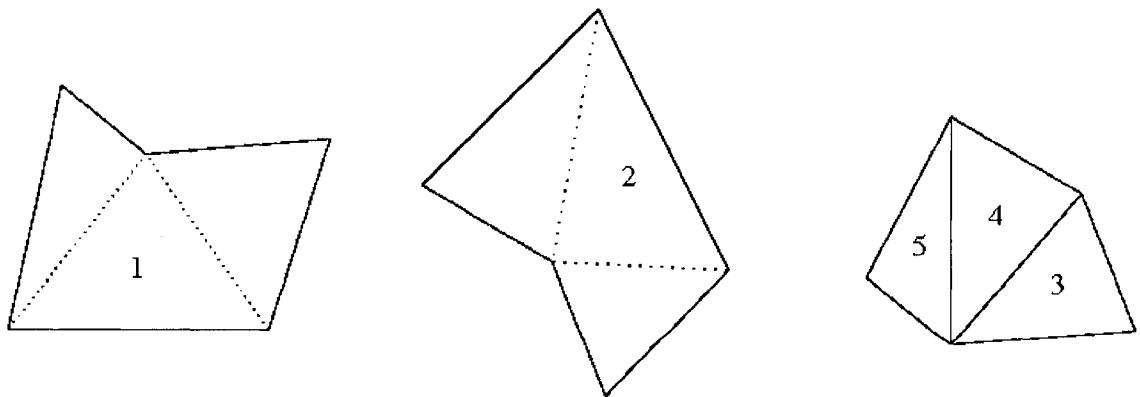


Figure 4.13: Mesh partitions.

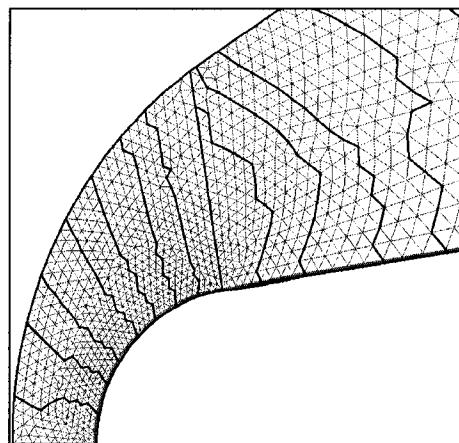
be observed in Fig. 4.14 which shows the partitions for a hybrid unstructured grid

and for a “structured like” grid. It is clear in these figures that lines growing normal to the body can reach their maximum length before reaching an inter-processor boundary.

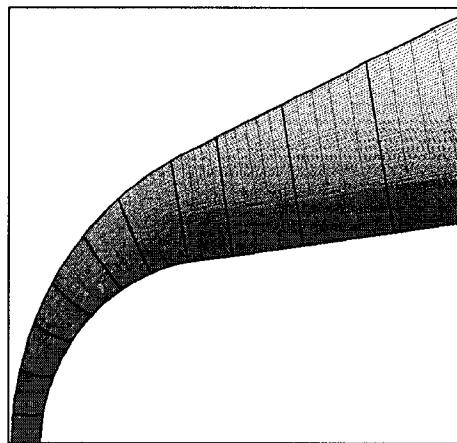
4.4.1 Inter-processor Boundary Conditions

The mesh partitioning creates logical boundaries inside the grid. Information must be shared between the processors so that the final result is exactly the same as if the problem was simulated in a single processor. The data is communicated between processors using MPI. The numerical code uses ghost cells to store the data that is received from each processor. Figure 4.15 shows part of an actual partitioned grid. It depicts the cell data that needs to be sent from processors 1, 2 and 3 to processor 0 for the inviscid flux calculations, viscous flux and node averaging and implicit integration. The calculation of the inviscid fluxes requires two layers of cells because of the 2nd order scheme. The node averaging requires corner cells to calculate node values. The viscous fluxes also need those corner cells because the node values are used in gradient calculations. The implicit integration requires one layer of cells to perform the sparse matrix multiplication in the iterative procedure. During pre-processing of the mesh, each processor stores the cells for which data has to be send to other processors. It also creates ghost cells with information regarding which processor will send data for that ghost cell. At each communication event, each processor will send the necessary data to other processors and also receive data for its inter-processor ghost cells. For simplicity, the code communicates all the cell data shown in Fig. 4.15(d) regardless of the task being performed, increasing the communication cost.

The conserved variables are communicated before the inviscid flux calculations



(a) Unstructured



(b) Structured

Figure 4.14: Mesh partitioning.

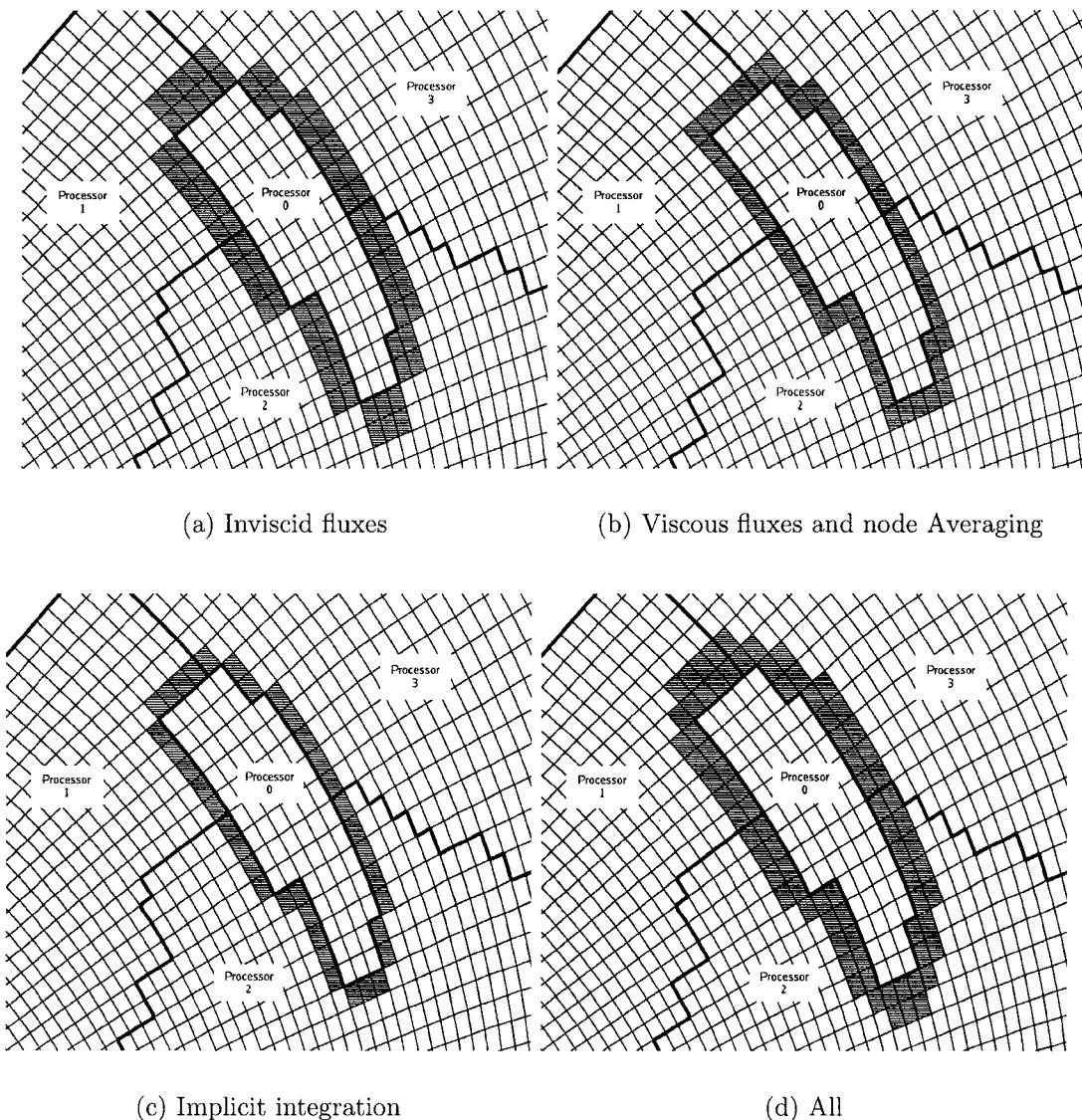


Figure 4.15: Cells used in inter-processor boundary conditions in different parts of the code.

and the primitive variables are communicated before the viscous flow calculation. At each iteration of the implicit integration, the corrections $\Delta Q^{(k,n)}$ are communicated as well. The primitive variable are communicated between processors before the results are output in order to perform the node averaging. The correct implementation of the inter-processor boundary conditions is illustrated by the smoothness of property lines across inter-processor boundaries in Fig. 4.16.

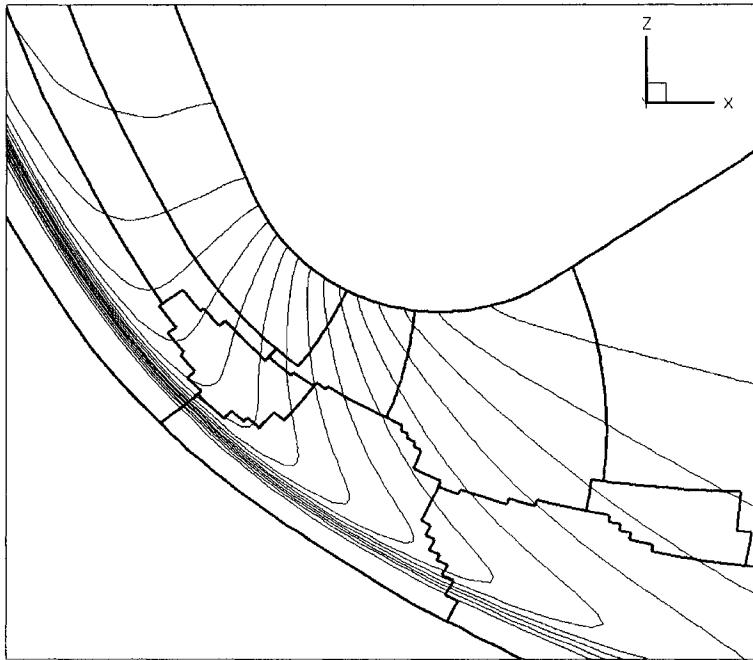


Figure 4.16: Property contours unaffected by mesh partitioning.

4.4.2 Speed-Up Measurements

The parallelized code speed-up is assessed by running the same problem using 1, 2, 4, 8 and 16 processors on two different parallel supercomputers. The problem used in this study is a 3D viscous simulation of an Apollo experimental model using a computational grid made of 93,600 cells. The simulation includes chemical reactions and thermal non-equilibrium effects. The first parallel supercomputer is the cluster NYX at the University of Michigan. This cluster is equipped with 762 AMD Opteron

processors connected through a non-blocking gigabit network. The other parallel supercomputer used is NASA's COLUMBIA cluster which is composed of 20 nodes with 512 Itanium processors each. Within each node of Columbia, processors are connected by the SGIRMK NumaLink fabric for fast inter-processor communication.

The results for this speedup study are presented in Fig. 4.17. It can be observed

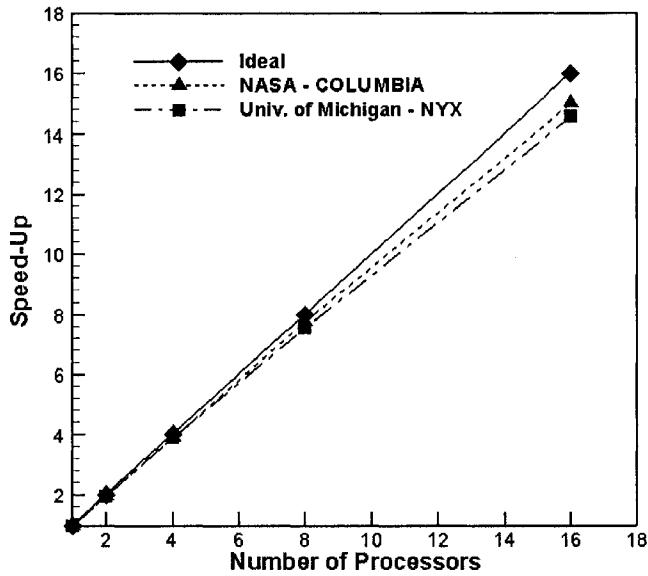


Figure 4.17: Computational speed-up of the numerical code in different parallel machines.

that in both clusters the speedup curves slowly distance themselves from the ideal curve as the number of processors increases. That happens because the amount of communication work in relation to the actual computational work increases. This behavior is similar to the increase in the surface area to volume ratio of a sphere as the size of the sphere decreases. As the size of each partition, or volume of computational work, decreases, the amount of communication, or inter-processor surface area, increases. Because the processors in COLUMBIA are connected with a much faster network, the loss in speedup is a little smaller than in NYX. Nevertheless,

the loss in efficiency in both clusters is small even for this relatively small problem.

CHAPTER V

RESULTS

5.1 Introduction

In this chapter, several physical and numerical aspects of the code are explored by simulating different geometries and conditions for which there are available data for comparisons. The first data set, an Apollo experimental model, is used to evaluate the fluid dynamics model used in the code by comparing calculated aerodynamic coefficients with available experimental data. The second set, a Mars entry spacecraft model, is used to evaluate the chemical and non-equilibrium models by comparing calculated and experimental values for heat transfer. The third set, the RAM-C II spacecraft, is used to evaluate the modeling of weakly ionized gases. The data used in comparisons are the electron number density profiles. The fourth set, the FIRE-II spacecraft, is used to evaluate the physical modeling of higher energy flows where ionization is more significant. In this case, the calculated results are compared against other established NASA codes currently used in the design of thermal protection systems. A sensitivity analysis of radiative heating to chemistry and non-equilibrium models is also performed.

5.2 Apollo Experimental Model

Several experimental studies of hypersonic flows were performed during the 1960's to validate engineering models used in the design of the Apollo spacecraft. One of these studies was performed in the Arnold Engineering Development Center (AEDC) tunnel F during 1966-1967 [32, 15] for a scaled model of the Apollo Command Module. One of the main objectives of this study was to obtain the aerodynamic coefficients of the spacecraft which are important for reentry trajectory analysis. The geometry of the spacecraft experimental model is presented in Fig. 5.1. The nose radius, R_n , is 168.86 mm, the base radius, R_b , is 70.36 mm, the cap radius, R_c , is 8.37 mm, the shoulder radius, R_s , is 7.04 mm, the length of the capsule, L , is 123.28 mm and the cone angle, α_c , is equal to 33 degrees.

The freestream conditions and wall temperature for the numerical simulations are presented in Table 5.1. These values were chosen to match the conditions of the experimental study. The wall temperature is assumed constant and equal to the ambient temperature because each test run lasted for only a few milliseconds. The freestream conditions are such that neither chemical reactions nor strong thermal non-equilibrium effects are observed in the flow field. This is a very good test case for the inviscid and viscous fluxes calculations in the code because it is a very high Mach number flow but it does not present other complicated physical phenomena.

Table 5.1: Freestream Conditions.				
Gas	M_∞	$T_\infty(K)$	$\rho_\infty(kg/m^3)$	$u_\infty(m/s)$
N_2	20.00	43.21	7.817×10^{-4}	2680

This set of conditions is simulated using only hexahedra cells because numerical results are very sensitive to the alignment of the shock wave with the grid. The two

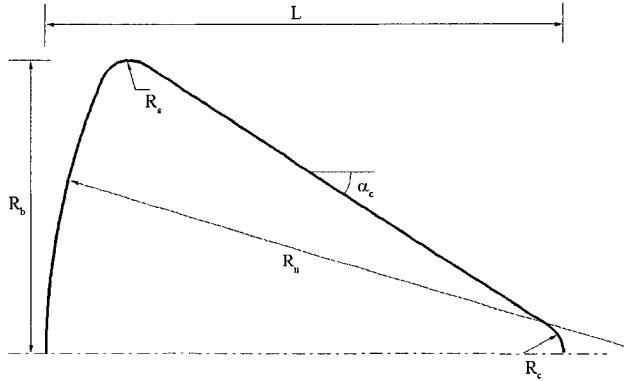


Figure 5.1: Apollo experimental model geometry.

hexahedra meshes used in this work are generated using the commercial software GAMBIT[6]. The first one, for angles-of-attack in the range from 180 to 165 degrees, has 93,600 cells and can be observed in Fig. 5.2. The second one, used for simulations with angles-of-attack in the range from 160 to 150 degrees, has 92,850 cells and is shown in Fig. 5.3. A special modification is performed close to the geometry symmetry point to avoid numerical oscillations that are common there [31, 90]. The modification employed in this work is to use an unstructured grid made of hexahedra in that region which is extremely easy to generate using GAMBIT. The quadrilateral surface grid in that region can be observed in Fig. 5.4. Another grid, with 180,000 grids is generated for a grid refinement study.

Figure 5.5 shows x-velocity contours in the pitch plane around the Apollo experimental model at an angle-of-attack of 150 degrees. One can observe the expected features for this type of flow, which are the strong detached shock wave in front of the

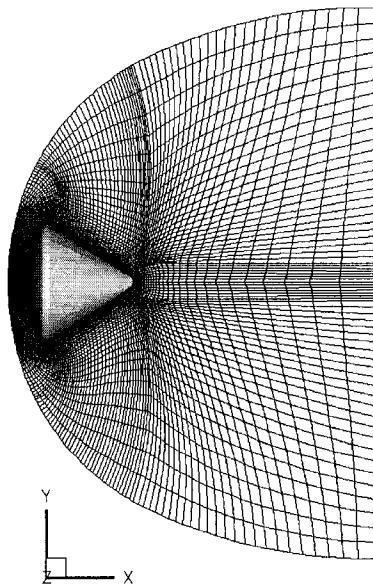


Figure 5.2: Mesh used for angle-of-attacks from 180 to 165 degrees.

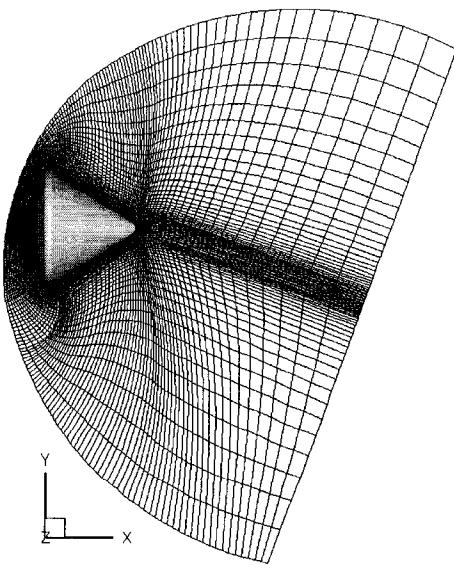


Figure 5.3: Mesh used for angle-of-attacks from 160 to 150 degrees.

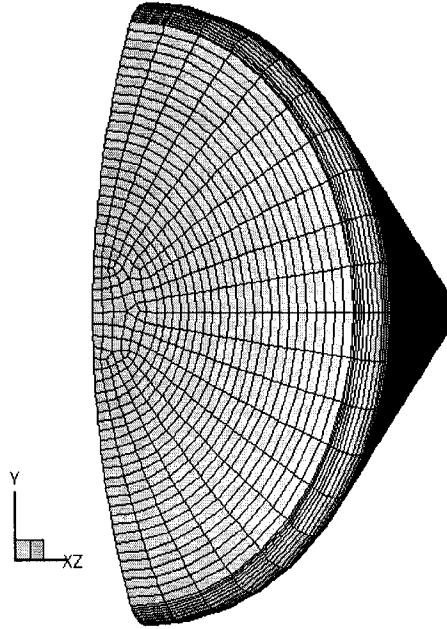


Figure 5.4: Detail of the surface mesh close to the symmetry point.

body, the expansion around the shoulder and the recirculation on the lee side. The bottom half of the detached shock wave presents a small discontinuity in slope. That is due to a difference in grid spacing between blocks, which highlights the difficulties encountered in structured grid generation for a 3D geometry.

The contours of temperature in the pitch plane around the spacecraft model at an angle-of-attack of 180 degrees are shown in Fig. 5.6. This figure shows that the maximum temperature which occurs in front of the spacecraft is around 3000 K which is not enough to cause dissociation of N_2 . The figure also shows the streamlines around the spacecraft, which indicates a small recirculation region on the model apex. The numerical results obtained for pressure and shear stress at the wall are integrated in order to obtain the axial and normal force coefficients, which are compared to experimental results[32, 15] in Figs. 5.7 and 5.8 respectively. The axial force coefficient decreases in modulus and the normal force coefficient increases in modulus

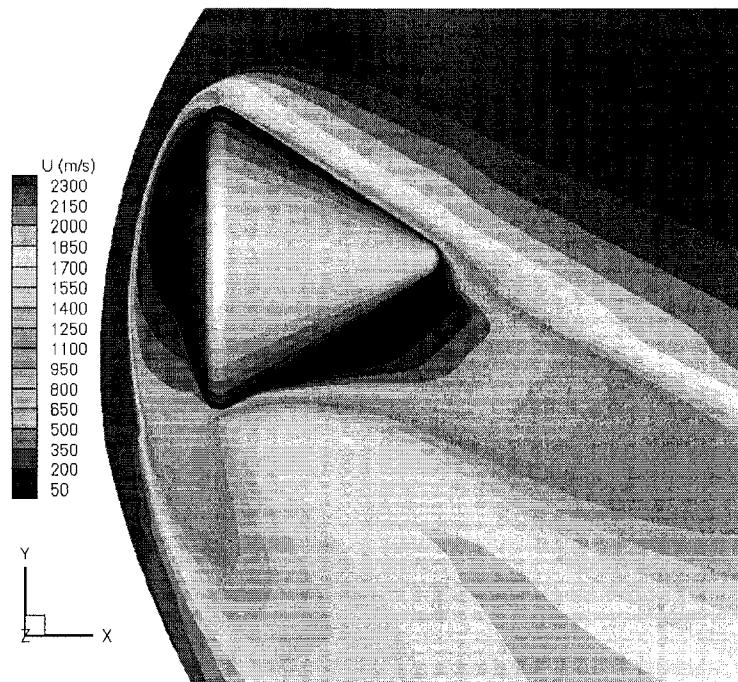


Figure 5.5: X-velocity component contours for $\alpha = 150$ degrees.

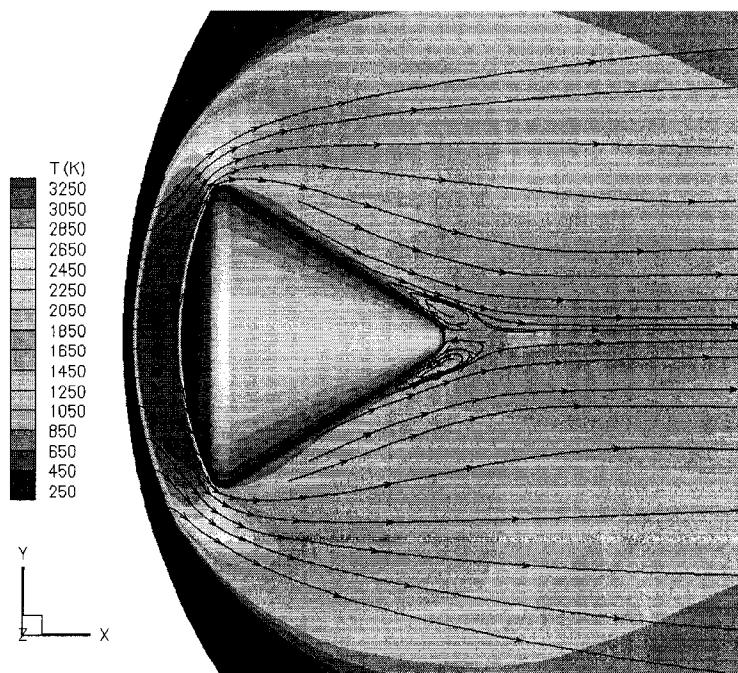


Figure 5.6: Temperature contours for $\alpha = 180$ degrees.

as the angle-of-attack increases. The agreement of the numerical solution with the experimental results is remarkably good. Results for the grid refinement study are also included in those figures. The grid refinement study is performed only for the 180 degrees angle-of-attack case. For the axial and normal force coefficients there is virtually no difference in results when using the finer grid. It is concluded that the coarser grid is adequate for the aerodynamic coefficient computations and therefore is used for all the angles-of-attack in this study.

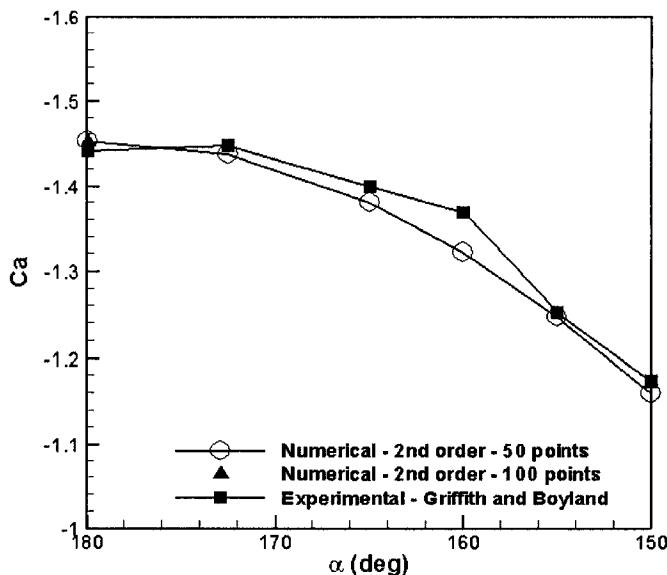


Figure 5.7: Axial force coefficient.

The agreement is also good for the pitch moment, which is compared to experimental results in Fig. 5.9. The result using the finer grid for the 180 degrees angle-of-attack is also in close agreement to the result obtained using the coarser grid.

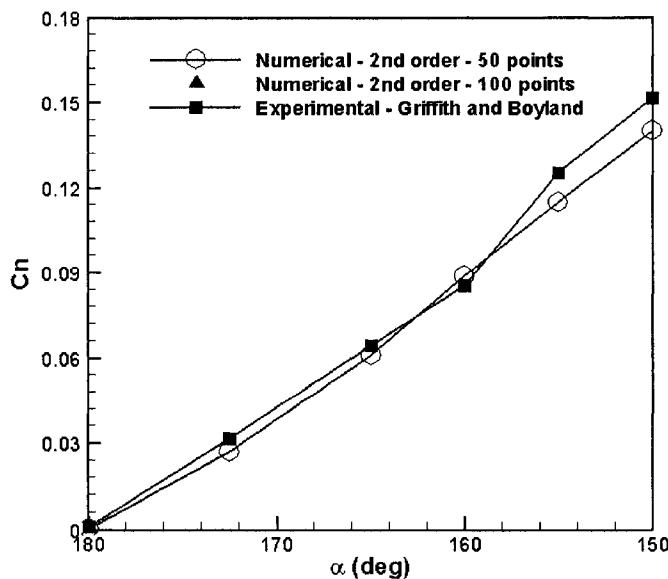


Figure 5.8: Normal force coefficient.

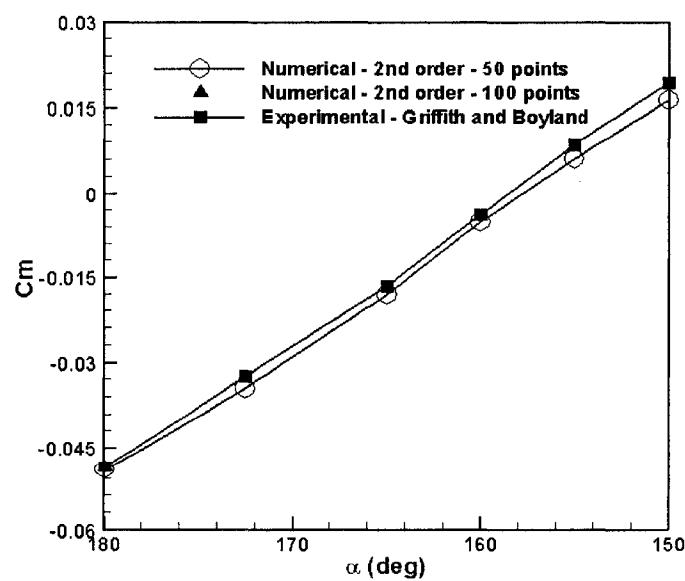


Figure 5.9: Pitching moment coefficient.

5.3 Mars Entry Spacecraft Model

This section studies the flow over a Mars entry spacecraft experimental model used in the HYPULSE expansion hypersonic wind tunnel[42, 43]. The experimental study focused on heat transfer measurements along the experimental model for air and carbon dioxide flows. The main objective of the study was to increase the aerothermodynamics database for validation of numerical tools used in the design of Mars entry spacecrafts. Most experimental results for hypersonic flows are obtained using shock tube wind tunnels. The shock tube tunnel flow that reaches the test section is usually in thermo-chemical non-equilibrium because of the extreme high pressures and temperatures at the reservoir and the series of shocks used to accelerate the flow. It is complicated to determine the actual composition and temperature of the flow that reaches the model, making comparisons between numerical and experimental results difficult[65]. The HYPULSE wind tunnel provides a cleaner flow in the test section than a shock tube tunnel by keeping the test section at very low pressures[42]. It still uses a shock to increase the pressure and temperature of the test gas but part of the acceleration is due to an expansion caused by the low test section pressure. Hence, for the same acceleration, the compression of the test gas does not need to be as high as in a shock tube yielding a flow at the test section closer to equilibrium.

The experimental model used in the study was a standard 70 degrees blunted cone. This shape has been used extensively for Mars entry including the pioneering Viking spacecraft. The geometry of the experimental model is presented in Fig. 5.10, where the nose radius, R_n , is 12.7 mm, the base radius, R_b , is 25.4 mm, the corner radius, R_c , is 1.27 mm, the frustum radius, R_f , is 15.24 mm, the sting radius, R_s , is

10.32 mm, the sting length, L_s , is 116.84 mm, the nose cone angle, α_n , is 70 degrees and the frustum cone angle is 40 degrees.

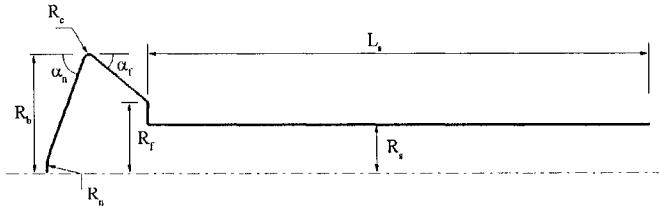


Figure 5.10: Mars entry spacecraft model geometry.

5.3.1 Air Flow

The freestream conditions for the air flow case are presented in Table 5.2. Any dissociation of the freestream flow is neglected. The base diameter Reynolds and Knudsen numbers indicate that the flow is laminar and in the continuum regime. In this work, only the zero angle-of-attack case is studied even though the experimental data set has results for several angles-of-attack. The main purpose of this simulation is to check the chemistry and non-equilibrium modeling which can be accomplished with an axisymmetric simulation.

The meshes used in this computation are generated using GAMBIT[6]. The meshes are made of quadrilaterals only, even though the code has the capability to use triangles in a 2D simulation. Generating a good mesh using only quadrilaterals

Table 5.2: Conditions for the Mars entry spacecraft experimental model air-flow case.

Freestream Condition	Value
Y_{N_2}	7.670×10^{-1}
Y_{O_2}	2.330×10^{-1}
Y_{NO}	0.0
Y_N	0.0
Y_O	0.0
$T_\infty(K)$	1113.0
$\rho_\infty(kg/m^3)$	5.710×10^{-3}
$u_\infty(m/s)$	5162.0
$T_w(K)$	300.0
M_∞	7.9
Re_d	3.39×10^4
Kn	2.8×10^{-4}

for this configuration is somewhat difficult. It is necessary to divide the domain into many blocks in order to ensure smooth mesh growth and to avoid abrupt changes in mesh orientation. The block division used in this work has 9 blocks and is shown in Fig. 5.11. One of the generated meshes is in Fig. 5.12. It has 200 cells in the direction normal to the body and 340 cells in the streamwise direction, totaling 68,000 cells.

Figure 5.13 presents the grid refinement study performed using the convective heating rate along the experimental model forebody. The convective heating rate is chosen as the quantity for this study because its value is very sensitive to the computational grid. The grid refinement study focus on the normal direction because the results are much more sensitive to the refinement in that direction. Three different grids with 100, 150 and 200 points in the normal direction are used. The grid with 100 points in the normal direction clearly under-predicts the convective heating rate. The results for the grids with 150 and 200 points in the normal direction are very close to each other and the solution on both grids could be used as grid independent solutions. The results presented in this section use the grid with 200 points in the normal direction. The level of refinement used in this simulations is close to what

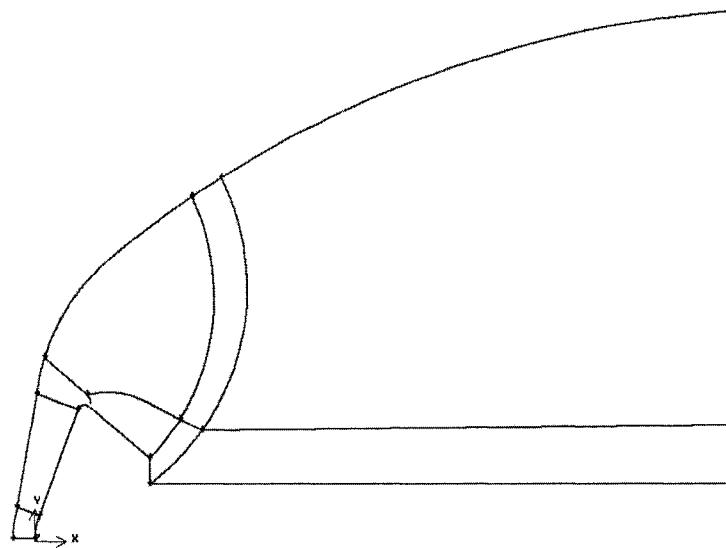


Figure 5.11: Block structure used for generating the grid over the Mars entry spacecraft model.

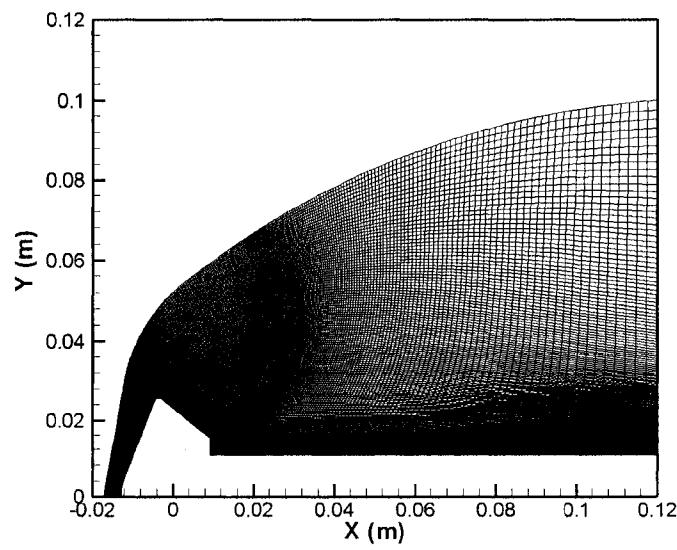


Figure 5.12: Mesh used in the simulations of a Mars entry spacecraft model.

other researchers have used[43].

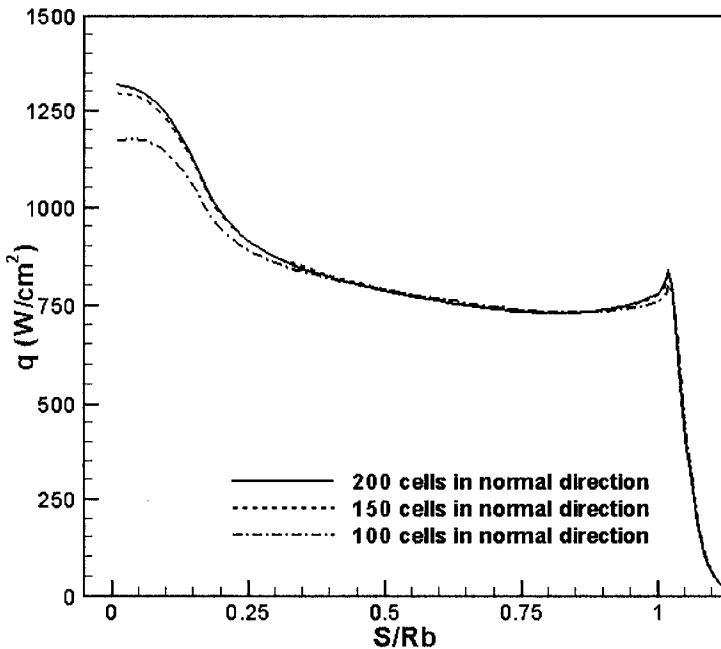


Figure 5.13: Convective heat transfer rate along a Mars entry spacecraft model.

Figure 5.14 shows the flow pattern around the model. The main features in the flow are the detached bow shock in front of the model, the expansion wave formed on the shoulder of the model and a strong recirculation zone with 3 vortices that covers most of the leeside of the model and part of the sting. The detachment point is located just after the shoulder of the model. The flow pattern is in good agreement with the numerical results in Ref. [43] for the same configuration.

The level of non-equilibrium in the flow can be evaluated from Fig. 5.15, which plots the ratio of the translational-rotational to the vibrational temperature, T_t/T_v . From the figure, this ratio can reach values as high as 3.6 in the shock wave and as low as 0.3 inside the expansion fan and close to the body on the leeside. This behavior is further highlighted in Fig. 5.16, which shows the temperatures along the

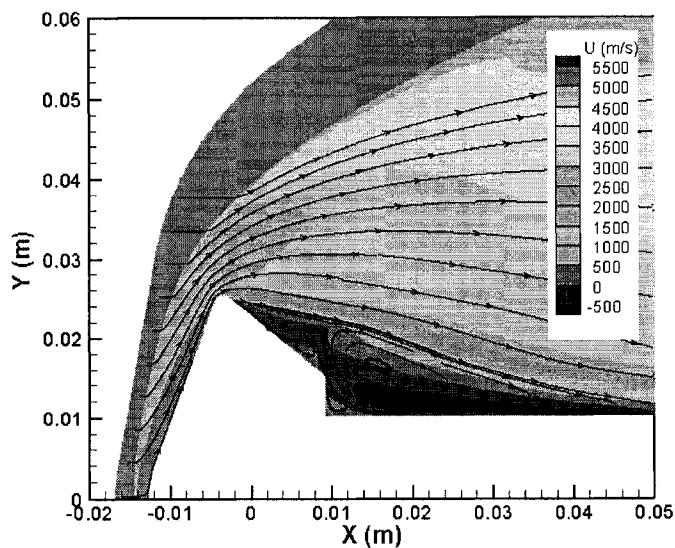


Figure 5.14: X-velocity contours and streamlines over the model.

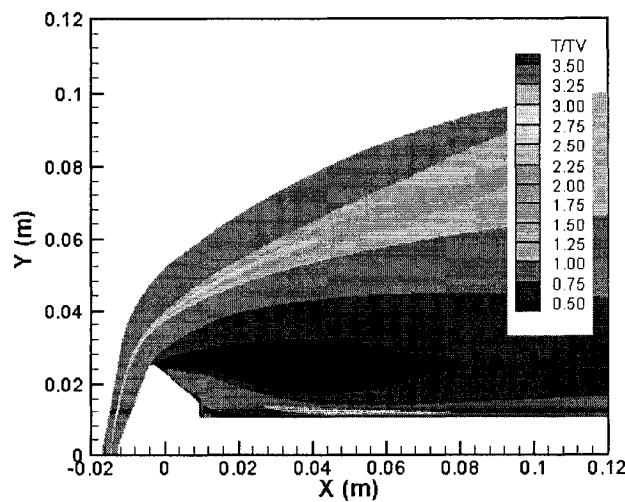


Figure 5.15: Ratio of translational-rotational to vibrational temperatures.

stagnation streamline. The maximum translational temperature is around 11,000 K due to the high freestream Mach number and high freestream temperature. The vibrational temperature lags the translational temperature because the energy transfer between the translational and the vibrational modes occurs at a finite rate. The vibrational temperature does not achieve values as high as the translational temperatures because of the relatively slow energy transfer process and because the air dissociates after the shock wave lowering the translational temperature of the flow.

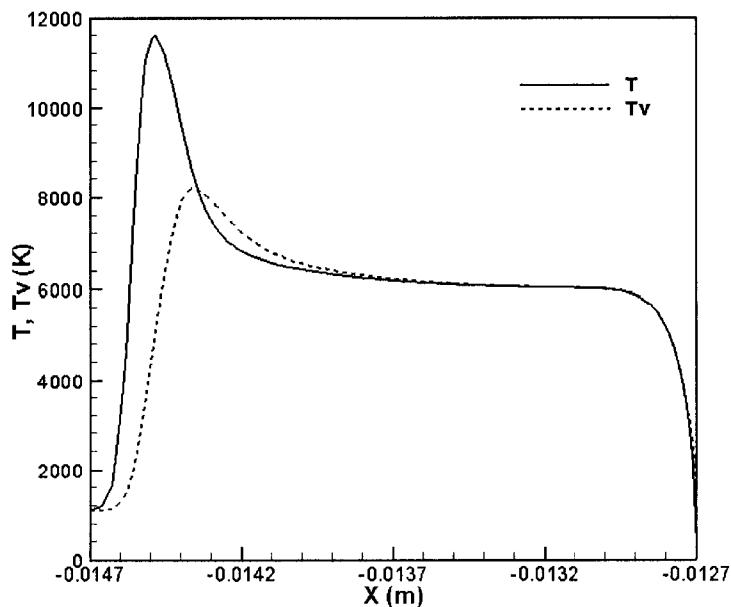


Figure 5.16: Translational and vibrational temperatures along the stagnation streamline of the Mars entry experimental model.

A closer look at the air dissociation is provided in Fig. 5.17, which shows the species mass fractions along the stagnation streamline. It is possible to observe the sudden rise of the dissociation products after the shock due to the high temperature. This figure also shows a decrease in N and O mass fractions close to the body, a sign that there is strong recombination inside the boundary layer. The recombination in the boundary layer releases energy and, therefore, increases the heat transfer to the

wall. It should be noted that the wall is considered non-catalytic. The recombination occurs inside the boundary layer and is due to the decrease in temperature caused by the low wall temperature. It is also possible to see that O_2 starts to dissociate slightly earlier than N_2 and that it is almost totally dissociated after the shock. On the other hand, N_2 does not fully dissociate due to the temperature after the shock not being high enough. The mass fractions of N and O rapidly decrease inside the boundary layer due to the decrease in the temperature, forming N_2 , O_2 and NO.

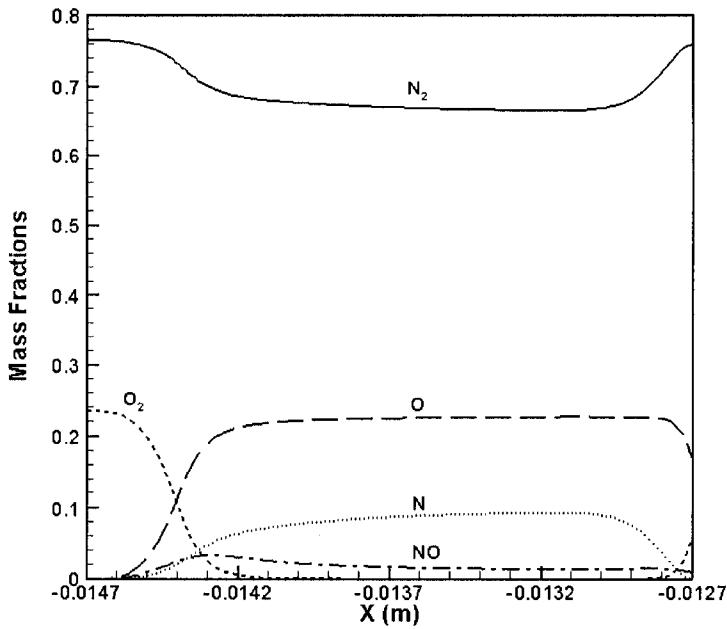


Figure 5.17: Mass fractions along the stagnation streamline of the Mars entry experimental model.

Results for heat transfer are compared to the experimental results of Refs. [41] and [42] and numerical results of Ref. [43]. The comparison for the forebody is in Fig. 5.18. The experimental and numerical results are in very good agreement when using a 2nd order numerical scheme. The first order scheme clearly fails close to the symmetry axis. The numerical results of Ref. [43] are a little lower than the experimental results at the stagnation point and consistently lower than the

present numerical simulations. Nevertheless, both numerical solutions are within the experimental uncertainty for most of the forebody. The comparison for the afterbody

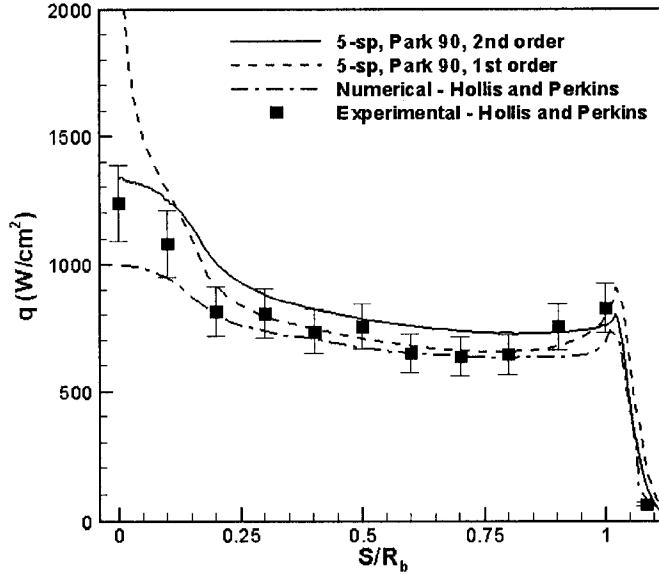


Figure 5.18: Heat transfer on the model forebody.

is shown in Fig. 5.19. The figure shows that the code is able to capture the overall shape of the heat distribution, but the agreement in detail is not as good. The main difference is that the numerical computation underestimates the value for the maximum heat transfer over the sting, which starts around $S/R_b = 2.0$. Some small differences are also present between $S/R_b = 1.7$ and $S/R_b = 2.0$, which corresponds to the back wall of the model. That is a complicated part of the flow where the 3 vortices strongly influence the heat transfer calculations. For instance, the peak in heat transfer at $S/R_b = 1.7$ corresponds to where the large vortex impinges on the corner of the back wall, as can be seen in Fig. 5.20 which shows contours of temperature in that region. This causes an increase in the temperature gradient and, therefore, in the heat transfer. The numerical results obtained on the afterbody

region are in very good agreement with the numerical results obtained in Ref. [43] which are also shown in Fig. 5.19.

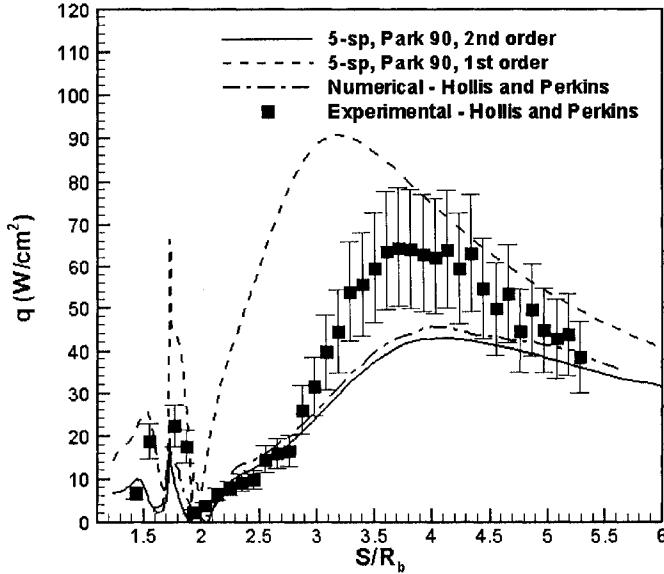


Figure 5.19: Heat transfer on the model afterbody.

5.3.2 Carbon Dioxide Flow

The conditions for this test case are presented in Table 5.3. The flow entering the test section is assumed to be pure CO₂. This numerical simulation uses a 8 species model for Martian atmosphere[63]. The species in that model are CO₂, CO, N₂, O₂, NO, N, O, and C. Because the stream is pure carbon dioxide, species involving nitrogen can be left out of the simulation. Carbon dioxide is not diatomic and it has additional degrees of freedom. The numerical code was specially modified to handle the additional degrees of freedom of vibration. For CO₂, the vibrational energy is given by[17]

$$e_v = \sum_r g_r \frac{R}{M} \frac{\theta_{v,r}}{e^{\theta_{v,r}/T} - 1} , \quad (5.1)$$

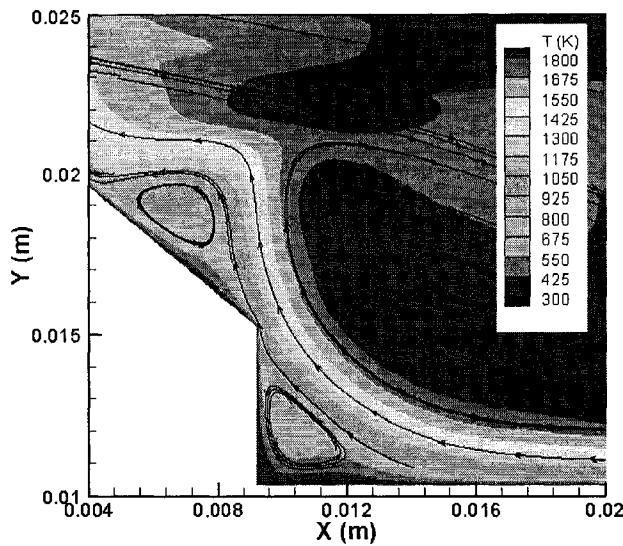


Figure 5.20: Temperature countours and streamlines over the model afterbody.

Table 5.3: Conditions for the Mars entry spacecraft experimental model carbon dioxide-flow case.

Freestream Condition	Value
Y_{CO_2}	1.0
Y_{CO}	0.0
Y_{N_2}	0.0
Y_{O_2}	0.0
Y_{NO}	0.0
Y_N	0.0
Y_O	0.0
Y_C	0.0
$T_\infty(K)$	1088.0
$\rho_\infty(kg/m^3)$	5.79×10^{-3}
$u_\infty(m/s)$	4772.0
$T_w(K)$	300.0
M_∞	9.7
Re_d	3.35×10^4
Kn	3.7×10^{-4}

where the subscript r stands for the vibrational modes of CO_2 and g_r is the degeneracy of each mode. The values of the characteristic temperature of vibration and the degeneracies for each vibrational mode are in Table 5.4.

Table 5.4: Vibrational energy data for CO_2 .

r	$\theta_{v,r}$ (K)	g_r
1	945	2
2	1903	1
3	3329	1

The translational-vibrational energy relaxation should take into account that there are three different vibrational modes. The Millikan and White correlation can be used for each vibrational mode to model the energy exchange between that mode and the translational mode. In addition, the vibrational-vibrational energy transfer process for CO_2 should be modeled to account for the energy transfer between the three modes. However, it is observed[17] that the energy transfer between the vibrational modes in CO_2 is very fast. Assuming that they are in equilibrium with each other, the overall energy transfer between the translational mode and all the vibrational modes will be set by the fastest translational-vibrational exchange, which happens for the $\theta_v = 945$ K mode. Thus, the entire translational-vibrational and vibrational-vibrational exchanges are modeled by using only the translational-vibrational relaxation time calculated using $\theta_v = 945$ K.

In this test case, only the experimental model forebody is simulated. The grid employed in the simulation has a total of 20,400 cells. The mesh has 200 cells in the normal direction and 102 along the body. The grid can be observed in Fig. 5.21. It is a subset of the grid used for the full body simulations in air. A general overview of the flow field is in Fig. 5.22 where the detached shock wave characteristic of blunt

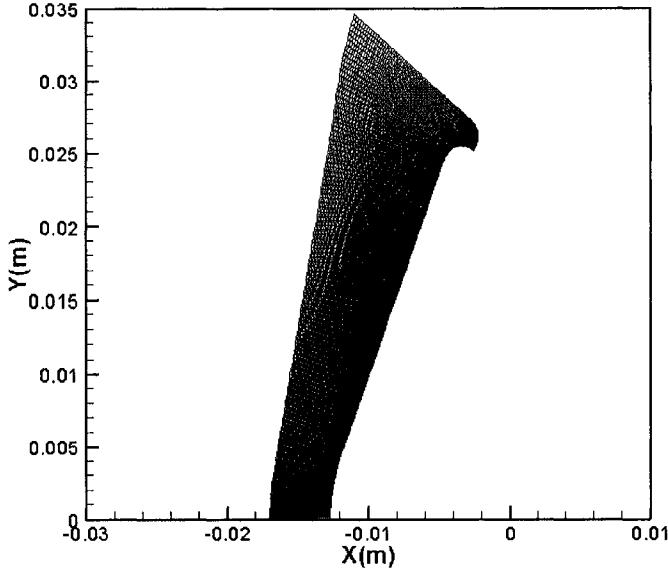


Figure 5.21: Computational grid used for CO₂ flow over the Mars entry experimental model forebody.

bodies and the expansion around the model shoulder can be observed. The shock stand-off distance for CO₂ flow is much smaller than for air flow. This fact can also be observed by comparing the temperature profiles along the stagnation line for air in Fig. 5.15 and for CO₂ in Fig. 5.23. The reason for the smaller shock standoff distance is the higher density in the shock layer, which is caused by a smaller temperature. In air, the shock layer temperature is around 6,000 K while in CO₂ the shock layer temperature is around 4,000 K. The smaller temperature occurs because CO₂ is easier to dissociate than N₂. This fact can be observed in Fig. 5.24 which shows the species mass fractions along the stagnation line. Inside the shock layer, around 60% of CO₂ has dissociated versus only 14% of N₂ in air as shown in Fig. 5.17. The mass fractions along the stagnation streamline show that CO₂ dissociates into CO and O and that some of the oxygen atoms recombine into O₂. Finally, Fig. 5.25 shows the heat transfer rates along the experimental model. The numerical results are in good

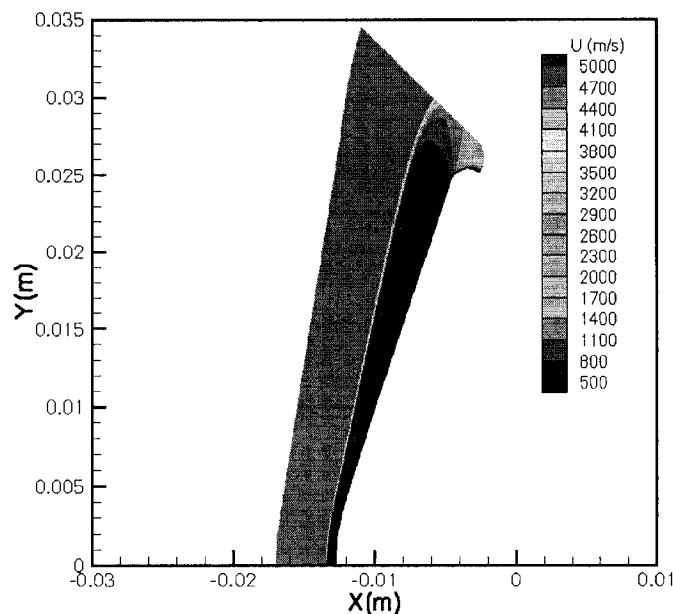


Figure 5.22: X-velocity contours over the Mars entry experimental model forebody for CO_2 flow.

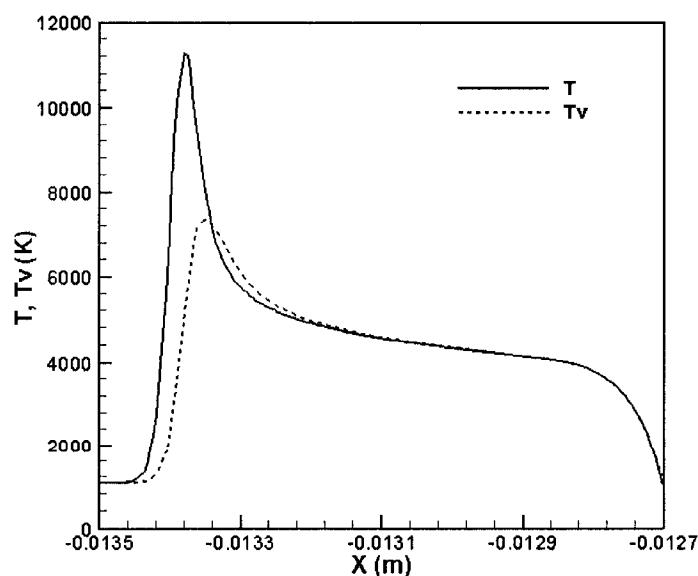


Figure 5.23: Temperature profiles along the stagnation streamline for CO_2 flow.

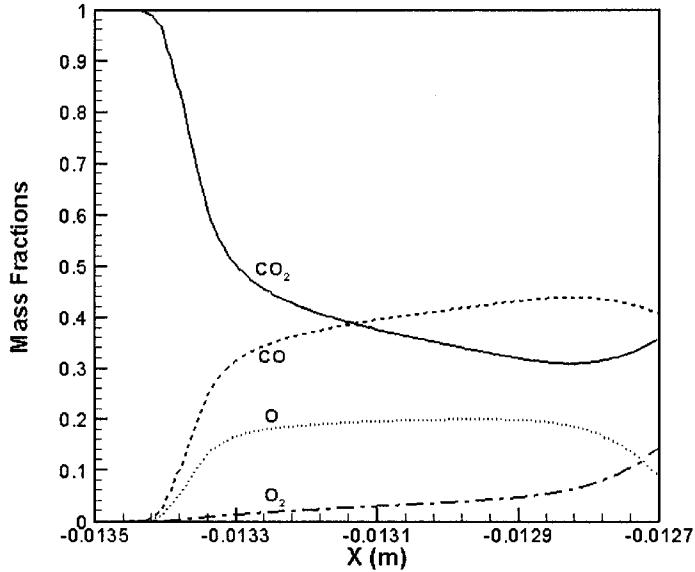


Figure 5.24: Species mass fractions profiles along the stagnation streamline for CO_2 flow.

agreement with the experimental data and numerical results from Ref. [43].

5.4 RAM-C Spacecraft

The Radio Attenuation Measurement (RAM) experiment was a series of hypersonic flights performed in the 1960's to study communications blackout. The communication blackout is caused by the presence of a plasma sheath around a spacecraft that attenuates the radio signals arriving at and leaving the spacecraft. It is a common phenomenon in reentry vehicles due to the ionization caused by the high temperature in the shock layer. From all the flights performed, the RAM-C II is the most useful for numerical comparisons. This flight performed measurements of the maximum electron number density normal to the surface at 4 different stations along the body using reflectometers. A rake of electrostatic probes measured the variation of electron number density close to the rear of the spacecraft.

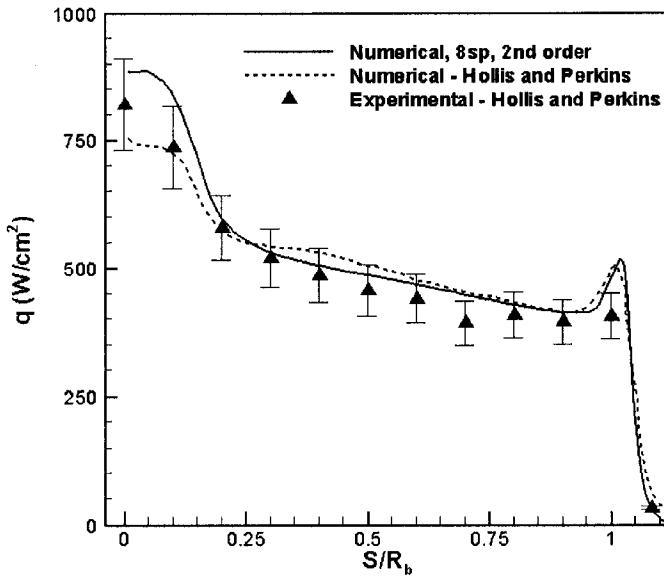


Figure 5.25: Heat transfer along the Mars entry experimental model forebody for CO_2 flow.

The presence of electrons and ions in the flow requires a differentiated treatment of charged and non-charged particles when calculating transport properties and non-equilibrium models. The RAM-C II flight test conditions are simulated in order to demonstrate the capabilities of the physical modeling of weakly ionized flows implemented in the code. The RAM-C II flight data is a good source of measurements for code comparisons and has been used extensively for code comparisons[18, 30, 45]. In this work, the flow condition corresponding to the 71km altitude trajectory point is simulated. The freestream conditions are tabulated in Table 5.5. The condition satisfies the limit for a continuum approximation and it is in the laminar regime as can be concluded from the nose radius Reynolds and Knudsen numbers. The geometry of the RAM-C II spacecraft, a blunt cone, is shown in Fig. 5.26. The nose radius is $R_n = 0.1524$ m, the cone angle is $\alpha_c = 9$ deg. and the vehicle length is $L = 1.3$ m.

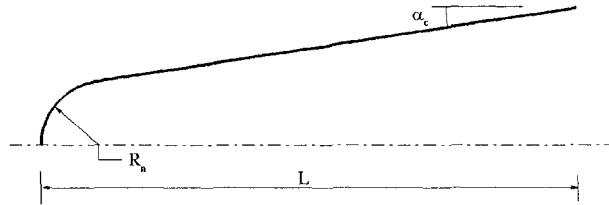


Figure 5.26: RAM-C geometry.

Table 5.5: Freestream Conditions.

Freestream Condition	Value
Y_{N_2}	7.670×10^{-1}
Y_{O_2}	2.330×10^{-1}
Y_{NO}	0.0
Y_N	0.0
Y_O	0.0
Y_{NO^+}	0.0
Y_{e^-}	0.0
$T_\infty(K)$	254.0
$\rho_\infty(kg/m^3)$	2.70236×10^{-4}
$u_\infty(m/s)$	7650.0
$T_w(K)$	1200.0
M_∞	23.9
Re_d	1.95×10^4
Kn	1.2×10^{-3}

Figure 5.27 presents one of the meshes used in this work. It has 128 points in the normal direction and 75 points along the geometry. Two different 7-species air models are used in this simulation[74, 73]. Both air models include N₂, O₂, NO, N, O, NO⁺ and electrons. The transport properties are calculated using Wilke's mixing rule and Blottner model and the diffusion coefficient is calculated using a constant Lewis number. Those approximations are still adequate for a 7 km/s hypersonic flow[29].

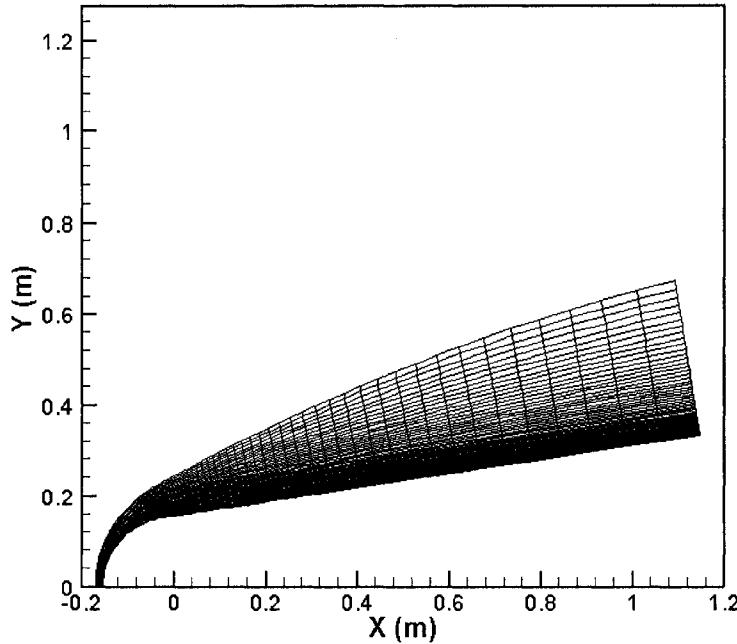


Figure 5.27: Computational grid used for the RAM-C II spacecraft at 71 km altitude simulations.

Figure 5.28 presents the grid refinement study performed using the convective heating rate along the vehicle. As before, the convective heating rate is chosen as the quantity for this study because its value is very sensitive to the computational grid. Three different grids with 64, 96 and 128 points in the normal direction are used. The difference in convective heating between the 3 meshes is very small, as is

expected because other researches reported solutions using much coarser grids[18]. The results presented in this section all use the grid with 128 points in the normal direction.

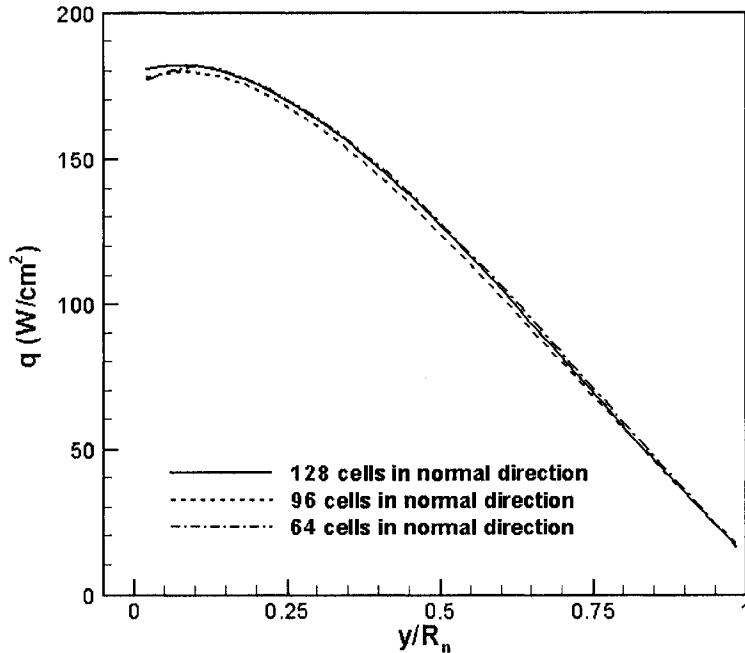


Figure 5.28: Convective heat transfer rate along the RAM-C II spacecraft.

Maximum electron number densities along the direction normal to the vehicle surface are compared to flight data and other numerical simulations in Fig. 5.29. The electron number density is very high at the stagnation region and falls rapidly as the flow temperature reduces due to the expansion along the spheric nose cap as shown in Fig. 5.30. The present numerical results are in good agreement with the numerical results of Grasso and Capano[30] which used a 2 temperature model as well. That work used forward reaction rates from Ref. [74] and equilibrium constants from Ref. [73]. The present results using the two different chemistry models bound the distribution from Grasso and Capano. The numerical results of Candler[18] and

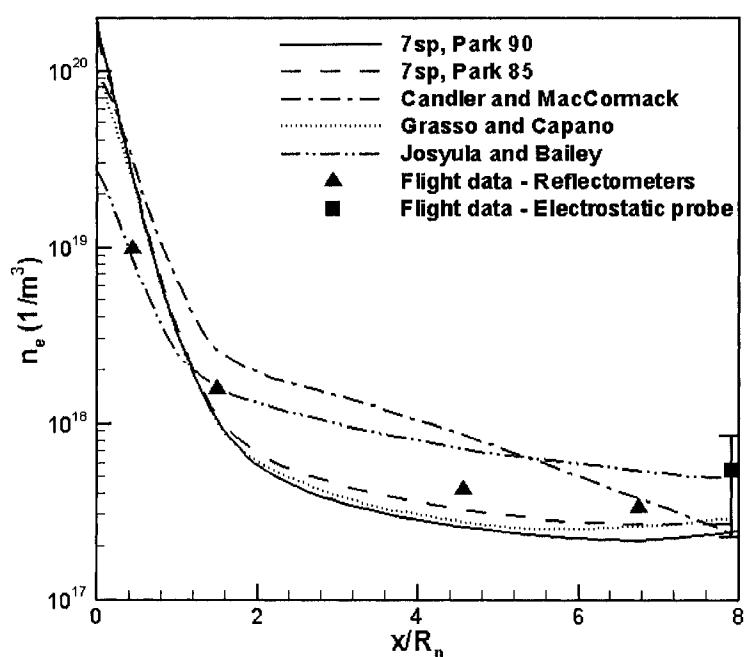


Figure 5.29: Electron number density comparisons along the body for RAM-C II flight test at 71 km altitude.

Josyula and Bailey[45] use a 6 temperature model and, not surprisingly, yield different results from the present simulation. The results by Josyula and Bailey are particularly in excellent agreement with the experimental data but all numerical results lie within the uncertainty of the flight test data evaluated using the electrostatic probe data.

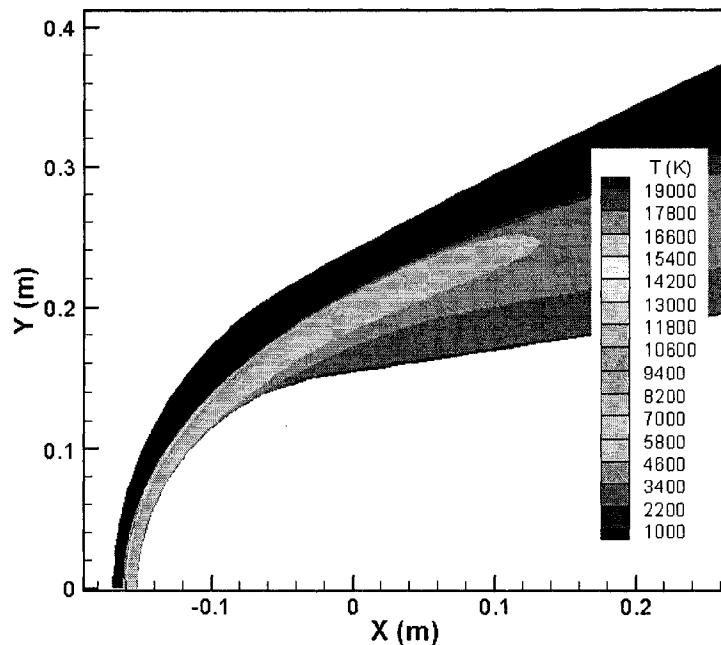


Figure 5.30: Translational-rotational temperature over the RAM-C II spacecraft at 71 km altitude using Park's 1990 chemistry set.

Figure 5.31 highlights the influence of the high temperature correction of Millikan and White's relaxation time suggested by Park[74]. One can observe that without the use of the correction, the energy transfer between the translational-rotational and the vibrational modes is so fast that the vibrational temperature reaches temperatures almost as high as the translational temperature. The high vibrational temperature leads to stronger dissociation in relation to the case using the high temperature correction which causes a smaller shock standoff distance, also observed in Fig. 5.31.

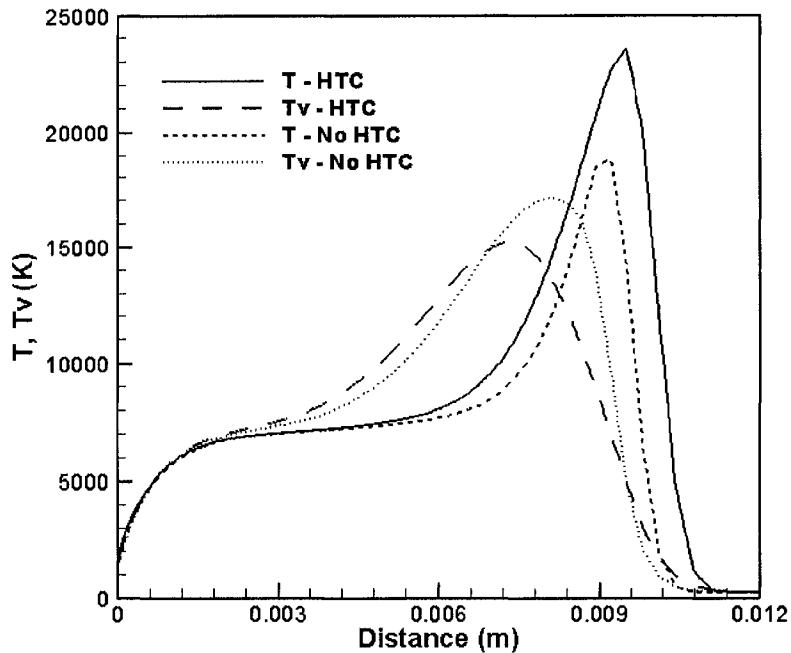


Figure 5.31: Influence of Park's high temperature correction on temperature profiles along the stagnation streamline for the RAM-C II flight test at 71 km altitude.

5.5 FIRE-II Spacecraft

The Flight Investigation of Reentry Environment (FIRE) flight experiment was performed in 1965 in order to address uncertainties of the aerothermodynamic models used to design the Apollo thermal protection system. This flight experiment measured the total heating rate to the spacecraft using calorimeters and the radiative heating rate using radiometers during reentry at lunar-return velocity. Even though flight data is available, the objective of this test case is to compare the numerical results for convective heating calculated using the code developed in this work, LE-MANS, with the results obtained using two different NASA codes. These codes are

the Data-Parallel Line Relaxation code, DPLR, developed at NASA Ames and the Langley Aerothermodynamic Upwind Relaxation Algorithm, LAURA, developed at NASA Langley. Both codes have been used extensively in the simulation and design of entry vehicles. In addition, LeMANS is used to perform a sensitivity analysis of the radiative heating to several thermo-chemical non-equilibrium models.

The FIRE-II thermal protection system used three non-ablating heat shields made of beryllium metal and asbestos impregnated phenolic resin heat shields in between them. The heat shields were ejected at prescribed times so that a new beryllium heat shield was exposed during each measurement period. Each measurement period ended when the corresponding beryllium heat shield started to ablate. The three freestream conditions in this work are the same as in Ref. [37]. They are chosen such that each one corresponds to a different beryllium heat shield. Figure 5.32 shows the geometry of the spacecraft forebody simulated in this work. The conditions and geometry information necessary to define the forebody are tabulated in Table 5.6 where t is the flight time, H is the altitude, R_n is the nose radius, R_s is the shoulder radius and R_b is the base radius. All conditions satisfy the limit for a continuum approximation - $Kn < 0.01$ and all of them are in the laminar regime.

Table 5.6: Freestream Conditions for the FIRE-II cases.

Freestream Condition	Trajectory time (s)		
	1636	1643	1651
H (km)	71.04	53.04	37.19
u_∞ (m/s)	1.131×10^4	1.048×10^4	6.19×10^3
ρ (kg/m ³)	8.57×10^{-5}	7.80×10^{-4}	6.05×10^{-3}
T_∞ (K)	210	276	253
T_w (K)	810	640	1060
R_n (m)	9.347×10^{-1}	8.052×10^{-1}	7.021×10^{-1}
R_s (m)	1.02×10^{-2}	3.56×10^{-2}	0.61×10^{-2}
R_b (m)	3.3575×10^{-1}	3.1495×10^{-1}	2.9395×10^{-1}

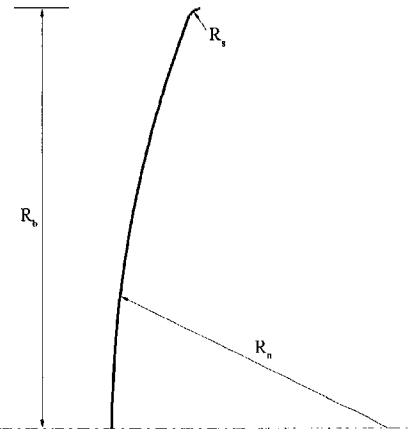


Figure 5.32: FIRE-II geometry.

5.5.1 Convective Heating

A total of 12 simulations are performed using LeMANS and compared to DPLR and LAURA. The simulations used 5 and 11-species air models and catalytic and supercatalytic wall boundary conditions for the 3 different trajectory times listed in Table 5.6. The meshes used in this work were provided by NASA Ames and are the same meshes used by DPLR in Ref. [37]. The results for convective heating are presented in Table 5.7, where the data for DPLR and LAURA are obtained from Ref. [37]. The results for LeMANS lie in between DPLR and LAURA for all conditions. One of the most interesting cases is the 1636, 11-species case which given the high speed and low density, can best highlight the differences between the physical modeling in the codes. For that particular case, the heat transfer calculated using LeMANS is very close to the one calculated by DPLR. That is not too surprising since much of the modeling inside LeMANS and DPLR is the same, including inviscid

fluxes, collision integrals and chemistry rates. Overall, the table shows that the difference between the 3 codes for the 11-species cases is at most 7%, a very good comparison given all the physical and numerical modeling involved. The results are consistent with the total heating rates measured in flight for the three conditions that are 290, 1025, and 390 W/cm² for the 1636, 1643 and 1651 seconds flight times respectively. For the 1636 and 1643 cases, the flight data is above the numerical results which is consistent because the measured data includes radiative heating. For the low altitude case, at 1651 seconds, all numerical codes overpredict the flight data. No investigation on the causes of this inconsistency is performed in this work because the objective of this test is a code to code comparison.

Table 5.7: FIRE-II Convective Heating.

Time (s)	Air Model ^a	Wall ^b	Heating rate (W/cm ²)			Difference (%)
			DPLR	LAURA	LeMANS	
1636	11	NC	183	191	185	4
		C	301	307	307	2
	5	NC	226	204	210	-10
		C	365	340	346	-7
1643	11	NC	769	753	753	-2
		C	805	802	810	-1
	5	NC	823	773	799	-6
		C	904	839	860	-7
1651	11	NC	478	449	473	-6
		C	511	477	503	-7
	5	NC	477	451	473	-5
		C	511	479	505	-6

^aThe air model classification corresponds to the number of species in the model.

^bC and NC represent catalytic and non-catalytic wall respectively.

In order to illustrate the results, the 1636 seconds , 11-species non-catalytic wall case is discussed in more detail. The mesh used in this calculation is presented in Fig. 5.33. It has 18,432 cells with 144 in the normal direction and 128 along the

body. The spacing at the wall is around 2×10^{-6} meters.

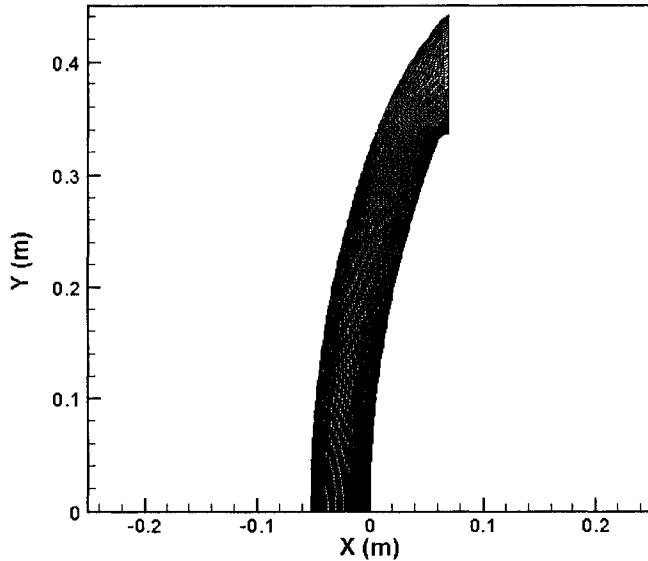


Figure 5.33: Mesh used by DPLR and LeMANS for the 1636 seconds, 11-species cases.

A general overview of the flowfield is presented in Figs. 5.34 and 5.35, which show contours of translational temperature and pressure in the flowfield. The maximum temperature in the flowfield is around 30,000 K occurring immediately after the shock wave. Due to dissociation and ionization reactions, the temperature decreases to around 10,000 K in the shock layer. The pressure contours show the expected blunt body behavior with high pressure at the stagnation region reaching values close to 10,000 Pa and an expansion around the shoulder of the spacecraft. The 2nd order scheme employed in the numerical calculation makes the contour lines very sharp.

The high energy of the flow causes high rates of chemical reactions after the shock wave leading to a drop in temperature after the shock. Figure 5.36 shows the species mass fractions along the stagnation streamline. Closer to the shock wave, the ions with larger mass fractions are N_2^+ , O_2^+ and NO^+ which are formed through

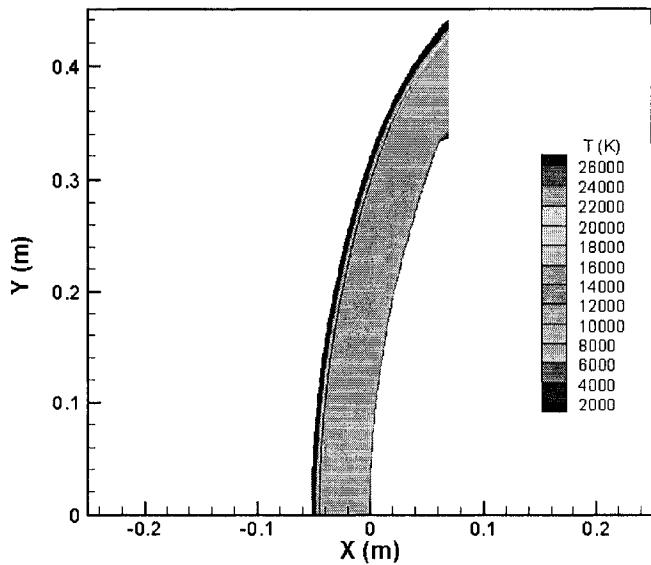


Figure 5.34: Temperature contours around the FIRE-II spacecraft for the 1636 seconds, 11-species, non-catalytic wall case.

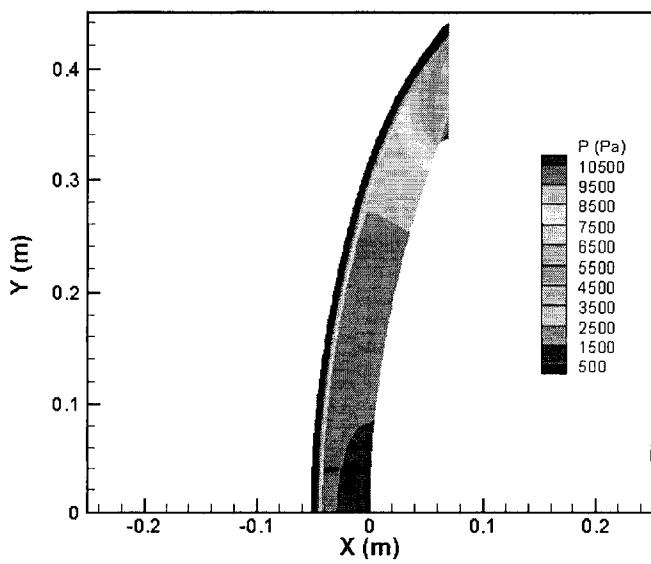


Figure 5.35: Pressure contours around the FIRE-II spacecraft for the 1636 seconds, 11-species, non-catalytic wall case.

associative ionization reactions. The atomic species N, O and their ions N^+ and O^+ are the species with larger mass fractions in the shock layer followed by N_2 . Inside the boundary layer, the decrease in temperature increases the rate of recombination reactions.

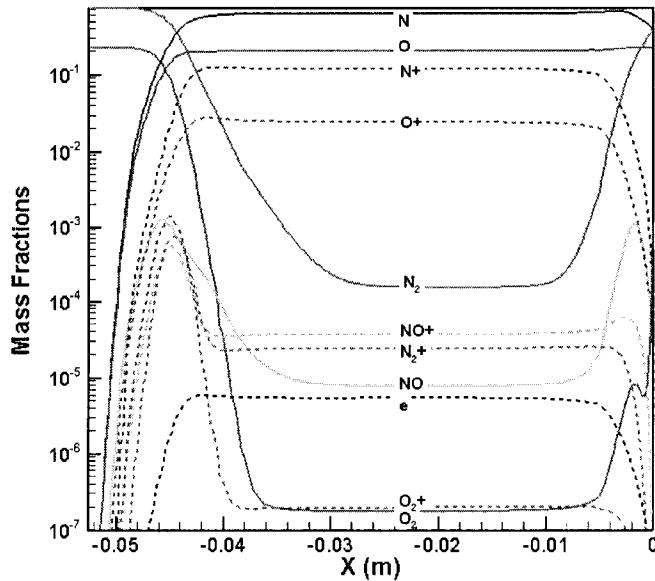


Figure 5.36: Species mass fractions along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case.

Figure 5.37 shows a comparison of heat transfer profiles calculated by LeMANS, DPLR and LAURA. The agreement between LeMANS and DPLR is remarkable. Both codes show a small dip in the profile close to the stagnation point, which could be attributed to a “carbuncle” like behavior. It is well known in the CFD community that grid alignment to the shock wave and the inviscid fluxes play important roles in the appearance of “carbuncles”. Both LeMANS and DPLR use the same grid and same inviscid numerical fluxes for these calculations which may explain why both codes present a similar behavior close to the stagnation point.

A comparison between the translational temperature profiles along the stagnation

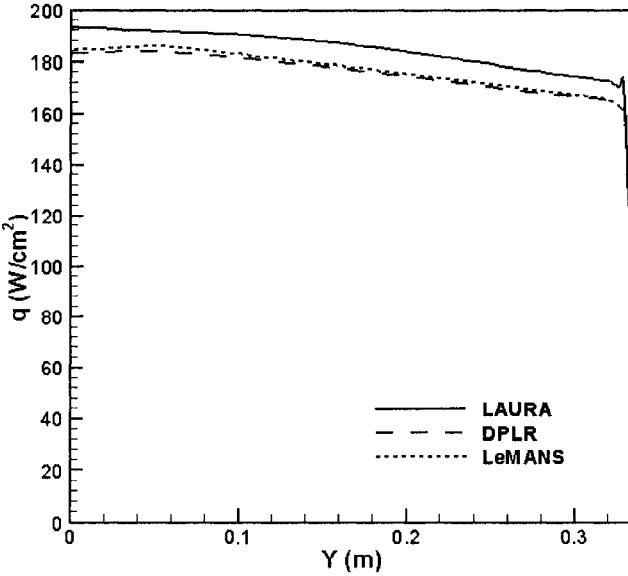


Figure 5.37: Convective heat transfer along the wall for the 1636 seconds, 11-species, non-catalytic wall case.

streamline calculated by the three codes is presented in Fig. 5.38. It can be observed that the three codes present good agreement in the equilibrium temperature after the shock and inside the boundary layer. The codes show distinct behavior at the shock wave. LAURA predicts a broader shock and the largest peak translational temperature while DPLR presents a thinner shock and the smallest peak in translational temperature. LeMANS stands somewhere in between both codes. LAURA's broader shock comes from a coarser grid that has 6,400 cells with 80 in the normal direction and 80 along the body. This grid, shown in Fig. 5.39, is clearly coarser close to the freestream boundary. While LeMANS and DPLR used the same mesh, LeMANS predicts a larger shock standoff distance. That is caused by the modeling of electron impact ionization reactions as discussed in the next section.

Figure 5.40 presents comparisons of the vibrational temperature profiles along the stagnation line calculated by the three codes. While the codes agree on the

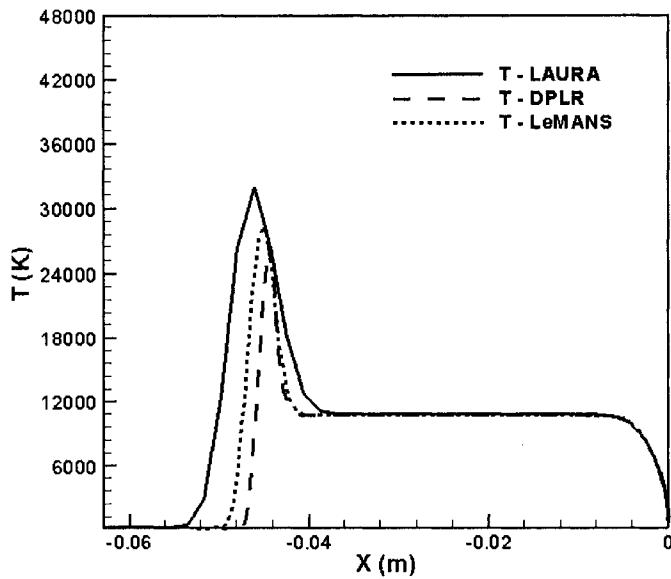


Figure 5.38: Translational temperature profiles along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case.

equilibrium value after the shock, the profiles are different at the shock wave and inside the boundary layer. LAURA and LeMANS predict a somewhat higher value of vibrational temperature inside the boundary layer when compared to DPLR. In addition, LAURA and LeMANS profiles present a peak in vibrational temperature at the shock while the DPLR profile does not. LAURA and LeMANS couple the electron and electronic energy with the vibrational energy while DPLR couples the electron and electronic energy with the translational mode. The peak in vibrational temperature for LAURA and LeMANS is caused by the increase of electron and electronic energy close to the shock wave.

5.5.2 Radiative Heating Sensitivity Analysis

The radiative heating calculations in this study are performed using NEQAIR[93]. NEQAIR is run using the Quasi-Steady State (QSS) assumption in which electronic

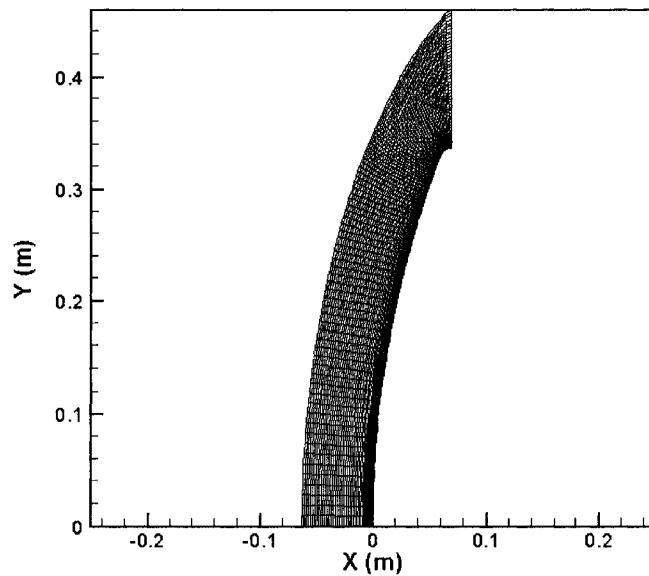


Figure 5.39: Mesh used by LAURA for the 1636 seconds, 11-species cases.

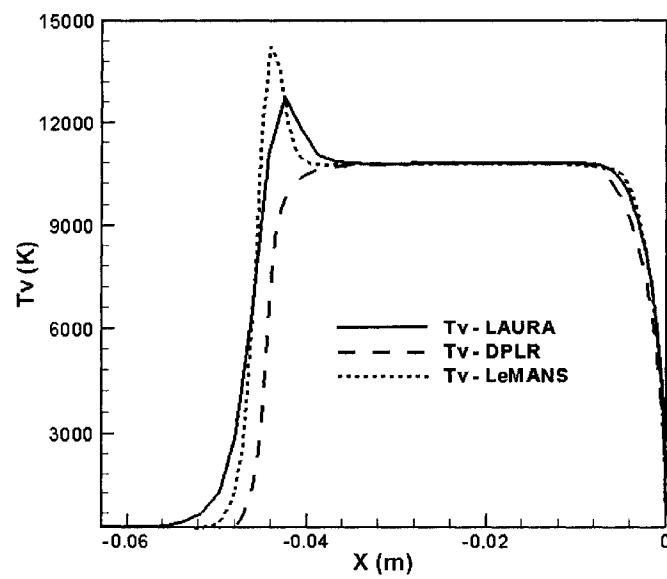


Figure 5.40: Vibrational temperature profiles along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case.

excitation rates are computed using the electron temperature. NEQAIR requires the translational, rotational, vibrational and electron temperatures and species number densities along a line of sight in the flow field to calculate the radiation that reaches the spacecraft wall. In this work, only the stagnation line is used. The 1636 seconds, 11-species case is used for comparing results obtained using the three different CFD codes. Results for DPLR and LAURA are from Ref. [37]. Table 5.8 shows the results from all codes for catalytic and non-catalytic wall conditions including the percentage difference in relation to LAURA results. LeMANS results are again in between those calculated by DPLR and LAURA for the vacuum ultraviolet (VUV) radiation range but about 10% lower for the long wavelength region of the spectrum, which is composed of ultraviolet, visible and infrared (UV+Vis+IR) radiation, for which flight data is available. For $200 \text{ nm} < \lambda < 4 \mu\text{m}$ the flight data radiative heating rate is 28.5 W/cm^2 . The agreement for the radiative heating between the three codes is not as good as for the convective heating. Most of the radiation to the spacecraft during a reentry in the Earth's atmosphere is related to the electron temperature and the number densities of electrons, atomic nitrogen and atomic oxygen. The electron temperature for LAURA and LeMANS is assumed to be in equilibrium with the vibrational temperature while in DPLR the electron temperature is assumed in equilibrium with the translational temperature. All codes assume that the rotational temperature is equilibrated with the translational temperature.

Figure 5.41 compares the electron number density profiles calculated by the three codes. It is clear from this figure that DPLR and LeMANS profiles present a peak close to the shock while the LAURA profile does not. The maximum electron number density calculated by DPLR is much larger than that calculated by LAURA while LeMANS lies somewhere in between. DPLR and LAURA profiles agree very

well inside the boundary layer while the LeMANS profile differs from both in that region. The disagreement comes from extra numerical dissipation in the inviscid fluxes which is switched off closer to the wall in LeMANS than it is in LAURA and DPLR. The extra numerical dissipation is obtained by modifying the inviscid fluxes eigenvalues[16, 21].

The profiles for atomic nitrogen are presented in Fig. 5.42. The difference between the three codes is much smaller in this case. The difference between LeMANS and the other codes in the boundary layer is due to the extra numerical dissipation as mentioned earlier. At the shock wave, the LeMANS and DPLR profiles seem to be in better agreement. The LAURA profile close to the shock is somewhat more diffused. The behavior for atomic oxygen is similar. The number density profiles for this species are presented in Fig. 5.43. The LeMANS profile disagrees with DPLR and LAURA in the boundary layer due to extra numerical dissipation. The LAURA profile is also more diffused at the shock wave.

Table 5.8: Stagnation point radiative heating calculations using different codes.

Code	Heating rate (W/cm^2)			
	Non-Catalytic		Catalytic	
	$\lambda < 200 \text{ nm}$ (VUV)	$\lambda \geq 200 \text{ nm}$ (UV+Vis+IR)	$\lambda < 200 \text{ nm}$ (VUV)	$\lambda \geq 200 \text{ nm}$ (UV+Vis+IR)
LAURA	42.06	28.03	39.24	28.00
Diff.	-	-	-	-
DPLR	49.33	26.02	49.21	25.38
Diff.	17%	-7%	25%	-9%
LeMANS	44.26	24.79	44.48	23.98
Diff.	5%	-12%	13%	-14%

Given the differences in the physical modeling employed by the codes, a sensitivity analysis is performed to understand the reasons for the differences in the stagnation line property profiles. The sensitivity analysis focuses on the thermal non-equilibrium

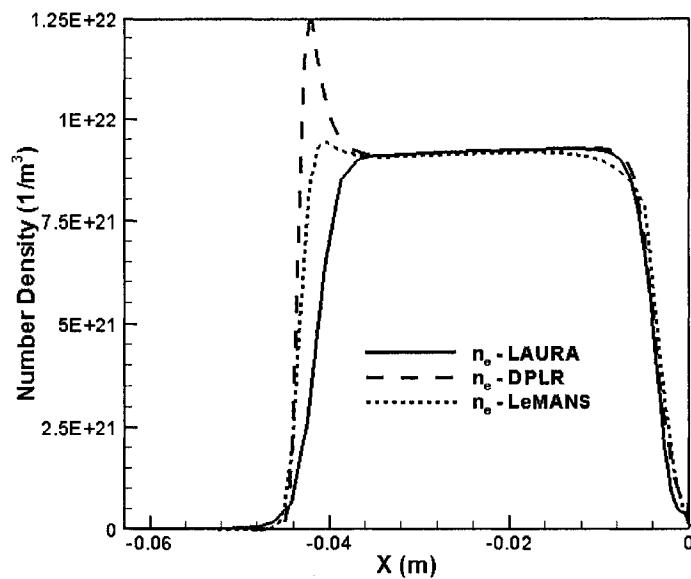


Figure 5.41: Electron number density along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case calculated using different codes.

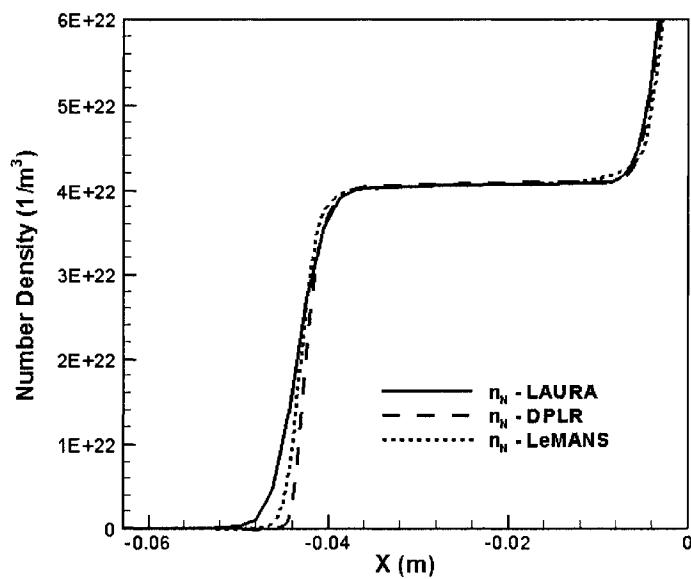


Figure 5.42: Atomic nitrogen number density along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case calculated using different codes.

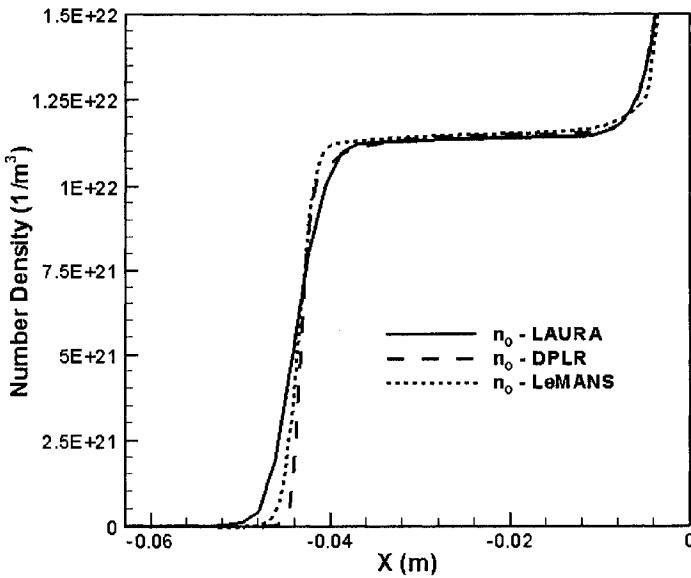


Figure 5.43: Atomic oxygen number density along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case calculated using different codes.

model because the flow is in thermo-chemical equilibrium in the shock layer. The thermal non-equilibrium model impacts the profiles close to the shock wave which is the region where the profiles show larger differences as seen in Figs. 5.38 to 5.43. Table 5.9 shows the impact of changes in some aspects of the thermochemistry modeling on the calculated radiation, including the percentage difference in relation to LeMANS baseline result.

The first change is to simulate a thermal-equilibrium flow. This is achieved by using a constant vibrational-electron-electronic relaxation time of the order 1×10^{-13} seconds. The equilibrium assumption drastically reduced the maximum translational temperature in the flowfield and the shock standoff distance, as can be observed in Fig. 5.44. The reduction in temperature is due to the immediate energy transfer to the other energy modes. The smaller temperature makes the fluid denser, thus

Table 5.9: Stagnation point radiative heating sensitivity analysis for Non-Catalytic wall.

Parameter	Heating rate (W/cm ²)	
	$\lambda < 200$ nm (VUV)	$\lambda \geq 200$ nm (UV+Vis+IR)
Baseline	44.26	24.79
Diff.	-	-
Thermal-Equilibrium	43.48	24.34
Diff.	-1.8%	-1.8%
Impact Ionization	44.55	24.41
Diff.	0.7%	-1.5%
Constant $\tau_v = 1 \times 10^{-5}$	42.43	24.41
Diff.	-4.1%	-1.5%
Pref. Diss. Model	44.15	24.78
Diff.	-0.2%	0.0%
$T^{0.7}T_v^{0.3}$	44.41	24.82
Diff.	0.3%	0.1%
Laura's Chemistry Set	43.46	25.38
Diff.	-1.8%	2.4%
Laura's Grid	42.67	24.01
Diff.	-3.6%	-3.1%

reducing the shock standoff distance. The vibrational temperature is a little larger in the thermal equilibrium simulation, as can be observed in Fig. 5.45, but it is in good agreement with the baseline solution. The thermal equilibrium calculation does not cause significant differences in the number densities of electrons, nitrogen and oxygen along the stagnation line as shown in Figs. 5.46, 5.47 and 5.48. Because of such small differences in vibrational temperature and the number densities, the difference in the calculated radiation is small.

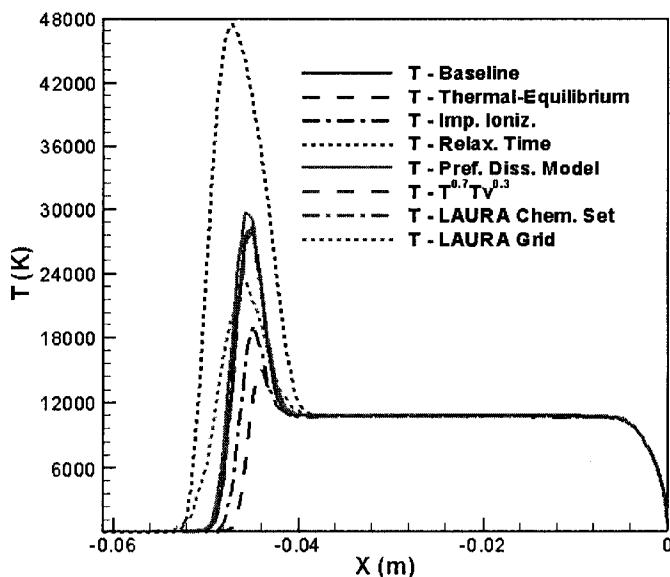


Figure 5.44: Translational temperature profiles along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case.

The second modification changes the model for impact ionization reactions. In LAURA those reaction rates are functions of the vibrational-electron-electronic temperature only while in DPLR they are function of the translational temperature only. This change reduces the translational temperature significantly as seen in Fig. 5.44, because it causes those endothermic reactions to occur close to the shock wave. As shown in Fig. 5.38, the smaller temperature also reduces the shock standoff distance,

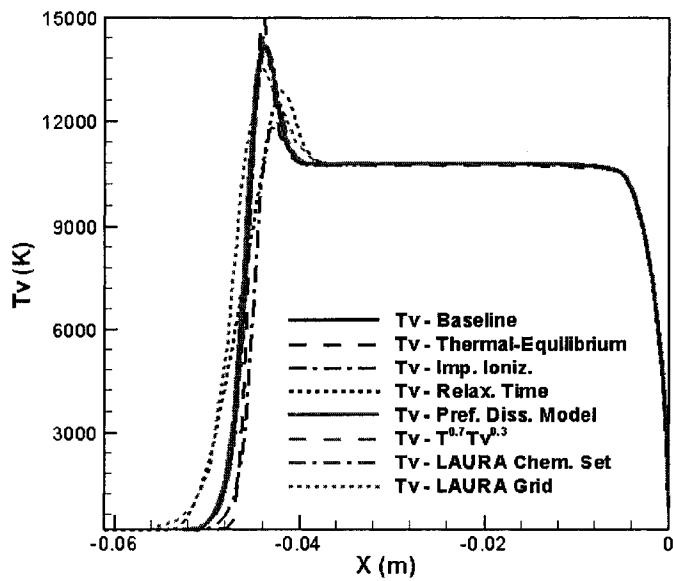


Figure 5.45: Vibrational temperature profiles along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case.

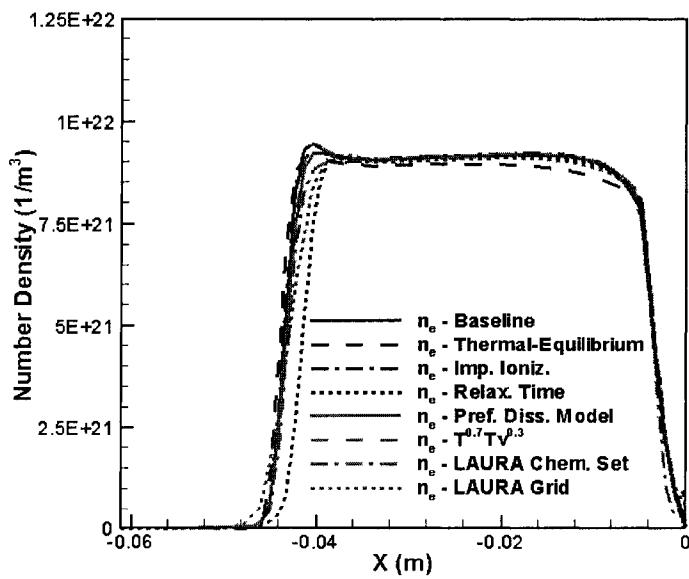


Figure 5.46: Electron number density along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case calculated using different thermal non-equilibrium models.

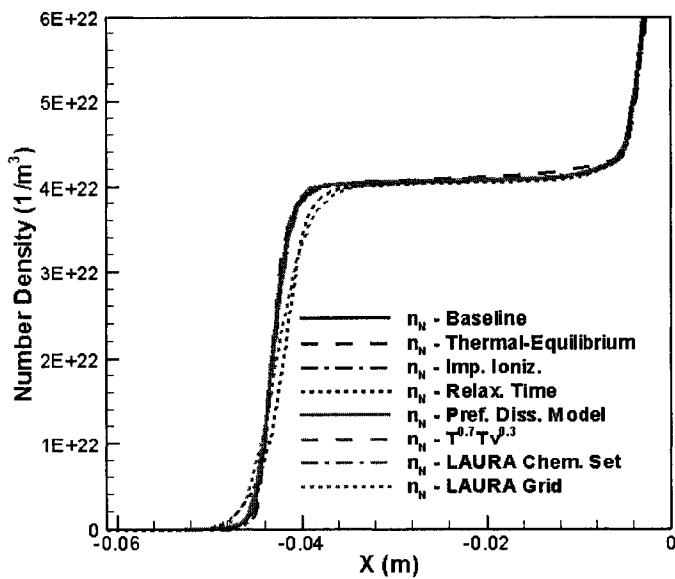


Figure 5.47: Atomic nitrogen number density along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case calculated using different thermal non-equilibrium models.

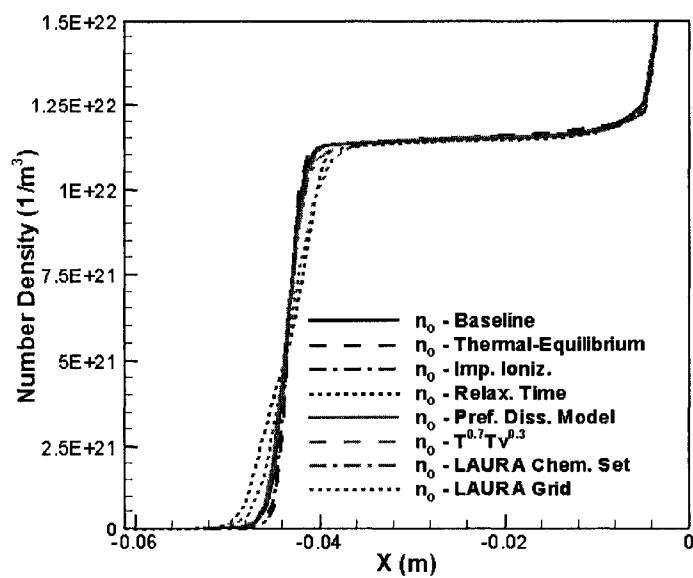


Figure 5.48: Atomic oxygen number density along the stagnation line for the 1636 seconds, 11-species, non-catalytic wall case calculated using different thermal non-equilibrium models.

yielding good agreement to the one calculated by DPLR. This modification also reduces the maximum vibrational temperature as observed in Fig. 5.45. Figures 5.46 to 5.48 show that the impact on number densities of electrons, nitrogen and oxygen is negligible. The change in radiation is negligible because the change in vibrational temperature occurs before the number densities of electrons, nitrogen and oxygen are significant enough to affect the radiation calculation.

The third modification increases the translational-vibrational energy transfer relaxation time. This study aims to contrast the thermal-equilibrium case by choosing a value that is larger than the one calculated using the standard Millikan and White correlation. The value chosen is $\tau_v = 1 \times 10^{-5}$ seconds. Figure 5.44 shows that this modification significantly increases the translational temperature which gets as high as 48,000 K. This occurs because the energy transfer rate between the translational and the other energy modes is slowed down. Due to the higher temperature, the density close to the shock is smaller, thus increasing the shock standoff distance. Figure 5.45 shows that the maximum vibrational temperature is smaller than the one in the baseline solution. Also, the maximum vibrational temperature occurs closer to the body. The steep increases at the shock in the number density profiles, shown in Figs. 5.46 to 5.48, are also closer to the body. The combined effect is a slight decrease in the calculated radiation.

The fourth modification uses a preferential dissociation model that accounts for the fact that molecules at higher energy vibrational states are more likely to dissociate. Figures 5.44 and 5.45 show that this modification does not cause significant changes in the translational or vibrational temperature profiles. Figures 5.46 to 5.48 show that there are no noticeable changes in the number density profiles either. Not surprisingly, the calculated radiation is very close to the baseline.

The fifth modification uses different parameters for calculating the temperature that is used in the forward dissociation reaction rates (Park's temperature). DPLR and LeMANS use $T_p = T^{0.5} \times T_V^{0.5}$ while LAURA uses that $T_p = T^{0.7} \times T_V^{0.3}$. This modification causes almost no difference in the translational and vibrational temperature profiles as can be seen in Figs. 5.44 and 5.45, respectively. Figures 5.46 to 5.48 show that there are no changes in the number density profiles. This explains the agreement between the radiation calculated in this case and in the baseline solution.

The sixth modification uses the LAURA chemistry set as reported in Ref. [37] and also in Appendix A. This modification does not cause significant changes in the translational nor in the vibrational temperature profiles, as can be observed in Figs. 5.44 and 5.45. However, the different chemistry set does cause a change in the electron number density profile. Figure 5.46 shows that this modification eliminates the peak in the electron number density that occurs close to the shock wave in the LeMANS baseline solution. The number density profiles for nitrogen and oxygen are mostly the same as in the baseline, as shown in Figs. 5.47 and 5.48. The reduction in electron number density causes a reduction in the radiation heat transfer when compared to the baseline.

The seventh and last modification uses the same grid as LAURA. This allows investigation of the diffusive character of LAURA's solution close to the shock wave. The coarser grid increases the amount of dissipation added at the shock wave, which broadens the translational temperature jump seen in Fig. 5.44. The broadening of the shock gives extra time for the energy transfer between the translational and the vibrational energy modes, thus causing a decrease in the maximum translational temperature. Figure 5.45 shows that the maximum vibrational temperature is a little smaller and that the steep increase in vibrational temperature occurs sooner because

of the thicker shock. Figures 5.46 to 5.48 show that the number density profiles are more diffused, in qualitative agreement with the LAURA profiles in Figs. 5.41 to 5.43. As a result, the calculated radiation shows better agreement with the LAURA calculation.

Overall, the tested modifications did not cause significant changes in radiative heating rates. The largest change happened due to grid refinement at the shock, which is independent of the thermo-chemical non-equilibrium models. It should be noted that the present sensitivity analysis is performed for only one flow condition and the observations are therefore limited.

CHAPTER VI

SUMMARY AND CONCLUSIONS

6.1 Summary

This work presents the mathematical and numerical formulation employed in the development of a multi-dimensional Computational Fluid Dynamics (CFD) code for the simulation of weakly ionized hypersonic flows in thermo-chemical non-equilibrium over entry configurations. The flow is modeled using the Navier-Stokes equations for continuum flow modified to include thermo-chemical non-equilibrium and weak ionization. The chemical non-equilibrium is modeled by including species conservation equations in the formulation and by using finite-rate chemistry to model the chemistry source terms. The thermal non-equilibrium is modeled by including a separate energy equation that tracks the vibrational-electron-electronic energy pool and by using relaxation rates to model the energy transfer rate between the translational-rotational mode and the vibrational-electron-electronic mode. For a three-dimensional simulation using a 11-species air model the partial differential equation system has 16 equations. The diffusion of species is modeled by an approximation to the solution of the Stefan-Maxwell equations and the transport properties of the gas mixture are modeled using collision cross section data and mixture rules based on gas kinetic theory.

This set of equations is solved numerically using modern CFD techniques. The flowfield is discretized using unstructured grids made of any mixture of quadrilaterals and triangles in two-dimensions or hexahedra, tetrahedra, prisms and pyramids in three-dimensions. The partial differential equations are integrated on such grids using the finite volume approach. The fluxes across grid faces are calculated using a modified form of the Steger-Warming Flux Vector Splitting scheme that has low numerical dissipation inside boundary layers but enough numerical dissipation in shock waves to ensure stability. Higher order extension of the inviscid fluxes is achieved by using standard ideas developed for structured grids applied on a local stencil constructed from the unstructured data structure. While this approach has been used before for unstructured grids made of hexahedra, this work extends its application to general unstructured grids. An important characteristic of the procedure proposed is that no interpolated values are used making the procedure very robust for hypersonic flows. Moreover, the stencil used in the reconstruction is fixed so that its cost is negligible in relation to the total cost of the calculation.

Steady state solutions are obtained by integrating the solution over time using the Euler implicit method and linearization of the inviscid and viscous fluxes and source terms. The resulting sparse linear system can be solved by using a point implicit method in which only the diagonal is retained in the implicit operator or by a line implicit method in which a tridiagonal matrix is assembled by using lines of cells that are formed starting at the wall. The use of two different numbering systems simplifies the originally sparse system of equations to the tridiagonal system. In both methods, the off-diagonal terms not included in the implicit operator are relaxed. An algorithm that assembles the lines used in the line implicit method using completely general unstructured grids is developed in this work.

The numerical code, called LeMANS (“Le” Michigan Aerothermodynamics Navier-Stokes Solver) is written using the C language to facilitate the development of a hybrid DSMC-CFD code at the University of Michigan. LeMANS is also parallelized by using METIS for grid partitioning and MPI for data transfer between processors. The code is designed to read FLUENT files so that most commercial grid generators can be used. LeMANS also parses chemistry file databases so that the inclusion of different species and reactions can be easily performed.

The code is a significant contribution to the aerothermodynamics field. While most of the physical and numerical modeling used is not new, the code itself is totally developed outside other established CFD groups that perform numerical simulations of hypersonic flows for entry applications. The code is going to be used by other researchers to test the limitations of current physical and numerical modeling and possibly in the development of new models. In fact, the code is already being used as part of the already mentioned hybrid DSMC-CFD code for the simulation of flows where the continuum approximation breaks down in localized regions, in the comparison of numerical results between DSMC and CFD, in the modeling of the flowfield typical of plasma guns and in benchmark studies against other CFD codes.

In this work LeMANS is employed in the simulation of several entry flows over space capsules as part of its validation process. Important quantities for the aerothermodynamics design of capsules such as aerodynamic coefficients and heat transfer rates are compared to available experimental and flight test data and other numerical results. A sensitivity analysis of radiative heating to several thermo-chemical non-equilibrium models is also performed.

6.2 Conclusions

LeMANS successfully calculated aerodynamic coefficients of an Apollo entry capsule model tested in a wind-tunnel at hypersonic speeds. The freestream conditions are at such low temperature that no chemical reactions are observed and vibrational energy excitation can be neglected. Meshes made of hexahedras are employed in these simulations with a special treatment of the stagnation region to avoid numerical problems. Most of the forces acting on the capsule are due to the pressure difference between the forebody and the after-body and accurate prediction of flow separation on the afterbody is necessary. The numerical solutions obtained are in very good agreement with experimental results obtained for the axial and normal force coefficients and also for the pitching moment coefficient.

The code is successfully used to simulate the heat transfer over a Mars entry capsule model tested in the HYPULSE wind tunnel for air and carbon dioxide hypersonic flows. It is demonstrated that second order discretization is necessary to obtain accurate values of heat transfer close to the symmetry axis in axisymmetric simulations. The numerical solutions are within experimental uncertainty for both fluids along the model forebody. The results for the afterbody air flow capture the general shape of the experimental distribution but underpredict the heat transfer rate along the sting in relation to the experimental data. Nevertheless, the results for the afterbody are in very good agreement to other numerical solutions. Numerical simulation of the RAM-C II flight data has also shown that electron number densities computed with LeMANS agree very well with other researchers' calculations that used similar physical models, and to flight data.

The numerical results obtained in these two test cases indicate that the flow-

fields around hypersonic vehicles are highly reactive and in thermo-chemical non-equilibrium under the conditions considered. The chemical reactions lower the temperature of the flow inside the shock layer by endothermic reactions. Vibrational excitation also lowers the temperature of the shock layer. Both processes occur at finite rates leading to thermo-chemical non-equilibrium in the flow field. The chemical non-equilibrium is observed in plots of chemical composition in which the equilibrium plateau is reached only some distance after the shock wave. The thermal equilibrium is observed in plots of temperature along the stagnation line characterized by significant differences between the translational and vibrational temperatures after the shock. These two processes are fundamental to the correct determination of the shock standoff distance and properties inside the shock layer which ultimately define the heat transfer to the wall and electron number density in the flow. The chemical reaction rates are dependent on the level of non-equilibrium of the flow and this is modeled using an approximate two-temperature model that seems to be a reasonable approximation given the good results obtained for heat transfer and electron number density. These two simulations also show that simplified models for diffusion and transport properties developed for relatively low speed are indeed adequate for application to hypersonic flow.

LeMANS is successfully used to simulate the convective heat transfer to the FIRE-II spacecraft at different trajectory points. The results are in remarkable agreement with two other established CFD codes routinely used in reentry calculations: DPLR (developed at NASA Ames Research Center) and LAURA (developed at NASA Langley Research Center). For the 1636 seconds case, LeMANS is very close to DPLR due to many similarities in the physical and numerical modeling. While DPLR couples the electron and electronic temperature to the translational and rotational

temperature, LeMANS couples them to the vibrational temperature. The remarkable agreement between the two codes indicates that the coupling choice does not have much effect on convective heating rates for these flow conditions. Overall, the differences in convective heating between LeMANS, DPLR and LAURA stay under 7% for 11-species air model calculations. The numerical results also indicate that the high speed flow (11 km/s) causes a much higher degree of ionization which must be accounted for in the physical modeling by using adequate models for air, diffusion and transport properties.

LeMANS flow field results are also employed to perform stagnation point radiative heating calculations using NASA's NEQAIR code. The results are in good agreement with the NEQAIR results obtained using DPLR and LAURA flow field solutions. A sensitivity study restricted to modifications in the thermochemical non-equilibrium model is also performed. It is found that reasonable changes in some of the parameters of this model do not lead to very significant changes in the radiation heat transfer. For radiative heating, the choice of electron-electronic coupling is visible in the modeling of electron impact ionization reactions. The modeling of this reaction rate depends on the electron temperature and it is shown that the coupling choice affects the shock standoff distance. Nevertheless, the impact on radiative heating is still small. One of the largest source of discrepancies in the radiative heating calculations is the computational grid. The results obtained indicate that CFD grids used in radiation analysis need to be refined at the shock wave in addition to at the boundary layer.

6.3 Future Research

The development of a parallel, unstructured, multi-dimensional CFD code with many physical models is a very time consuming task. During the development of this code, many improvements and not essential models were left behind so that the most fundamental physical modeling would be implemented. In this section, ideas for the implementation of improved algorithms and additional physics modeling are discussed.

The inviscid, viscous, chemistry and non-equilibrium Jacobians used in LeMANS are calculated by several matrix multiplications as shown in Appendix B. This approach yields a code that is simple to understand by just looking at the matrices and comparing them with the generated code. Running time can be saved by performing these multiplications analytically using a symbolic mathematical program and implementing the results. This approach can potentially save time when the matrices are large as in the simulation of flows in thermo-chemical non-equilibrium.

The multi-dimensionality in LeMANS is handled by using loops over the number of dimensions of the problem which is determined from the grid. These loops are small (at most a length of three for three-dimensional flows) but they can make difficult the vectorization of long loops that contain them. The use of compiler directives that force the compiler to unroll those loops can be tested to evaluate possible improvements in performance. The same approach can be employed for the species in the chemistry model. LeMANS loops over the number of species in many operations. These loops are also small (a length of eleven for a 11-species air simulation) and the unrolling of these loops may improve performance as well.

The linear algebra routines in LeMANS can be improved by using pre-compiled

libraries available in most scientific computing clusters. Several levels of optimization are available and could be tested. The block matrix-vector and matrix-matrix multiplications used in the solution of the linear system of equations can be performed using routines from the BLAS[13, 2] library which has routines optimized according to processor memory hierarchy. Another option is ATLAS[92, 1] which provides routines from BLAS that are empirically optimized during its compilation in the system. A higher level of optimization can be attained by using the LAPACK[7, 3] library which has several methods for solving the entire linear system of equations. It should be noted that LAPACK uses BLAS for fast block matrices operations. Some of the methods implemented in LAPACK require pre-conditioning of the linear system which could be done using a line solver similar to the one used in this work. An even higher level of optimization could be tested by using the PETSc[8, 4] libraries which offer an entire framework for the solution of partial differential equations on parallel computers using implicit methods. BLAS and LAPACK could be implemented in a fairly straightforward way in LeMANS. Using PETSc seems to require more extensive recoding but possible gains, as reported by the PETSc-FUN3D project which attained record number of float point operations for an unstructured CFD code[33], may be worth the extra work.

All the results presented in this dissertation use the modified Steger-Warming flux vector splitting for modeling the inviscid fluxes. There are several other options for their modeling such as Van Leer's flux vector splitting[50], Roe's flux difference splitting[79, 78], the AUSM family[56], the HLL family[36], the Rusanov scheme[80] and rotated schemes[54]. The implementation of these flux functions for flows in thermo-chemical non-equilibrium is a valuable addition to the numerical code. It should be noted that each inviscid model would require a different Jacobian to be

used in solving the linear system. The influence of using the same Jacobian derived for the modified Steger-Warming flux vector splitting for all the methods in relation to the exact Jacobian of each method would be also an interesting investigation that has been done before for transonic and supersonic flows[53]. Another possibility is to perform the calculation of the Jacobians by using numerical differentiation inside the code.

The addition of artificial dissipation is standard practice in the simulation of hypersonic flows but its influence on calculated properties, such as an increase in computed heat transfer rates, is not explicitly shown in the literature. Comparing the impact of adding artificial dissipation with the impact of changing physical models could bring more attention to the development of new inviscid fluxes. A fundamental problem to be solved in the numerical simulation of hypersonic flows is how to improve results using truly unstructured grids of triangles and quadrilaterals. It is well known in the aerothermodynamics community that meshes of triangles and tetrahedra do not yield accurate heat transfer rates in hypersonic flow simulations. Most flux functions are not able to obtain reasonable solutions when using these grids and a new flux function with an automatic way of adding artificial dissipation may be required. An assessment of available inviscid flux functions for the simulation of hypersonic flow using unstructured grids could be the first step in such a project.

While a definitive flux function for unstructured grids is not developed, users will probably have to employ grids made of hexahedra and quadrilaterals for simulations. Those grids can minimize oscillations at the shock when aligned to the strong shock wave. The creation of a grid aligned to the shock wave requires a lot of experience and at least one preliminary solution to determine the shock location. A mesh adaptation routine can be extremely useful to reduce the time spent generating

grids. The solution could be started on a quickly constructed grid and the mesh adaptation would improve the mesh in terms of shock alignment and mesh quality.

In LeMANS, the output of files and the creation of restart files are also parallelized. This means that for a simulation running on N processors LeMANS creates N output files and N restart files. It would be interesting to modify the output phase of LeMANS such that just one output and one restart files are created. That would allow the restart of simulations using a different number of processors than before. Currently LeMANS has to restart using the same number of processors that was used to create the restart files.

One of the physical processes left out of LeMANS that is important is wall catalysis. LeMANS has the option to run non-catalytic and super-catalytic wall boundary conditions. In the super-catalytic case it is assumed that the flow recombines to the composition of the freestream which is not physically possible. In reality, the recombination at the wall is limited by the diffusion of species towards the wall. In that case, a fraction of the diffused species recombines according to specific wall reactions. If that fraction is set to 100%, the wall is called fully-catalytic. This model is not implemented and it is a necessary addition to the code. Turbulence modeling is left out of the current implementation as well. Entry flows are in the laminar regime only for the low density encountered at high altitudes. Turbulence modeling will be important at lower altitudes. In addition, the injection of material from an ablating surface may cause transition to a turbulent boundary layer.

The present code is developed to simulate flows in thermo-chemical non-equilibrium. The code can be used to simulate flows in thermo-chemical equilibrium by setting the relaxation time to be very small (on the order to 10^{-13} seconds) and by multiplying the reactions rates by a large factor (such as 10^3) to emulate equilibrium chemistry.

While this approach gives accurate results, it is not computationally efficient. Some specific techniques used in the numerical simulation of flows in equilibrium, such as reduced order kinetics, could be employed in the code for more efficient simulation of these flows.

Finally, the code should be placed in a version control system so that the contributions of all researchers are available. By using this approach, each incremental improvement in the code would be immediately available to all the users. Such a system would require someone responsible for the code that would perform tests on the new versions to make sure that some benchmark problems are always computed correctly.

APPENDICES

APPENDIX A

Chemistry Sets

A.1 Species data

Table A.1 has the basic data for each species in the 11-species air model, including molecular weight, enthalpy of formation assuming 0K as the base temperature, constants A, B and C for the Blottner viscosity model, the characteristic temperature of vibration and the dissociation energy of molecules.

Table A.1: Species Chemistry Data

Species	M (kg/m ³)	h_f° (J/kg)	A	B	C	θ_v (K)	D (J/kg)	\hat{I} (J/kg)
N ₂	28	0	2.68E-02	3.18E-01	-1.13E+01	3395	3.36E+07	2.89E+07
O ₂	32	0	4.49E-02	-8.26E-02	-9.20E+00	2239	1.54E+07	2.69E+07
NO	30	3.00E+06	4.36E-02	-3.36E-02	-9.58E+00	2817	2.09E+07	0
N	14	3.36E+07	1.16E-02	6.03E-01	-1.24E+01	0	0	0
O	16	1.54E+07	2.03E-02	4.29E-01	-1.16E+01	0	0	0
NO ⁺	29.9994514	3.28E+07	3.02E-01	-3.5039791	-3.74E+00	2817	3.49E+07	0
N ₂ ⁺	27.9994514	5.43E+07	2.68E-02	3.18E-01	-1.13E+01	3395	3.00E+07	0
O ₂ ⁺	31.9994514	3.66E+07	4.49E-02	-8.26E-02	-9.20E+00	2239	2.01E+07	0
N ⁺	13.9994514	1.34E+08	1.16E-02	6.03E-01	-1.24E+01	0	0	0
O ⁺	15.9994514	9.77E+07	2.03E-02	4.29E-01	-1.16E+01	0	0	0
e	0.0005486	0	0	0	-1.20E+01	0	0	0

A.2 Electronic energy data

Tables A.2 to A.4 present the electronic characteristic temperature and degeneracies of the electronic energy levels of the 11-species air model.

Table A.2: Electronic Energy Levels.

Species	Level	θ_{el} (K)	g
O	0	0.000000000000000E+00	5
O	1	2.277077570280000E+02	3
O	2	3.265688785704000E+02	1
O	3	2.283028632262240E+04	5
O	4	4.861993036434160E+04	1
O ⁺	0	0.000000000000000E+00	4
O ⁺	1	3.858334678336000E+04	10
O ⁺	2	5.822349152848000E+04	6
O ₂	0	0.000000000000000E+00	3
O ₂	1	1.139156019700800E+04	2
O ₂	2	1.898473947826400E+04	1
O ₂	3	4.755973576639200E+04	1
O ₂	4	4.991242097343200E+04	6
O ₂	5	5.092268575561600E+04	3
O ₂	6	7.189863255967200E+04	3
O ₂ ⁺	0	0.000000000000000E+00	4
O ₂ ⁺	1	4.735440815760000E+04	8
O ₂ ⁺	2	5.837398741440000E+04	4
O ₂ ⁺	3	5.841427312000000E+04	6
O ₂ ⁺	4	6.229896616000000E+04	4
O ₂ ⁺	5	6.733467936000000E+04	2
O ₂ ⁺	6	7.121937240000000E+04	4
O ₂ ⁺	7	7.654284064000000E+04	4
O ₂ ⁺	8	8.819691976000000E+04	4
O ₂ ⁺	9	8.891630736000000E+04	4
O ₂ ⁺	10	9.423977560000000E+04	8
O ₂ ⁺	11	9.495916320000000E+04	4
O ₂ ⁺	12	9.592026503360000E+04	2
O ₂ ⁺	13	9.985099880000000E+04	2
O ₂ ⁺	14	1.035918144000000E+05	4

Table A.3: Electronic Energy Levels - continued.

Species	Level	θ_{el} (K)	g
N ₂ ⁺	0	0.00000000000000E+00	2
N ₂ ⁺	1	1.31899716460000E+04	4
N ₂ ⁺	2	3.663323087728000E+04	2
N ₂ ⁺	3	3.66887676000000E+04	4
N ₂ ⁺	4	5.98530483200000E+04	8
N ₂ ⁺	5	6.61836592000000E+04	8
N ₂ ⁺	6	7.598991933064000E+04	4
N ₂ ⁺	7	7.62550856000000E+04	4
N ₂ ⁺	8	8.20101864000000E+04	4
N ₂ ⁺	9	8.41683492000000E+04	4
N ₂ ⁺	10	8.63265120000000E+04	8
N ₂ ⁺	11	8.92040624000000E+04	8
N ₂ ⁺	12	9.20816128000000E+04	4
N ₂ ⁺	13	9.22254903200000E+04	4
N ₂ ⁺	14	9.29376840440000E+04	2
N ₂ ⁺	15	9.63979384000000E+04	2
N ₂ ⁺	16	1.03591814400000E+05	4
NO	0	0.00000000000000E+00	4
NO	1	5.46734576000000E+04	8
NO	2	6.317139627802400E+04	2
NO	3	6.599450342445600E+04	4
NO	4	6.90612096000000E+04	4
NO	5	7.04999848000000E+04	4
NO	6	7.491055017560000E+04	4
NO	7	7.628875293968000E+04	2
NO	8	8.676188537552000E+04	4
NO	9	8.714431182368000E+04	2
NO	10	8.886077063728000E+04	4
NO	11	8.981755614528000E+04	4
NO	12	8.988445919208000E+04	2
NO	13	9.042702132000000E+04	2
NO	14	9.06428376000000E+04	2
NO	15	9.111763341600000E+04	4
NO ⁺	0	0.00000000000000E+00	1
NO ⁺	1	7.508967768800000E+04	3
NO ⁺	2	8.525462447600000E+04	6
NO ⁺	3	8.903572570160000E+04	6
NO ⁺	4	9.746982592400000E+04	3
NO ⁺	5	1.000553049584000E+05	1
NO ⁺	6	1.028033655904000E+05	2
NO ⁺	7	1.057138639424800E+05	2
e	0	0.000000000000000E+00	1

Table A.4: Electronic energy levels - continued.

Species	Level	θ_{el} (K)	g
N	0	0.000000000000000E+00	4
N	1	2.766469645581980E+04	10
N	2	4.149309313560210E+04	6
N^+	0	0.000000000000000E+00	1
N^+	1	7.006835224000000E+01	3
N^+	2	1.881917961600000E+02	5
N^+	3	2.203656871824000E+04	5
N^+	4	4.703183475776000E+04	1
N^+	5	6.731252222192000E+04	5
N^+	6	1.327190797527310E+05	15
N_2	1	0.000000000000000E+00	1
N_2	2	7.223156514095200E+04	3
N_2	3	8.577862640384000E+04	6
N_2	4	8.605026716160000E+04	6
N_2	5	9.535118627874400E+04	3
N_2	6	9.805635702203200E+04	1
N_2	7	9.968267656935200E+04	2
N_2	8	1.048976467715200E+05	2
N_2	9	1.116489555200000E+05	5
N_2	10	1.225836470400000E+05	1
N_2	11	1.248856873600000E+05	6
N_2	12	1.282476158188320E+05	6
N_2	13	1.338060936000000E+05	10
N_2	14	1.404296391107200E+05	6
N_2	15	1.504958859200000E+05	6

A.3 Species thermodynamic curve fits

Table A.5 presents the thermodynamic curve fit parameters for the 11-species air model. It includes the formation enthalpy assuming 298.15 K as base temperature, the valid temperature range for each set of coefficients and the coefficients.

Table A.5: Lewis Research Center Curve fits.

Species	h_f^o (J/kg)	T_1 (K)	T^u (K)	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
e	0.000E+00	2.000E+02	1.000E+03	0.000E+00	0.000E+00	2.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	-7.454E+02	-1.172E+01
e	0.000E+00	1.000E+03	6.000E+03	0.000E+00	0.000E+00	2.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	-7.454E+02	-1.172E+01
e	0.000E+00	6.000E+03	2.000E+04	0.000E+00	0.000E+00	2.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	-7.454E+02	-1.172E+01
N	4.727E+05	2.000E+02	1.000E+03	0.000E+00	0.000E+00	2.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	5.610E+04	4.194E+00
N	4.727E+05	1.000E+03	6.000E+03	8.877E+04	-1.971E+02	2.362E+00	2.917E+04	-1.730E+07	4.013E-11	-2.677E-15	0.000E+00	5.697E+04	4.865E+00
N	4.727E+05	2.000E+04	6.000E+03	6.000E+03	-3.108E+05	6.917E+01	-6.848E-03	3.828E-07	-1.098E-11	1.278E-15	0.000E+00	-5.849E+02	-5.849E+02
N ⁺	1.882E+06	2.000E+02	1.000E+03	5.237E+03	2.300E+00	2.487E+00	2.737E-05	-3.134E-08	1.850E-11	-4.447E-15	0.000E+00	2.256E+05	5.077E+00
N ⁺	1.882E+06	1.000E+03	6.000E+03	2.905E+05	-8.558E+02	3.477E+00	-5.288E-04	1.352E-07	-1.390E-11	5.046E-16	0.000E+00	2.311E+05	-1.994E+00
N ⁺	1.882E+06	6.000E+03	2.000E+04	1.646E+07	-1.113E+04	4.977E+00	-2.005E-04	1.022E-08	-2.691E-13	3.540E-18	0.000E+00	3.136E+05	-1.707E+01
N ₂	0.000E+00	2.000E+02	1.000E+03	2.210E+04	-3.818E+02	6.083E+00	-8.531E-03	1.385E-05	-9.626E-09	2.520E-12	0.000E+00	7.108E+02	-1.076E+01
N ₂	0.000E+00	1.000E+00	6.000E+03	5.877E+05	-2.239E+03	6.067E+00	-6.140E-04	1.492E-07	-1.923E-11	1.062E-15	0.000E+00	1.283E+04	-1.587E+01
N ₂	0.000E+00	6.000E+03	2.000E+04	8.310E+08	-6.420E+05	2.020E+02	-3.065E-02	2.487E-06	-9.706E-11	1.438E-15	0.000E+00	4.939E+06	-1.672E+03
N ₂ ⁺	1.510E+06	2.000E+02	1.000E+03	-3.474E+04	2.698E+02	3.165E+00	-2.132E-03	6.730E-06	-5.637E-09	1.622E-12	0.000E+00	1.790E+05	6.833E+00
N ₂ ⁺	1.510E+06	1.000E+03	6.000E+03	-2.846E+06	7.0595E+03	-2.885E+00	3.069E-03	-4.362E-07	2.103E-11	5.412E-16	0.000E+00	1.340E+05	5.091E+01
N ₂ ⁺	1.510E+06	6.000E+03	2.000E+04	-3.713E+08	3.1349E+05	-9.604E+01	1.571E-02	-1.175E-06	4.144E-11	-5.622E-16	0.000E+00	-2.217E+06	8.436E+02
NO	9.127E+04	2.000E+02	1.000E+03	-1.144E+04	1.5365E+02	3.431E+00	-2.668E-03	8.481E-06	-7.685E-09	2.387E-12	0.000E+00	9.098E+03	6.729E+00
NO	9.127E+04	2.000E+03	6.000E+03	-1.239E+05	-1.290E+03	5.434E+00	-1.656E-04	9.881E-08	-1.416E-11	9.380E-16	0.000E+00	-8.502E+00	-8.502E+00
NO	9.127E+04	6.000E+03	2.000E+04	-9.575E+08	5.9120E+05	-1.385E+02	1.694E-02	-1.007E-06	2.913E-11	-3.295E-16	0.000E+00	-4.675E+06	1.242E+03
NO ⁺	9.908E+05	2.000E+02	1.000E+03	1.389E+03	-1.590E+02	5.123E+00	-6.3394E-03	1.124E-05	-7.989E-09	2.107E-12	0.000E+00	1.187E+05	-4.399E+00
NO ⁺	9.908E+05	1.000E+03	6.000E+03	6.070E+05	-2.2778E+03	6.080E+00	-6.067E-04	1.432E-07	-1.748E-11	8.935E-16	0.000E+00	1.323E+05	-1.520E+01
NO ⁺	9.908E+05	6.000E+03	2.000E+04	2.676E+09	-1.8338E+06	5.909E+02	-7.114E-07	5.318E-06	-7.1963E-10	2.805E-15	0.000E+00	-4.324E+03	-4.324E+03
O	2.492E+05	2.000E+02	1.000E+03	-7.954E+03	1.607E+02	1.966E+00	1.014E-03	-1.110E-06	6.518E-10	-1.585E-13	0.000E+00	2.840E+04	8.404E+00
O	2.492E+05	1.000E+03	6.000E+03	2.619E+05	-7.299E+02	3.317E+00	-4.281E-04	1.036E-07	-9.458E-12	2.725E-16	0.000E+00	3.392E+04	-6.680E-01
O	2.492E+05	6.000E+03	2.000E+04	1.779E+08	-1.082E+06	2.811E+01	-2.975E-03	1.855E-07	-5.796E-12	7.192E-17	0.000E+00	8.891E+05	-2.182E+02
O ⁺	1.569E+06	2.000E+02	1.000E+03	0.000E+00	0.000E+00	2.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	1.879E+05	4.393E+00
O ⁺	1.569E+06	1.000E+03	6.000E+03	-2.167E+05	6.665E+02	1.702E+00	4.715E-04	-1.427E-07	2.017E-11	-9.107E-16	0.000E+00	1.837E+05	1.006E+01
O ⁺	1.569E+06	6.000E+03	2.000E+04	-2.144E+08	1.4705E+05	-3.661E+01	5.036E-03	-3.088E-07	9.187E-12	-1.074E-16	0.000E+00	-9.614E+05	3.426E+02
O ₂	0.000E+00	2.000E+02	1.000E+03	1.562E+09	-1.161E+06	4.711E-02	3.303E+02	4.711E-15	1.590E-15	1.168E-10	0.000E+00	-8.555E+06	2.855E+03
O ₂	0.000E+00	1.000E+03	6.000E+03	-1.038E+06	2.347E+02	1.119E+00	4.294E-03	-6.836E-07	-2.023E-09	1.039E-12	0.000E+00	-3.391E+03	1.850E+01
O ₂	0.000E+00	6.000E+03	2.000E+04	4.975E+08	-2.866E+05	1.820E+00	1.268E-03	-2.188E-07	3.054E-11	-8.193E-16	0.000E+00	-1.683E+04	1.739E+01
O ₂ ⁺	1.172E+06	2.000E+02	1.000E+03	-8.177E+04	1.005E+03	-3.365E-01	6.105E-03	-2.704E-06	3.012E-10	4.163E-13	0.000E+00	1.348E+05	2.783E+01
O ₂ ⁺	1.172E+06	1.000E+03	6.000E+03	7.366E+04	-8.459E+02	4.985E+00	-1.613E-04	6.434E-08	-1.506E-11	1.579E-15	0.000E+00	-5.813E+05	-5.813E+00
O ₂ ⁺	1.172E+06	6.000E+03	2.000E+04	1.562E+09	1.161E+06	-3.303E+02	4.711E-02	3.355E-06	1.168E-10	1.590E-15	0.000E+00	-8.555E+06	2.855E+03

A.4 Collision cross section data

Tables A.6 to A.9 present the curve fit parameters of the neutral-neutral, neutral-ion and neutral-electrons collision integrals $\Omega_{0,0}$ and $\Omega_{1,1}$. Table A.10 presents the curve fit parameters for the ion-ion, electron-ion and electron-electron collision integrals. The collision integrals are used to calculate the viscosity, heat conduction and binary diffusion coefficients.

Table A.6: $\Omega_{0,0}$ integrals for collisions involving neutrals only.

Collision pair	A_0	A_1	A_2	A_3
$N_2 - N_2$	-6.0614558E-03	1.2689102E-01	-1.0616948E+00	8.0955466E+02
$N_2 - O_2$	-3.7959091E-03	9.5708295E-02	-1.0070611E+00	8.9392313E+02
$O_2 - O_2$	-8.0682650E-04	1.6602480E-02	-3.1472774E-01	1.4116458E+02
$N_2 - N$	-1.0796249E-02	2.2656509E-01	-1.7910602E+00	4.0455218E+03
$O_2 - N$	-1.1453028E-03	1.2654140E-02	-2.2435218E-01	7.7201588E+01
$N - N$	-9.6083779E-03	2.0938971E-01	-1.7386904E+00	3.3587983E+03
$N_2 - O$	-2.7244269E-03	6.9587171E-02	-7.9538667E-01	4.0673730E+02
$O_2 - O$	-4.8405803E-03	1.0297688E-01	-9.6876576E-01	6.1629812E+02
$N - O$	-7.8147689E-03	1.6792705E-01	-1.4308628E+00	1.6628859E+03
$O - O$	-6.4040535E-03	1.4629949E-01	-1.3892121E+00	2.0903441E+03
$N_2 - NO$	-1.9295666E-03	2.7995735E-02	-3.1588514E-01	1.2880734E+02
$O_2 - NO$	-6.4433840E-04	8.5378580E-03	-2.3225102E-01	1.1371608E+02
$NO - N$	-1.5770918E-03	1.9578381E-02	-2.7873624E-01	9.9547944E+01
$NO - O$	-1.0885815E-03	1.1883688E-02	-2.1844909E-01	7.5512560E+01
$NO - NO$	0.0000000E+00	-1.1056066E-02	-5.9216250E-02	7.2542367E+01
$N_2 - NO^+$	0.0000000E+00	9.1205839E-02	-1.8728231E+00	2.4432020E+05
$O_2 - NO^+$	-3.7822765E-03	1.7967016E-01	-2.5409098E+00	1.1840435E+06
$N - NO^+$	-1.9605234E-02	5.5570872E-01	-5.4285702E+00	1.3574446E+09
$O - NO^+$	-1.6409054E-02	4.6352852E-01	-4.5479735E+00	7.4250671E+07
$NO - NO^+$	-8.1158474E-03	2.1474280E-01	-2.0148450E+00	6.2986385E+04
$e - N_2$	-1.0525124E-02	1.3498950E-01	1.2524805E-01	1.5066506E-01
$e - O_2$	2.3527001E-02	-6.9632323E-01	6.8035475E+00	1.8335509E-09
$e - N$	0.0000000E+00	1.6554247E-01	-3.4986344E+00	5.9268038E+08
$e - O$	9.9865506E-03	-2.7407431E-01	2.6561032E+00	4.3080676E-04
$e - NO$	1.0414818E-01	-2.8369126E+00	2.5323135E+01	7.7138358E-32

Table A.7: $\Omega_{0,0}$ integrals for collisions involving neutrals only - continued.

Collision pair	A ₀	A ₁	A ₂	A ₃
N ₂ - N ⁺	-1.0687805E-02	2.4479697E-01	-2.3192863E+00	1.0689229E+05
O ₂ - N ⁺	0.0000000E+00	8.7745537E-02	-1.8347158E+00	1.9830120E+05
N - N ⁺	-4.0078980E-03	1.0327487E-01	-9.9473323E-01	2.8178290E+03
O - N ⁺	-1.5767256E-02	3.5405830E-01	-3.0686783E+00	4.6336779E+05
NO - N ⁺	0.0000000E+00	8.6531037E-02	-1.8117931E+00	1.8621272E+05
N ₂ - O ⁺	1.0352091E-02	-1.5733723E-01	2.9326150E-02	2.1003616E+03
O ₂ - O ⁺	0.0000000E+00	9.3559978E-02	-1.9842999E+00	4.3097490E+05
N - O ⁺	-2.4288224E-02	5.6305072E-01	-4.6849679E+00	2.7303024E+07
O - O ⁺	-3.8347988E-03	9.9930498E-02	-9.6288891E-01	1.9669897E+03
NO - O ⁺	8.3856973E-03	-1.0972656E-01	-3.3896281E-01	5.2004690E+03
N ₂ - N ₂ ⁺	-2.9123716E-03	9.6850678E-02	-1.1416540E+00	7.9252169E+03
O ₂ - N ₂ ⁺	-4.0893007E-03	1.7795266E-01	-2.3800543E+00	5.1949298E+05
N - N ₂ ⁺	-1.4501284E-02	4.1085338E-01	-4.0115094E+00	1.5735451E+07
O - N ₂ ⁺	-1.6923472E-02	4.6067692E-01	-4.3294966E+00	2.5538927E+07
NO - N ₂ ⁺	-9.2292933E-03	2.9813226E-01	-3.2899475E+00	4.9147046E+06
N ₂ - O ₂ ⁺	1.2405624E-02	-2.0452111E-01	3.5478475E-01	1.0778357E+03
O ₂ - O ₂ ⁺	-8.9520932E-03	2.2749642E-01	-2.0758341E+00	6.7674419E+04
N - O ₂ ⁺	0.0000000E+00	8.3065769E-02	-1.7501512E+00	1.0846799E+05
O - O ₂ ⁺	-2.9417970E-03	1.5129273E-01	-2.2497964E+00	2.9325215E+05
NO - O ₂ ⁺	1.3731123E-02	-2.3920299E-01	6.7093226E-01	4.0068731E+02

Table A.8: $\Omega_{1,1}$ integrals for collisions involving neutrals only.

Collision pair	A ₀	A ₁	A ₂	A ₃
N ₂ - N ₂	-7.6303990E-03	1.6878089E-01	-1.4004234E+00	2.1427708E+03
N ₂ - O ₂	-8.0457321E-03	1.9228905E-01	-1.7102854E+00	5.2213857E+03
O ₂ - O ₂	-6.2931612E-03	1.4624645E-01	-1.3006927E+00	1.8066892E+03
N ₂ - N	-8.3493693E-03	1.7808911E-01	-1.4466155E+00	1.9324210E+03
O ₂ - N	-1.0608832E-03	1.1782595E-02	-2.1246301E-01	8.4561598E+01
N - N	-7.7439615E-03	1.7129007E-01	-1.4809088E+00	2.1284951E+03
N ₂ - O	-8.3110691E-03	1.9617877E-01	-1.7205427E+00	4.0812829E+03
O ₂ - O	-3.7969686E-03	7.6789981E-02	-7.3056809E-01	3.3958171E+02
N - O	-5.0478143E-03	1.0236186E-01	-9.0058935E-01	4.4472565E+02
O - O	-4.2451096E-03	9.6820337E-02	-9.9770795E-01	8.3320644E+02
N ₂ - NO	-6.8237776E-03	1.4360616E-01	-1.1922240E+00	1.2433086E+03
O ₂ - NO	-6.8508672E-03	1.5524564E-01	-1.3479583E+00	2.0037890E+03
NO - N	-1.4719259E-03	1.8446968E-02	-2.6460411E-01	1.0911124E+02
NO - O	-1.0066279E-03	1.1029264E-02	-2.0671266E-01	8.2644384E+01
NO - NO	-7.4942466E-03	1.6626193E-01	-1.4107027E+00	2.3097604E+03
N ₂ - NO ⁺	0.0000000E+00	8.5112236E-02	-1.7460044E+00	1.4498969E+05
O ₂ - NO ⁺	0.0000000E+00	8.4737359E-02	-1.7290488E+00	1.2485194E+05
N - NO ⁺	-2.1009546E-02	5.8910426E-01	-5.6681361E+00	2.4486594E+09
O - NO ⁺	-1.5315132E-02	4.3541627E-01	-4.2864279E+00	3.5125207E+07
NO - NO ⁺	1.1055777E-02	-1.6621846E-01	1.4372166E-01	1.3182061E+03
e - N ₂	-4.2254948E-03	-5.2965163E-02	1.9157708E+00	6.3263309E-04
e - O ₂	9.6744867E-03	-3.3759583E-01	3.7952121E+00	6.8468036E-06
e - N	-1.0903638E-01	2.8678381E+00	-2.5297550E+01	3.4838798E+33
e - O	-1.7924100E-02	4.0402656E-01	-2.6712374E+00	4.1447669E+02
e - NO	0.0000000E+00	5.4444485E-02	-1.2854128E+00	1.3857556E+04

Table A.9: $\Omega_{1,1}$ integrals for collisions involving neutrals only - continued.

Collision pair	A_0	A_1	A_2	A_3
$N_2 - N^+$	-7.0776069E-03	1.7917938E-01	-1.9102410E+00	4.6736263E+04
$O_2 - N^+$	9.8019578E-03	-1.4699425E-01	3.9382460E-02	1.5112165E+03
$N - N^+$	-1.4271306E-02	3.0401993E-01	-2.4573879E+00	5.3694705E+04
$O - N^+$	-1.7907392E-02	4.1207892E-01	-3.5343610E+00	1.4987678E+06
$NO - N^+$	1.1716366E-02	-1.9289789E-01	4.0269474E-01	6.0891590E+02
$N_2 - O^+$	1.8733000E-02	-3.6163781E-01	1.6947101E+00	2.5244859E+01
$O_2 - O^+$	1.4207970E-02	-2.4736726E-01	7.4561859E-01	3.2519188E+02
$N - O^+$	-2.1681211E-02	5.2300453E-01	-4.5118623E+00	2.3467766E+07
$O - O^+$	-1.6032919E-02	3.7114396E-01	-3.2050078E+00	5.8099314E+05
$NO - O^+$	1.8015337E-02	-3.4415293E-01	1.5658151E+00	3.3303758E+01
$N_2 - N_2^+$	-1.6447237E-02	4.7759522E-01	-4.7641986E+00	2.9127542E+08
$O_2 - N_2^+$	-4.9176811E-03	1.9694738E-01	-2.5025540E+00	6.8213629E+05
$N - N_2^+$	-1.2882395E-02	3.7306469E-01	-3.7106760E+00	7.5444981E+06
$O - N_2^+$	-1.7420606E-02	4.7126950E-01	-4.3841087E+00	2.8275095E+07
$NO - N_2^+$	-4.0133981E-03	1.7290664E-01	-2.2855449E+00	3.6429320E+05
$N_2 - O_2^+$	2.2455421E-02	-4.5106797E-01	2.3763420E+00	4.7754696E+00
$O_2 - O_2^+$	2.8664463E-02	-5.8087240E-01	3.2564558E+00	6.6890428E-01
$N - O_2^+$	1.1205000E-02	-1.8182149E-01	3.2624972E-01	5.5186183E+02
$O - O_2^+$	0.0000000E+00	8.3446262E-02	-1.7191179E+00	8.0539928E+04
$NO - O_2^+$	2.2679271E-02	-4.5710920E-01	2.4427275E+00	3.6733514E+00

Table A.10: Curve fits for shielded coulomb potential.

Collision integral	Attractive potential			Repulsive potential		
	C_n	c_n	D_n	C_n	c_n	D_n
$\Omega^{1,1}$	-0.476	0.0313	0.784	0.138	0.0106	0.765
$\Omega^{2,2}$	-0.146	0.0377	1.262	0.157	0.0274	1.235

A.5 Reactions data

Tables A.11 and A.12 present the reactions and respective forward reaction rate coefficients used in the numerical codes DPLR and LAURA.

Table A.11: Reaction data used in the DPLR code.

Reaction	$A \text{ (cm}^3/\text{mols})$	α	E (K)
Electron Impact Dissociation			
$\text{N}_2 + e \rightleftharpoons \text{N} + \text{N} + e$	3.0E+24	-1.60	113200
Dissociation			
$\text{N}_2 + M \rightleftharpoons \text{N} + \text{N} + M \text{ (M=N}_2, \text{O}_2, \text{NO)}$	7.0E+21	-1.60	113200
$\text{N}_2 + M \rightleftharpoons \text{N} + \text{N} + M \text{ (M=N, O)}$	3.0E+22	-1.60	113200
$\text{N}_2 + M \rightleftharpoons \text{N} + \text{N} + M \text{ (M=N}_2^+, \text{O}_2^+, \text{NO}^+)$	7.0E+21	-1.60	113200
$\text{N}_2 + M \rightleftharpoons \text{N} + \text{N} + M \text{ (M=N}^+, \text{O}^+)$	3.0E+22	-1.60	113200
$\text{O}_2 + M \rightleftharpoons \text{O} + \text{O} + M \text{ (M=N}_2, \text{O}_2, \text{NO)}$	2.0E+21	-1.50	59500
$\text{O}_2 + M \rightleftharpoons \text{O} + \text{O} + M \text{ (M=N, O)}$	1.0E+22	-1.50	59500
$\text{O}_2 + M \rightleftharpoons \text{O} + \text{O} + M \text{ (M=N}_2^+, \text{O}_2^+, \text{NO}^+)$	2.0E+21	-1.50	59500
$\text{O}_2 + M \rightleftharpoons \text{O} + \text{O} + M \text{ (M=N}^+, \text{O}^+)$	1.0E+22	-1.50	59500
$\text{NO} + M \rightleftharpoons \text{N} + \text{O} + M \text{ (M=N}_2, \text{O}_2, \text{NO)}$	5.0E+15	0.00	75500
$\text{NO} + M \rightleftharpoons \text{N} + \text{O} + M \text{ (M=N, O)}$	1.1E+17	0.00	75500
$\text{NO} + M \rightleftharpoons \text{N} + \text{O} + M \text{ (M=N}_2^+, \text{O}_2^+, \text{NO}^+)$	5.0E+15	0.00	75500
$\text{NO} + M \rightleftharpoons \text{N} + \text{O} + M \text{ (M=N}^+, \text{O}^+)$	1.1E+17	0.00	75500
Electron Impact Ionization			
$\text{N} + e \rightleftharpoons \text{N}^+ + e + e$	2.5E+34	-3.82	168600
$\text{O} + e \rightleftharpoons \text{O}^+ + e + e$	3.9E+33	-3.78	158500
Exchange			
$\text{N}_2 + \text{O} \rightleftharpoons \text{NO} + \text{N}$	6.4E+17	-1.00	38400
$\text{NO} + \text{O} \rightleftharpoons \text{O}_2 + \text{N}$	8.4E+12	0.00	19450
Dissociative Recombination			
$\text{N} + \text{O} \rightleftharpoons \text{NO}^+ + e$	5.3E+12	0.00	31900
$\text{N} + \text{N} \rightleftharpoons \text{N}_2^+ + e$	2.0E+13	0.00	67500
$\text{O} + \text{O} \rightleftharpoons \text{O}_2^+ + e$	1.1E+13	0.00	80600
Charge Exchange			
$\text{O}^+ + \text{N}_2 \rightleftharpoons \text{N}_2^+ + \text{O}$	9.1E+11	0.36	22800
$\text{O}^+ + \text{NO} \rightleftharpoons \text{N}^+ + \text{O}_2$	1.4E+05	1.90	15300
$\text{NO}^+ + \text{O}_2 \rightleftharpoons \text{O}_2^+ + \text{NO}$	2.4E+13	0.41	32600
$\text{NO}^+ + \text{N} \rightleftharpoons \text{N}_2^+ + \text{O}$	7.2E+13	0.00	35500
$\text{NO}^+ + \text{O} \rightleftharpoons \text{N}^+ + \text{O}_2$	1.0E+12	0.50	77200
$\text{O}_2^+ + \text{N} \rightleftharpoons \text{N}^+ + \text{O}_2$	8.7E+13	0.14	28600
$\text{O}_2^+ + \text{N}_2 \rightleftharpoons \text{N}_2^+ + \text{O}_2$	9.9E+12	0.00	40700
$\text{NO}^+ + \text{N} \rightleftharpoons \text{O}^+ + \text{N}_2$	3.4E+13	-1.08	12800
$\text{NO}^+ + \text{O} \rightleftharpoons \text{O}_2^+ + \text{N}$	7.2E+12	0.29	48600
$\text{O}_2^+ + \text{O} \rightleftharpoons \text{O}^+ + \text{O}_2$	-	-	-
$\text{N}^+ + \text{N}_2 \rightleftharpoons \text{N}_2^+ + \text{N}$	-	-	-

Table A.12: Reaction data used in the LAURA code.

Reaction	A ($\text{cm}^3/(\text{mols})$)	α	E (K)
Electron Impact Dissociation			
$\text{N}_2 + \text{e} \rightleftharpoons \text{N} + \text{N} + \text{e}$	3.0E+24	-1.60	113200
Dissociation			
$\text{N}_2 + \text{M} \rightleftharpoons \text{N} + \text{N} + \text{M}$ (M=N ₂ ,O ₂ ,NO)	7.0E+21	-1.60	113200
$\text{N}_2 + \text{M} \rightleftharpoons \text{N} + \text{N} + \text{M}$ (M=N,O)	3.0E+22	-1.60	113200
$\text{N}_2 + \text{M} \rightleftharpoons \text{N} + \text{N} + \text{M}$ (M=N ₂ ⁺ ,O ₂ ⁺ ,NO ⁺)	-	-	-
$\text{N}_2 + \text{M} \rightleftharpoons \text{N} + \text{N} + \text{M}$ (M=N ⁺ ,O ⁺)	-	-	-
$\text{O}_2 + \text{M} \rightleftharpoons \text{O} + \text{O} + \text{M}$ (M=N ₂ ,O ₂ ,NO)	2.0E+21	-1.50	59500
$\text{O}_2 + \text{M} \rightleftharpoons \text{O} + \text{O} + \text{M}$ (M=N,O)	1.0E+22	-1.50	59360
$\text{O}_2 + \text{M} \rightleftharpoons \text{O} + \text{O} + \text{M}$ (M=N ₂ ⁺ ,O ₂ ⁺ ,NO ⁺)	2.0E+21	-1.50	59360
$\text{O}_2 + \text{M} \rightleftharpoons \text{O} + \text{O} + \text{M}$ (M=N ⁺ ,O ⁺)	-	-	-
$\text{NO} + \text{M} \rightleftharpoons \text{N} + \text{O} + \text{M}$ (M=N ₂ ,O ₂ ,NO)	5.0E+15	0.00	75500
$\text{NO} + \text{M} \rightleftharpoons \text{N} + \text{O} + \text{M}$ (M=N,O)	1.1E+17	0.00	75500
$\text{NO} + \text{M} \rightleftharpoons \text{N} + \text{O} + \text{M}$ (M=N ₂ ⁺ ,O ₂ ⁺ ,NO ⁺)	-	-	-
$\text{NO} + \text{M} \rightleftharpoons \text{N} + \text{O} + \text{M}$ (M=N ⁺ ,O ⁺)	-	-	-
Electron Impact Ionization			
$\text{N} + \text{e} \rightleftharpoons \text{N}^+ + \text{e} + \text{e}$	2.5E+34	-3.82	168600
$\text{O} + \text{e} \rightleftharpoons \text{O}^+ + \text{e} + \text{e}$	3.9E+33	-3.78	158500
Exchange			
$\text{N}_2 + \text{O} \rightleftharpoons \text{NO} + \text{N}$	5.7E+12	0.42	42938
$\text{NO} + \text{O} \rightleftharpoons \text{O}_2 + \text{N}$	8.4E+12	0.00	19400
Dissociative Recombination			
$\text{N} + \text{O} \rightleftharpoons \text{NO}^+ + \text{e}$	5.3E+12	0.00	31900
$\text{N} + \text{N} \rightleftharpoons \text{N}_2^+ + \text{e}$	4.4E+07	1.50	67500
$\text{O} + \text{O} \rightleftharpoons \text{O}_2^+ + \text{e}$	7.1E+02	2.70	80600
Charge Exchange			
$\text{O}^+ + \text{N}_2 \rightleftharpoons \text{N}_2^+ + \text{O}$	9.1E+11	0.36	22800
$\text{O}^+ + \text{NO} \rightleftharpoons \text{N}^+ + \text{O}_2$	1.4E+05	1.90	26600
$\text{NO}^+ + \text{O}_2 \rightleftharpoons \text{O}_2^+ + \text{NO}$	2.4E+13	0.41	32600
$\text{NO}^+ + \text{N} \rightleftharpoons \text{N}_2^+ + \text{O}$	7.2E+13	0.00	35500
$\text{NO}^+ + \text{O} \rightleftharpoons \text{N}^+ + \text{O}_2$	1.0E+12	0.50	77200
$\text{O}_2^+ + \text{N} \rightleftharpoons \text{N}^+ + \text{O}_2$	8.7E+13	0.14	28600
$\text{O}_2^+ + \text{N}_2 \rightleftharpoons \text{N}_2^+ + \text{O}_2$	9.9E+12	0.00	40700
$\text{NO}^+ + \text{N} \rightleftharpoons \text{O}^+ + \text{N}_2$	3.4E+13	-1.08	12800
$\text{NO}^+ + \text{O} \rightleftharpoons \text{O}_2^+ + \text{N}$	7.2E+12	0.29	48600
$\text{O}_2^+ + \text{O} \rightleftharpoons \text{O}^+ + \text{O}_2$	4.0E+12	-0.09	18000
$\text{N}^+ + \text{N}_2 \rightleftharpoons \text{N}_2^+ + \text{N}$	1.0E+12	0.50	12200

A.6 Equilibrium constant data

Tables A.13 to A.15 present the equilibrium constant curve fit coefficients for the reactions considered in Park's 1990 11-species air model. The coefficients are a function of mixture number density.

Table A.13: Equilibrium Constants Curve Fits for Park's 90 model.

Reaction	n (1/cm ³)	A ₀	A ₁	A ₂	A ₃	A ₄
$N_2 + M \rightleftharpoons 2N + M$	1.00E+14	3.4907	0.83133	4.0978	-12.728	0.07487
	1.00E+15	2.0723	1.3897	2.0617	-11.828	0.015105
	1.00E+16	1.606	1.5732	1.3923	-11.533	-0.004543
	1.00E+17	1.5351	1.6061	1.2993	-11.494	-0.00698
	1.00E+18	1.4766	1.6291	1.2153	-11.457	-0.009444
	1.00E+19	1.4766	1.6291	1.2153	-11.457	-0.009444
$O_2 + M \rightleftharpoons 2O + M$	1.00E+14	1.8103	1.9607	3.5716	-7.3623	0.083861
	1.00E+15	0.91354	2.316	2.2885	-6.7969	0.046338
	1.00E+16	0.64183	2.4253	1.9026	-6.6277	0.035151
	1.00E+17	0.55388	2.46	1.7763	-6.572	0.031445
	1.00E+18	0.52455	2.4715	1.7342	-6.5534	0.030209
	1.00E+19	0.50989	2.4773	1.7132	-6.5441	0.029591
$NO + M \rightleftharpoons N + O + M$	1.00E+14	2.1649	0.078577	2.8508	-8.5422	0.053043
	1.00E+15	1.0072	0.53545	1.1911	-7.8098	0.004394
	1.00E+16	0.63817	0.68189	0.66336	-7.5773	-0.011025
	1.00E+17	0.55889	0.71558	0.55396	-7.5304	-0.014089
	1.00E+18	0.515	0.73286	0.49096	-7.5025	-0.015938
	1.00E+19	0.50765	0.73575	0.48042	-7.4979	-0.016247
$N_2 + O \rightleftharpoons NO + N$	1.00E+14	1.3261	0.75268	1.2474	-4.1857	0.02184
	1.00E+15	1.0653	0.85417	0.87093	-4.0188	0.010721
	1.00E+16	0.96794	0.89131	0.7291	-3.9555	0.006488
	1.00E+17	0.97646	0.89043	0.74572	-3.9642	0.007123
	1.00E+18	0.96188	0.89617	0.72479	-3.955	0.006509
	1.00E+19	0.96921	0.89329	0.73531	-3.9596	0.006818
$NO + O \rightleftharpoons O_2 + N$	1.00E+14	0.35438	-1.8821	-0.72111	-1.1797	-0.030831
	1.00E+15	0.093613	-1.7806	-1.0975	-1.0128	-0.041949
	1.00E+16	-0.003732	-1.7434	-1.2394	-0.94952	-0.046182
	1.00E+17	0.004815	-1.7443	-1.2227	-0.95824	-0.045545
	1.00E+18	-0.009758	-1.7386	-1.2436	-0.949	-0.046159
	1.00E+19	-0.002428	-1.7415	-1.2331	-0.95365	-0.04585
$N + O \rightleftharpoons NO^+ + e$	1.00E+14	-2.1852	-6.6709	-4.2968	-2.2175	-0.050748
	1.00E+15	-1.0276	-7.1278	-2.637	-2.95	-0.0021
	1.00E+16	-0.65871	-7.2742	-2.1096	-3.1823	0.01331
	1.00E+17	-0.57924	-7.3079	-1.9999	-3.2294	0.016382
	1.00E+18	-0.53538	-7.3252	-1.937	-3.2572	0.01823
	1.00E+19	-0.52801	-7.3281	-1.9264	-3.2618	0.01854
$N + N \rightleftharpoons N_2^+ + e$	1.00E+14	-4.3785	-4.2726	-7.8709	-4.4628	-0.12402
	1.00E+15	-2.9601	-4.831	-5.8348	-5.3621	-0.064252
	1.00E+16	-2.4938	-5.0145	-5.1654	-5.6577	-0.044602
	1.00E+17	-2.4229	-5.0474	-5.0724	-5.6961	-0.042167
	1.00E+18	-2.3644	-5.0704	-4.9885	-5.7332	-0.039703
	1.00E+19	-2.3644	-5.0704	-4.9885	-5.7332	-0.039703

Table A.14: Equilibrium Constants Curve Fits for Park's 90 model - continued.

Reaction	n (1/cm ³)	A ₀	A ₁	A ₂	A ₃	A ₄
$O + O \rightleftharpoons O_2^+ + e$	1.00E+14	-0.11682	-7.6883	-2.2498	-7.7905	-0.011079
	1.00E+15	0.77986	-8.0436	-0.96678	-8.3559	0.02644
	1.00E+16	1.0516	-8.153	-0.58082	-8.5251	0.037629
	1.00E+17	1.1395	-8.1876	-0.45461	-8.5808	0.041333
	1.00E+18	1.1689	-8.1991	-0.41245	-8.5995	0.042571
	1.00E+19	1.1835	-8.2049	-0.39146	-8.6087	0.043187
$N_2 + O_2^+ \rightleftharpoons N_2^+ + O_2$	1.00E+14	-2.5811	2.2863	-5.0946	-2.0378	-0.12192
	1.00E+15	-2.5811	2.2863	-5.0946	-2.0378	-0.12192
	1.00E+16	-2.5811	2.2863	-5.0946	-2.0378	-0.12192
	1.00E+17	-2.5811	2.2863	-5.0946	-2.0378	-0.12192
	1.00E+18	-2.5811	2.2863	-5.0946	-2.0378	-0.12192
	1.00E+19	-2.5811	2.2863	-5.0946	-2.0378	-0.12192
$NO^+ + N \rightleftharpoons O^+ + N_2$	1.00E+14	-1.2255	0.10039	-1.2212	-0.89883	-0.025232
	1.00E+15	-0.51629	-0.17877	-0.20321	-1.3485	0.004649
	1.00E+16	-0.28311	-0.27056	0.13152	-1.4963	0.014474
	1.00E+17	-0.24765	-0.28699	0.17802	-1.5155	0.015692
	1.00E+18	-0.21842	-0.29849	0.21998	-1.534	0.016923
	1.00E+19	-0.21842	-0.29849	0.21998	-1.534	0.016923
$NO^+ + O \rightleftharpoons N^+ + O_2$	1.00E+14	-1.5349	1.6836	-2.969	-6.464	-0.083316
	1.00E+15	-1.0864	1.5059	-2.3273	-6.7468	-0.064551
	1.00E+16	-0.95072	1.4513	-2.1346	-6.8313	-0.058964
	1.00E+17	-0.90672	1.434	-2.0714	-6.8592	-0.05711
	1.00E+18	-0.89206	1.4282	-2.0504	-6.8685	-0.056493
	1.00E+19	-0.88472	1.4254	-2.0398	-6.8731	-0.056184
$NO^+ + O_2 \rightleftharpoons O_2^+ + NO$	1.00E+14	1.7139	0.86469	2.7679	-4.3932	0.070493
	1.00E+15	1.7139	0.86469	2.7679	-4.3932	0.070493
	1.00E+16	1.7139	0.86469	2.7679	-4.3932	0.070493
	1.00E+17	1.7139	0.86469	2.7679	-4.3932	0.070493
	1.00E+18	1.7139	0.86469	2.7679	-4.3932	0.070493
	1.00E+19	1.7139	0.86469	2.7679	-4.3932	0.070493
$NO^+ + N \rightleftharpoons N_2^+ + O$	1.00E+14	-2.1934	2.3983	-3.5743	-2.2452	-0.073271
	1.00E+15	-1.9325	2.2968	-3.1978	"-2,412,100"	-0.062149
	1.00E+16	-1.8352	2.2597	-3.056	-2.4754	-0.057919
	1.00E+17	-1.8438	2.2606	-3.0726	-2.4667	-0.058554
	1.00E+18	-1.8292	2.2548	-3.0517	-2.4759	-0.05794
	1.00E+19	-1.8365	2.2577	-3.0622	-2.4713	-0.058248
$O_2^+ + N \rightleftharpoons N^+ + O_2$	1.00E+14	-3.603	2.701	-5.0155	-0.89125	-0.12297
	1.00E+15	-2.8938	2.4218	-3.9975	-1.3409	-0.093088
	1.00E+16	-2.6607	2.33	-3.6628	-1.4887	-0.083264
	1.00E+17	-2.6252	2.3136	-3.6163	-1.5079	-0.082048
	1.00E+18	-2.596	2.3021	-3.5744	-1.5264	-0.080816
	1.00E+19	-2.596	2.3021	-3.5744	-1.5264	-0.080816

Table A.15: Equilibrium Constants Curve Fits for Park's 90 model - continued.

Reaction	n (1/cm ³)	A ₀	A ₁	A ₂	A ₃	A ₄
$O^+ + NO \rightleftharpoons N^+ + O_2$	1.00E+14	-1.6355	0.83058	-2.9952	-1.3794	-0.079927
	1.00E+15	-1.6355	0.83058	-2.9952	-1.3794	-0.079927
	1.00E+16	-1.6355	0.83058	-2.9952	-1.3794	-0.079927
	1.00E+17	-1.6355	0.83058	-2.9952	-1.3794	-0.079927
	1.00E+18	-1.6355	0.83058	-2.9952	-1.3794	-0.079927
	1.00E+19	-1.6355	0.83058	-2.9952	-1.3794	-0.079927
$NO^+ + O \rightleftharpoons O_2^+ + N$	1.00E+14	2.0681	-1.0173	2.0466	-5.5728	0.039655
	1.00E+15	1.8073	-0.91584	1.6701	-5.4058	0.028533
	1.00E+16	1.71	-0.87869	1.5282	-5.3426	0.024301
	1.00E+17	1.7185	-0.87958	1.5449	-5.3513	0.024936
	1.00E+18	1.7039	-0.87383	1.5239	-5.342	0.024321
	1.00E+19	1.7112	-0.87672	1.5345	-5.3467	0.024631
$O^+ + N_2 \rightleftharpoons N_2^+ + O$	1.00E+14	-0.96795	2.2979	-2.3531	-1.3463	-0.048042
	1.00E+15	-1.4164	2.4756	-2.9947	-1.0636	-0.066805
	1.00E+16	-1.5522	2.5303	-3.1876	-0.97903	-0.072396
	1.00E+17	-1.5962	2.5476	-3.2507	-0.95116	-0.074249
	1.00E+18	-1.6108	2.5533	-3.2718	-0.94186	-0.074867
	1.00E+19	-1.6181	2.5562	-3.2823	-0.93721	-0.075176
$N_2 + e \rightleftharpoons 2N + e$	1.00E+14	3.4907	0.83133	4.0978	-12.728	0.07487
	1.00E+15	2.0723	1.3897	2.0617	-11.828	0.015105
	1.00E+16	1.606	1.5732	1.3923	-11.533	-0.004543
	1.00E+17	1.5351	1.6061	1.2993	-11.494	-0.00698
	1.00E+18	1.4766	1.6291	1.2153	-11.457	-0.009444
	1.00E+19	1.4766	1.6291	1.2153	-11.457	-0.009444
$N + e \rightleftharpoons N^+ + e + e$	1.00E+14	-1.9094	-3.0267	-3.6935	-16.044	-0.050183
	1.00E+15	-1.2002	-3.3059	-2.6755	-16.494	-0.020301
	1.00E+16	-0.96709	-3.3976	-2.3408	-16.642	-0.010477
	1.00E+17	-0.93184	-3.414	-2.2946	-16.661	-0.009269
	1.00E+18	-0.9026	-3.4255	-2.2526	-16.679	-0.008037
	1.00E+19	-0.9026	-3.4255	-2.2526	-16.679	-0.008037
$O + e \rightleftharpoons O^+ + e + e$	1.00E+14	0.08045	-5.7393	-1.4195	-15.844	-0.001087
	1.00E+15	0.52883	-5.917	-0.77795	-16.127	0.017675
	1.00E+16	0.66478	-5.9716	-0.58486	-16.212	0.023273
	1.00E+17	0.70879	-5.989	-0.52169	-16.24	0.025127
	1.00E+18	0.72341	-5.9947	-0.5007	-16.249	0.025743
	1.00E+19	0.73078	-5.9976	-0.49012	-16.254	0.026054

A.7 Modified Millikan and White coefficients

Tables A.16 and A.17 present the modified coefficients a and b used in the Millikan and White's curve fits used to determine vibrational relaxation times.

Table A.16: Modified Millikan and White's Parameters.

Vibrator	Partner	a	b
N ₂	N ₂	221.53	0.029
N ₂	O ₂	228.76	0.0295
N ₂	N	180.88	0.0262
N ₂	O	72.4	0.015
N ₂	NO	225.3	0.0293
N ₂	N ₂ ⁺	221.53	0.029
N ₂	O ₂ ⁺	228.76	0.0295
N ₂	N ⁺	180.88	0.0262
N ₂	O ⁺	188.89	0.0268
N ₂	NO ⁺	225.3	0.0293
N ₂	e	1.39	0.0023
O ₂	N ₂	131.32	0.0295
O ₂	O ₂	135.91	0.03
O ₂	N	72.4	0.015
O ₂	O	47.7	0.059
O ₂	NO	133.71	0.0298
O ₂	N ₂ ⁺	131.32	0.0295
O ₂	O ₂ ⁺	135.91	0.03
O ₂	N ⁺	106.06	0.0265
O ₂	O ⁺	110.97	0.0271
O ₂	NO ⁺	133.71	0.0298
O ₂	e	0.8	0.0023
NO	N ₂	49.5	0.042
NO	O ₂	49.5	0.042
NO	N	49.5	0.042
NO	O	49.5	0.042
NO	NO	49.5	0.042
NO	N ₂ ⁺	175.67	0.0293
NO	O ₂ ⁺	181.6	0.0298
NO	N ⁺	142.62	0.0264
NO	O ⁺	149.08	0.027
NO	NO ⁺	178.76	0.0295
NO	e	1.08	0.0023

Table A.17: Modified Millikan and White's Parameters - continued.

Vibrator	Partner	a	b
N ₂ ⁺	N ₂	221.53	0.029
N ₂ ⁺	O ₂	228.76	0.0295
N ₂ ⁺	N	180.88	0.0262
N ₂ ⁺	O	188.89	0.0268
N ₂ ⁺	NO	225.3	0.0293
N ₂ ⁺	N ₂ ⁺	221.53	0.029
N ₂ ⁺	O ₂ ⁺	228.75	0.0295
N ₂ ⁺	N ⁺	180.88	0.0262
N ₂ ⁺	O ⁺	188.88	0.0268
N ₂ ⁺	NO ⁺	225.3	0.0293
N ₂ ⁺	e	1.39	0.0023
O ₂ ⁺	N ₂	131.32	0.0295
O ₂ ⁺	O ₂	135.91	0.03
O ₂ ⁺	N	106.06	0.0265
O ₂ ⁺	O	110.97	0.0271
O ₂ ⁺	NO	133.71	0.0298
O ₂ ⁺	N ₂ ⁺	131.32	0.0295
O ₂ ⁺	O ₂ ⁺	135.9	0.03
O ₂ ⁺	N ⁺	106.05	0.0265
O ₂ ⁺	O ⁺	110.97	0.0271
O ₂ ⁺	NO ⁺	133.7	0.0298
O ₂ ⁺	e	0.8	0.0023
NO ⁺	N ₂	175.67	0.0293
NO ⁺	O ₂	181.61	0.0298
NO ⁺	N	142.62	0.0264
NO ⁺	O	149.09	0.027
NO ⁺	NO	178.76	0.0295
NO ⁺	N ₂ ⁺	175.67	0.0293
NO ⁺	O ₂ ⁺	181.6	0.0298
NO ⁺	N ⁺	142.62	0.0264
NO ⁺	O ⁺	149.08	0.027
NO ⁺	NO ⁺	178.76	0.0295
NO ⁺	e	1.08	0.0023

APPENDIX B

Jacobian Matrices

B.1 Inviscid Jacobians

The homogeneous property of the inviscid flux vector yields that

$$\vec{F} \cdot \vec{n} = F_n = \frac{dF_n}{dQ} Q = A Q \quad . \quad (\text{B.1})$$

The matrix A is called the Jacobian of the inviscid flux. It can be diagonalized by the matrices of its eigenvectors L and R as[24]

$$A = L \Lambda R \quad (\text{B.2})$$

where

$$R = \begin{bmatrix} a^2 \delta_{sr} - c_s \tilde{\gamma}_{sr} & \beta u c_s & \beta v c_s & \beta w c_s & -\beta c_s & -\phi c_s \\ -v' & l_x & l_y & l_z & 0 & 0 \\ -w' & m_x & m_y & m_z & 0 & 0 \\ \tilde{\gamma}_r - u'a & an_x - \beta u & an_y - \beta v & an_z - \beta w & \beta & \phi \\ \tilde{\gamma}_r + u'a & -an_x - \beta u & -an_y - \beta v & -an_z - \beta w & \beta & \phi \\ -e_V \tilde{\gamma}_r & \beta u e_V & \beta v e_V & \beta w e_V & -\beta e_V & a^2 - \phi e_V \end{bmatrix} \quad (\text{B.3})$$

and

$$L = \begin{bmatrix} \delta_{sr}/a^2 & 0 & 0 & c_s/(2a^2) & c_s/(2a^2) & 0 \\ u/a^2 & l_x & m_x & (u + an_x)/(2a^2) & (u - an_x)/(2a^2) & 0 \\ v/a^2 & l_y & m_y & (v + an_y)/(2a^2) & (v - an_y)/(2a^2) & 0 \\ w/a^2 & l_z & m_z & (w + an_z)/(2a^2) & (w - an_z)/(2a^2) & 0 \\ \eta & v' & w' & (H + aU)/(2a^2) & (H - aU)/(2a^2) & -\phi/(\beta a^2) \\ 0 & 0 & 0 & e_v/(2a^2) & e_v/(2a^2) & 1/a^2 \end{bmatrix} . \quad (\text{B.4})$$

In the previous matrices,

$$\eta = (\beta(u^2 + v^2 + w^2) - \tilde{\gamma}_r)/(\beta a^2) \quad , \quad (\text{B.5})$$

$$\beta = \frac{\partial p}{\partial E} = \frac{Ru}{\rho C v_t} \sum_{s \neq e} \frac{\rho_s}{M_s} \quad , \quad (\text{B.6})$$

$$\phi = \frac{\partial p}{\partial Ev} = \frac{Ru}{\rho C v_{v,el}} \frac{\rho_e}{M_e} - \beta \quad , \quad (\text{B.7})$$

$$\gamma_s = \frac{\partial p}{\partial \rho_s} = \frac{Ru T_s}{M_s} + \beta \frac{u^2 + v^2 + w^2}{2} - \beta e_s - \phi e_{V,s} \quad , \quad (\text{B.8})$$

and

$$a^2 = (1 + \beta) \frac{p}{\rho} \quad . \quad (\text{B.9})$$

It should be observed that when the flow is not ionized and $Y_e = 0$ then $\beta = \gamma - 1$.

The matrix of eigenvalues is given by

$$\Lambda = \begin{bmatrix} u' & 0 & 0 & 0 & 0 & 0 \\ 0 & u' & 0 & 0 & 0 & 0 \\ 0 & 0 & u' & 0 & 0 & 0 \\ 0 & 0 & 0 & u' + a & 0 & 0 \\ 0 & 0 & 0 & 0 & u' - a & 0 \\ 0 & 0 & 0 & 0 & 0 & u' \end{bmatrix} \quad (\text{B.10})$$

B.2 True Inviscid Jacobians

While the original Euler flux is a homogeneous function for which

$$\frac{\partial F}{\partial Q} = A \quad \text{and} \quad F = A Q \quad , \quad (\text{B.11})$$

the split fluxes are not. The split fluxes are such that

$$\frac{\partial F^\pm}{\partial Q} \neq A^\pm \quad \text{but} \quad F^\pm = A^\pm Q \quad . \quad (\text{B.12})$$

For increased stability of the numerical method, it is desired to use the actual matrices $\frac{\partial F^\pm}{\partial Q}$ in the implicit operator. The split fluxes, originally given by

$$F^\pm = A^\pm Q \quad (\text{B.13})$$

can also be written in a way that eliminates the matrix multiplication as[40]

$$F^\pm = \frac{\rho}{2\gamma} \left\{ \begin{array}{l} 2(\gamma - 1)\lambda_1^\pm Y_s + \lambda_2^\pm Y_s + \lambda_3^\pm Y_s \\ 2(\gamma - 1)\lambda_1^\pm u + \lambda_2^\pm(u + an_x) + \lambda_3^\pm(u - an_x) \\ 2(\gamma - 1)\lambda_1^\pm v + \lambda_2^\pm(v + an_y) + \lambda_3^\pm(v - an_y) \\ 2(\gamma - 1)\lambda_1^\pm w + \lambda_2^\pm(w + an_z) + \lambda_3^\pm(w - an_z) \\ 2(\gamma - 1)\lambda_1^\pm \epsilon_1 + \lambda_2^\pm \epsilon_2 + \lambda_3^\pm \epsilon_3 \\ 2(\gamma - 1)\lambda_1^\pm(e_{ve}) + \lambda_2^\pm(e_{ve}) + \lambda_3^\pm(e_{ve}) \end{array} \right\}, \quad (\text{B.14})$$

where

$$\begin{aligned} \epsilon_0 &= \frac{1}{2}(u^2 + v^2 + w^2) + e_{ve} + h^o \\ \epsilon_1 &= \frac{(3-\gamma)a^2}{2(\gamma-1)} + \frac{1}{2}((u + an_x)^2 + (v + an_y)^2 + (w + an_z)^2) + e_{ve} + h^o \\ \epsilon_2 &= \frac{(3-\gamma)a^2}{2(\gamma-1)} + \frac{1}{2}((u - an_x)^2 + (v - an_y)^2 + (w - an_z)^2) + e_{ve} + h^o \end{aligned} \quad (\text{B.15})$$

$$e_{ve} = \sum Y_s e_{ve,s}$$

$$h^o = \sum Y_s h_s^o$$

To obtain the Jacobian, it is easier to work using the set of variables

$$W = \left\{ \begin{array}{l} \rho_s \\ u \\ v \\ w \\ a \\ e_{ve} \end{array} \right\}. \quad (\text{B.16})$$

The relation between W and Q is given by

$$\frac{\partial W}{\partial Q} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 & 0 \\ -\frac{v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 & 0 \\ -\frac{w}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 & 0 \\ \frac{\partial a}{\partial \rho_s} & \frac{\partial a}{\partial \rho u} & \frac{\partial a}{\partial \rho v} & \frac{\partial a}{\partial \rho w} & \frac{\partial a}{\partial E} & \frac{\partial a}{\partial E_v} \\ -\frac{e_{ve}}{\rho} & 0 & 0 & 0 & 0 & \frac{1}{\rho} \end{bmatrix}, \quad (\text{B.17})$$

where

$$\begin{aligned} \frac{\partial a}{\partial \rho_s} &= \frac{\gamma(\gamma-1)}{2\rho a} \left[\frac{1}{2}(u^2 + v^2 + w^2) - \frac{a^2}{\gamma(\gamma-1)} - h_s^o \right] \\ \frac{\partial a}{\partial \rho u} &= -\frac{\gamma(\gamma-1)}{2\rho a} u \\ \frac{\partial a}{\partial \rho v} &= -\frac{\gamma(\gamma-1)}{2\rho a} v \\ \frac{\partial a}{\partial \rho w} &= -\frac{\gamma(\gamma-1)}{2\rho a} w \\ \frac{\partial a}{\partial E} &= \frac{\gamma(\gamma-1)}{2\rho a} \\ \frac{\partial a}{\partial E_v} &= -\frac{\gamma(\gamma-1)}{2\rho a} . \end{aligned} \quad (\text{B.18})$$

The Jacobian is given by

$$\frac{\partial F^\pm}{\partial W} = \begin{bmatrix} \frac{\partial F_\rho^\pm}{\partial \rho_s} & \frac{\partial F_\rho^\pm}{\partial u} & \frac{\partial F_\rho^\pm}{\partial v} & \frac{\partial F_\rho^\pm}{\partial w} & \frac{\partial F_\rho^\pm}{\partial a} & \frac{\partial F_\rho^\pm}{\partial e_{ve}} \\ \frac{\partial F_{\rho u}^\pm}{\partial \rho_s} & \frac{\partial F_{\rho u}^\pm}{\partial u} & \frac{\partial F_{\rho u}^\pm}{\partial v} & \frac{\partial F_{\rho u}^\pm}{\partial w} & \frac{\partial F_{\rho u}^\pm}{\partial a} & \frac{\partial F_{\rho u}^\pm}{\partial e_{ve}} \\ \frac{\partial F_{\rho v}^\pm}{\partial \rho_s} & \frac{\partial F_{\rho v}^\pm}{\partial u} & \frac{\partial F_{\rho v}^\pm}{\partial v} & \frac{\partial F_{\rho v}^\pm}{\partial w} & \frac{\partial F_{\rho v}^\pm}{\partial a} & \frac{\partial F_{\rho v}^\pm}{\partial e_{ve}} \\ \frac{\partial F_{\rho w}^\pm}{\partial \rho_s} & \frac{\partial F_{\rho w}^\pm}{\partial u} & \frac{\partial F_{\rho w}^\pm}{\partial v} & \frac{\partial F_{\rho w}^\pm}{\partial w} & \frac{\partial F_{\rho w}^\pm}{\partial a} & \frac{\partial F_{\rho w}^\pm}{\partial e_{ve}} \\ \frac{\partial F_E^\pm}{\partial \rho_s} & \frac{\partial F_E^\pm}{\partial u} & \frac{\partial F_E^\pm}{\partial v} & \frac{\partial F_E^\pm}{\partial w} & \frac{\partial F_E^\pm}{\partial a} & \frac{\partial F_E^\pm}{\partial e_{ve}} \\ \frac{\partial F_{Ev}^\pm}{\partial \rho_s} & \frac{\partial F_{Ev}^\pm}{\partial u} & \frac{\partial F_{Ev}^\pm}{\partial v} & \frac{\partial F_{Ev}^\pm}{\partial w} & \frac{\partial F_{Ev}^\pm}{\partial a} & \frac{\partial F_{Ev}^\pm}{\partial e_{ve}} \end{bmatrix}, \quad (\text{B.19})$$

where

$$\begin{aligned} \frac{\partial F_{\rho s}^{\pm}}{\partial \rho_r} &= \frac{1}{2\gamma} (2(\gamma - 1)\lambda_1^{\pm} + \lambda_2^{\pm} + \lambda_3^{\pm}) \\ \frac{\partial F_{\rho s}^{\pm}}{\partial u} &= \frac{\rho_s}{2\gamma} \left(2(\gamma - 1) \frac{\partial \lambda_1^{\pm}}{\partial u} + \frac{\partial \lambda_2^{\pm}}{\partial u} + \frac{\partial \lambda_3^{\pm}}{\partial u} \right) \\ \frac{\partial F_{\rho s}^{\pm}}{\partial v} &= \frac{\rho_s}{2\gamma} \left(2(\gamma - 1) \frac{\partial \lambda_1^{\pm}}{\partial v} + \frac{\partial \lambda_2^{\pm}}{\partial v} + \frac{\partial \lambda_3^{\pm}}{\partial v} \right) \\ \frac{\partial F_{\rho s}^{\pm}}{\partial w} &= \frac{\rho_s}{2\gamma} \left(2(\gamma - 1) \frac{\partial \lambda_1^{\pm}}{\partial w} + \frac{\partial \lambda_2^{\pm}}{\partial w} + \frac{\partial \lambda_3^{\pm}}{\partial w} \right) \\ \frac{\partial F_{\rho s}^{\pm}}{\partial a} &= \frac{\rho_s}{2\gamma} \left(2(\gamma - 1) \frac{\partial \lambda_1^{\pm}}{\partial a} + \frac{\partial \lambda_2^{\pm}}{\partial a} + \frac{\partial \lambda_3^{\pm}}{\partial a} \right) \\ \frac{\partial F_{\rho s}^{\pm}}{\partial e_{ve}} &= \frac{\rho_s}{2\gamma} \left(2(\gamma - 1) \frac{\partial \lambda_1^{\pm}}{\partial e_{ve}} + \frac{\partial \lambda_2^{\pm}}{\partial e_{ve}} + \frac{\partial \lambda_3^{\pm}}{\partial e_{ve}} \right) \end{aligned}, \quad (\text{B.20})$$

$$\begin{aligned} \frac{\partial F_{\rho u}^{\pm}}{\partial \rho_r} &= \frac{1}{2\gamma} (2(\gamma - 1)u\lambda_1^{\pm} + (u + an_x)\lambda_2^{\pm} + (u - an_x)\lambda_3^{\pm}) \\ \frac{\partial F_{\rho u}^{\pm}}{\partial u} &= \frac{\rho_s}{2\gamma} \left(2(\gamma - 1)u \frac{\partial \lambda_1^{\pm}}{\partial u} + (u + an_x) \frac{\partial \lambda_2^{\pm}}{\partial u} + (u - an_x) \frac{\partial \lambda_3^{\pm}}{\partial u} \right) + \\ &\quad \frac{\rho}{2\gamma} (2(\gamma - 1)\lambda_1^{\pm} + \lambda_2^{\pm} + \lambda_3^{\pm}) \\ \frac{\partial F_{\rho u}^{\pm}}{\partial v} &= \frac{\rho_s}{2\gamma} \left(2(\gamma - 1)u \frac{\partial \lambda_1^{\pm}}{\partial v} + (u + an_x) \frac{\partial \lambda_2^{\pm}}{\partial v} + (u - an_x) \frac{\partial \lambda_3^{\pm}}{\partial v} \right) \\ \frac{\partial F_{\rho u}^{\pm}}{\partial w} &= \frac{\rho_s}{2\gamma} \left(2(\gamma - 1)u \frac{\partial \lambda_1^{\pm}}{\partial w} + (u + an_x) \frac{\partial \lambda_2^{\pm}}{\partial w} + (u - an_x) \frac{\partial \lambda_3^{\pm}}{\partial w} \right) \\ \frac{\partial F_{\rho u}^{\pm}}{\partial a} &= \frac{\rho_s}{2\gamma} \left(2(\gamma - 1)u \frac{\partial \lambda_1^{\pm}}{\partial a} + (u + an_x) \frac{\partial \lambda_2^{\pm}}{\partial a} + (u - an_x) \frac{\partial \lambda_3^{\pm}}{\partial a} \right) + \\ &\quad \frac{\rho}{2\gamma} [\lambda_2^{\pm} n_x - \lambda_3^{\pm} n_x] \end{aligned}, \quad (\text{B.21})$$

$$\begin{aligned} \frac{\partial F_{\rho v}^{\pm}}{\partial \rho_r} &= \frac{1}{2\gamma} (2(\gamma - 1)v\lambda_1^{\pm} + (v + an_y)\lambda_2^{\pm} + (v - an_y)\lambda_3^{\pm}) \\ \frac{\partial F_{\rho v}^{\pm}}{\partial u} &= \frac{\rho_s}{2\gamma} \left(2(\gamma - 1)v \frac{\partial \lambda_1^{\pm}}{\partial u} + (v + an_y) \frac{\partial \lambda_2^{\pm}}{\partial u} + (v - an_y) \frac{\partial \lambda_3^{\pm}}{\partial u} \right) \\ \frac{\partial F_{\rho v}^{\pm}}{\partial v} &= \frac{\rho_s}{2\gamma} \left(2(\gamma - 1)v \frac{\partial \lambda_1^{\pm}}{\partial v} + (v + an_y) \frac{\partial \lambda_2^{\pm}}{\partial v} + (v - an_y) \frac{\partial \lambda_3^{\pm}}{\partial v} \right) + \\ &\quad \frac{\rho}{2\gamma} (2(\gamma - 1)\lambda_1^{\pm} + \lambda_2^{\pm} + \lambda_3^{\pm}) \\ \frac{\partial F_{\rho v}^{\pm}}{\partial w} &= \frac{\rho_s}{2\gamma} \left(2(\gamma - 1)v \frac{\partial \lambda_1^{\pm}}{\partial w} + (v + an_y) \frac{\partial \lambda_2^{\pm}}{\partial w} + (v - an_y) \frac{\partial \lambda_3^{\pm}}{\partial w} \right) \\ \frac{\partial F_{\rho v}^{\pm}}{\partial a} &= \frac{\rho_s}{2\gamma} \left(2(\gamma - 1)v \frac{\partial \lambda_1^{\pm}}{\partial a} + (v + an_y) \frac{\partial \lambda_2^{\pm}}{\partial a} + (v - an_y) \frac{\partial \lambda_3^{\pm}}{\partial a} \right) + \\ &\quad \frac{\rho}{2\gamma} [\lambda_2^{\pm} n_y - \lambda_3^{\pm} n_y] \end{aligned}, \quad (\text{B.22})$$

$$\frac{\partial F_{\rho v}^{\pm}}{\partial e_{ve}} = \frac{\rho_s}{2\gamma} \left(2(\gamma - 1)v \frac{\partial \lambda_1^{\pm}}{\partial e_{ve}} + (v + an_y) \frac{\partial \lambda_2^{\pm}}{\partial e_{ve}} + (v - an_y) \frac{\partial \lambda_3^{\pm}}{\partial e_{ve}} \right)$$

$$\begin{aligned}
\frac{\partial F_{\rho w}^{\pm}}{\partial \rho_r} &= \frac{1}{2\gamma} (2(\gamma-1)w\lambda_1^{\pm} + (w+an_z)\lambda_2^{\pm} + (w-an_z)\lambda_3^{\pm}) \\
\frac{\partial F_{\rho w}^{\pm}}{\partial u} &= \frac{\rho_s}{2\gamma} \left(2(\gamma-1)w\frac{\partial \lambda_1^{\pm}}{\partial u} + (w+an_z)\frac{\partial \lambda_2^{\pm}}{\partial u} + (w-an_z)\frac{\partial \lambda_3^{\pm}}{\partial u} \right) \\
\frac{\partial F_{\rho w}^{\pm}}{\partial v} &= \frac{\rho_s}{2\gamma} \left(2(\gamma-1)w\frac{\partial \lambda_1^{\pm}}{\partial v} + (w+an_z)\frac{\partial \lambda_2^{\pm}}{\partial v} + (w-an_z)\frac{\partial \lambda_3^{\pm}}{\partial v} \right) \\
\frac{\partial F_{\rho w}^{\pm}}{\partial w} &= \frac{\rho_s}{2\gamma} \left(2(\gamma-1)w\frac{\partial \lambda_1^{\pm}}{\partial w} + (w+an_z)\frac{\partial \lambda_2^{\pm}}{\partial w} + (w-an_z)\frac{\partial \lambda_3^{\pm}}{\partial w} \right) + \\
&\quad \frac{\rho}{2\gamma} (2(\gamma-1)\lambda_1^{\pm} + \lambda_2^{\pm} + \lambda_3^{\pm}) , \tag{B.23}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial F_{\rho w}^{\pm}}{\partial a} &= \frac{\rho_s}{2\gamma} \left(2(\gamma-1)w\frac{\partial \lambda_1^{\pm}}{\partial a} + (w+an_z)\frac{\partial \lambda_2^{\pm}}{\partial a} + (w-an_z)\frac{\partial \lambda_3^{\pm}}{\partial a} \right) + \\
&\quad \frac{\rho}{2\gamma} [\lambda_2^{\pm} n_z - \lambda_3^{\pm} n_z] \\
\frac{\partial F_{\rho w}^{\pm}}{\partial e_{ve}} &= \frac{\rho_s}{2\gamma} \left(2(\gamma-1)w\frac{\partial \lambda_1^{\pm}}{\partial e_{ve}} + (w+an_z)\frac{\partial \lambda_2^{\pm}}{\partial e_{ve}} + (w-an_z)\frac{\partial \lambda_3^{\pm}}{\partial e_{ve}} \right)
\end{aligned}$$

$$\begin{aligned}
\frac{\partial F_E^{\pm}}{\partial \rho_r} &= \frac{1}{2\gamma} (2(\gamma-1)\epsilon_1\lambda_1^{\pm} + \epsilon_2\lambda_2^{\pm} + \epsilon_3\lambda_3^{\pm}) \\
\frac{\partial F_E^{\pm}}{\partial u} &= \frac{\rho_s}{2\gamma} \left(2(\gamma-1)\epsilon_1\frac{\partial \lambda_1^{\pm}}{\partial u} + \epsilon_2\frac{\partial \lambda_2^{\pm}}{\partial u} + \epsilon_3\frac{\partial \lambda_3^{\pm}}{\partial u} \right) + \\
&\quad \frac{\rho}{2\gamma} (2(\gamma-1)u\lambda_1^{\pm} + (u+an_x)\lambda_2^{\pm} + (u-an_x)\lambda_3^{\pm}) \\
\frac{\partial F_E^{\pm}}{\partial v} &= \frac{\rho_s}{2\gamma} \left(2(\gamma-1)\epsilon_1\frac{\partial \lambda_1^{\pm}}{\partial v} + \epsilon_2\frac{\partial \lambda_2^{\pm}}{\partial v} + \epsilon_3\frac{\partial \lambda_3^{\pm}}{\partial v} \right) + \\
&\quad \frac{\rho}{2\gamma} (2(\gamma-1)v\lambda_1^{\pm} + (v+an_y)\lambda_2^{\pm} + (v-an_y)\lambda_3^{\pm}) \\
\frac{\partial F_E^{\pm}}{\partial w} &= \frac{\rho_s}{2\gamma} \left(2(\gamma-1)\epsilon_1\frac{\partial \lambda_1^{\pm}}{\partial w} + \epsilon_2\frac{\partial \lambda_2^{\pm}}{\partial w} + \epsilon_3\frac{\partial \lambda_3^{\pm}}{\partial w} \right) + \\
&\quad \frac{\rho}{2\gamma} (2(\gamma-1)w\lambda_1^{\pm} + (w+an_z)\lambda_2^{\pm} + (w-an_z)\lambda_3^{\pm}) , \tag{B.24}
\end{aligned}$$

$$\begin{aligned}
\frac{\partial F_E^{\pm}}{\partial a} &= \frac{\rho_s}{2\gamma} \left(2(\gamma-1)\epsilon_1\frac{\partial \lambda_1^{\pm}}{\partial a} + \epsilon_2\frac{\partial \lambda_2^{\pm}}{\partial a} + \epsilon_3\frac{\partial \lambda_3^{\pm}}{\partial a} \right) + \\
&\quad \frac{\rho}{2\gamma} \lambda_2^{\pm} \left(\frac{(3-\gamma)a}{2(\gamma-1)} + n_x(u+an_x) + n_y(v+an_y) + n_z(w+an_z) \right) + \\
&\quad \frac{\rho}{2\gamma} \lambda_3^{\pm} \left(\frac{(3-\gamma)a}{2(\gamma-1)} - n_x(u-an_x) - n_y(v-an_y) - n_z(w-an_z) \right) \\
\frac{\partial F_E^{\pm}}{\partial e_{ve}} &= \frac{\rho_s}{2\gamma} \left(2(\gamma-1)\epsilon_1\frac{\partial \lambda_1^{\pm}}{\partial e_{ve}} + \epsilon_2\frac{\partial \lambda_2^{\pm}}{\partial e_{ve}} + \epsilon_3\frac{\partial \lambda_3^{\pm}}{\partial e_{ve}} \right) + \\
&\quad \frac{\rho}{2\gamma} (2(\gamma-1)\lambda_1^{\pm} + \lambda_2^{\pm} + \lambda_3^{\pm})
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial F_{E_v}^\pm}{\partial \rho_r} &= \frac{e_{ve}}{2\gamma} (2(\gamma - 1)\lambda_1^\pm + \lambda_2^\pm + \lambda_3^\pm) \\
\frac{\partial F_{E_v}^\pm}{\partial u} &= \frac{\rho e_{ve}}{2\gamma} \left(2(\gamma - 1) \frac{\partial \lambda_1^\pm}{\partial u} + \frac{\partial \lambda_2^\pm}{\partial u} + \frac{\partial \lambda_3^\pm}{\partial u} \right) \\
\frac{\partial F_{E_v}^\pm}{\partial v} &= \frac{\rho e_{ve}}{2\gamma} \left(2(\gamma - 1) \frac{\partial \lambda_1^\pm}{\partial v} + \frac{\partial \lambda_2^\pm}{\partial v} + \frac{\partial \lambda_3^\pm}{\partial v} \right) \\
\frac{\partial F_{E_v}^\pm}{\partial w} &= \frac{\rho e_{ve}}{2\gamma} \left(2(\gamma - 1) \frac{\partial \lambda_1^\pm}{\partial w} + \frac{\partial \lambda_2^\pm}{\partial w} + \frac{\partial \lambda_3^\pm}{\partial w} \right) \\
\frac{\partial F_{E_v}^\pm}{\partial a} &= \frac{\rho e_{ve}}{2\gamma} \left(2(\gamma - 1) \frac{\partial \lambda_1^\pm}{\partial a} + \frac{\partial \lambda_2^\pm}{\partial a} + \frac{\partial \lambda_3^\pm}{\partial a} \right) \\
\frac{\partial F_{E_v}^\pm}{\partial e_{ve}} &= \frac{\rho e_{ve}}{2\gamma} \left(2(\gamma - 1) \frac{\partial \lambda_1^\pm}{\partial e_{ve}} + \frac{\partial \lambda_2^\pm}{\partial e_{ve}} + \frac{\partial \lambda_3^\pm}{\partial e_{ve}} \right) + \frac{\rho}{2\gamma} (2(\gamma - 1)\lambda_1^\pm + \lambda_2^\pm + \lambda_3^\pm)
\end{aligned} \quad . \quad (\text{B.25})$$

The eigenvalues in those equations are given by

$$\begin{aligned}
\lambda_1^\pm &= \frac{1}{2} \left(u' \pm \sqrt{(u')^2 + \epsilon^2} \right) \\
\lambda_2^\pm &= \frac{1}{2} \left(u' + a \pm \sqrt{(u' + a)^2 + \epsilon^2} \right) \\
\lambda_3^\pm &= \frac{1}{2} \left(u' - a \pm \sqrt{(u' - a)^2 + \epsilon^2} \right)
\end{aligned} \quad , \quad (\text{B.26})$$

where $u' = un_x + vn_y + wn_z$ and ϵ is given in Chapter III, and their derivatives are given by

$$\begin{aligned}
\frac{\partial \lambda_1^\pm}{\partial \rho_s} &= 0 \\
\frac{\partial \lambda_1^\pm}{\partial u} &= \frac{n_x}{2} \left(1 \pm \frac{u'}{\sqrt{(u')^2 + \epsilon^2}} \right) \\
\frac{\partial \lambda_1^\pm}{\partial v} &= \frac{n_y}{2} \left(1 \pm \frac{u'}{\sqrt{(u')^2 + \epsilon^2}} \right) \\
\frac{\partial \lambda_1^\pm}{\partial w} &= \frac{n_z}{2} \left(1 \pm \frac{u'}{\sqrt{(u')^2 + \epsilon^2}} \right) \\
\frac{\partial \lambda_1^\pm}{\partial a} &= 0 \\
\frac{\partial \lambda_1^\pm}{\partial e_{ve}} &= 0
\end{aligned} \quad , \quad (\text{B.27})$$

$$\begin{aligned}
\frac{\partial \lambda_2^\pm}{\partial \rho_s} &= 0 \\
\frac{\partial \lambda_2^\pm}{\partial u} &= \frac{n_x}{2} \left(1 \pm \frac{(u'+a)}{\sqrt{(u'+a)^2 + \epsilon^2}} \right) \\
\frac{\partial \lambda_2^\pm}{\partial v} &= \frac{n_y}{2} \left(1 \pm \frac{(u'+a)}{\sqrt{(u'+a)^2 + \epsilon^2}} \right) \\
\frac{\partial \lambda_2^\pm}{\partial w} &= \frac{n_z}{2} \left(1 \pm \frac{(u'+a)}{\sqrt{(u'+a)^2 + \epsilon^2}} \right) \\
\frac{\partial \lambda_2^\pm}{\partial a} &= \frac{1}{2} \left(1 \pm \frac{(u'+a)}{\sqrt{(u'+a)^2 + \epsilon^2}} \right) \\
\frac{\partial \lambda_2^\pm}{\partial e_{ve}} &= 0
\end{aligned} , \tag{B.28}$$

and

$$\begin{aligned}
\frac{\partial \lambda_3^\pm}{\partial \rho_s} &= 0 \\
\frac{\partial \lambda_3^\pm}{\partial u} &= \frac{n_x}{2} \left(1 \pm \frac{(u'-a)}{\sqrt{(u'-a)^2 + \epsilon^2}} \right) \\
\frac{\partial \lambda_3^\pm}{\partial v} &= \frac{n_y}{2} \left(1 \pm \frac{(u'-a)}{\sqrt{(u'-a)^2 + \epsilon^2}} \right) \\
\frac{\partial \lambda_3^\pm}{\partial w} &= \frac{n_z}{2} \left(1 \pm \frac{(u'-a)}{\sqrt{(u'-a)^2 + \epsilon^2}} \right) \\
\frac{\partial \lambda_3^\pm}{\partial a} &= -\frac{1}{2} \left(1 \pm \frac{(u'-a)}{\sqrt{(u'-a)^2 + \epsilon^2}} \right) \\
\frac{\partial \lambda_3^\pm}{\partial e_{ve}} &= 0
\end{aligned} . \tag{B.29}$$

B.3 Viscous Jacobians

The derivation of the viscous Jacobians is easier using a face based reference frame define by the vectors \vec{n} , \vec{t} and \vec{r} . Based on that reference frame, the viscous

fluxes are given by

$$F_{vn} = \left\{ \begin{array}{l} -J_{n,1} \\ \vdots \\ -J_{n,ns} \\ \tau_{nn} \\ \tau_{nt} \\ \tau_{nr} \\ \tau_{nn}u_n + \tau_{nt}u_t + \tau_{nr}u_r - (q_{tr,n} + q_{ve,n}) - \sum(J_{n,s}h_s) \\ -q_{ve,n} - \sum(J_{n,s}e_{ve,s}) \end{array} \right\}, \quad (\text{B.30})$$

and the fluxes in the t and r directions are similar but not used because the normal vector to the face in this reference frame is $(1, 0, 0)$ so that

$$F'_v = F_{vn} \times 1 + F_{vt} \times 0 + F_{vr} \times 0 \quad . \quad (\text{B.31})$$

To map the fluxes from the face based reference frame to the cartesian frame, the rotation matrices

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & n_x & n_y & n_z & 0 & 0 \\ 0 & t_x & t_y & t_z & 0 & 0 \\ 0 & r_x & r_y & r_z & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad R^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & n_x & t_x & r_x & 0 & 0 \\ 0 & n_y & t_y & r_y & 0 & 0 \\ 0 & n_z & t_z & r_z & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{B.32})$$

are used so that

$$F_v = R^{-1}F'_v \quad . \quad (\text{B.33})$$

Using the thin-layer approximation, the derivatives $\partial/\partial t$ and $\partial/\partial r$ are neglected

leaving only $\partial/\partial n$. With that approximation, the viscous flux F_{vn} becomes

$$F_{vn} = \left\{ \begin{array}{l} -\rho D_1 \frac{\partial Y_1}{\partial n} \\ \vdots \\ -\rho D_{ns} \frac{\partial Y_{ns}}{\partial n} \\ (2\mu + \lambda) \frac{\partial u_n}{\partial n} \\ \mu \frac{\partial u_t}{\partial n} \\ \mu \frac{\partial u_r}{\partial n} \\ (2\mu + \lambda) \frac{\partial u_n}{\partial n} u_n + \mu \frac{\partial u_t}{\partial n} u_t + \mu \frac{\partial u_r}{\partial n} u_r - \\ (-\kappa_{tr,n} \frac{\partial T_{tr,n}}{\partial n} - \kappa_{ve,n} \frac{\partial T_{ve,n}}{\partial n}) - \sum (-\rho D_s \frac{\partial Y_s}{\partial n} h_s) \\ -\kappa_{ve,n} \frac{\partial T_{ve,n}}{\partial n} - \sum (-\rho D_s \frac{\partial Y_s}{\partial n} e_{ve,s}) \end{array} \right\}. \quad (\text{B.34})$$

To obtain the Jacobian, F_{vn} is linearized assuming that terms that do not involve derivatives are constant such that

$$F_{vn} = M \frac{\partial V_n}{\partial n} \quad (\text{B.35})$$

where

$$V_n = \left\{ \begin{array}{l} Y_s \\ u_n \\ v_n \\ w \\ T \\ T_{ve} \end{array} \right\}. \quad (\text{B.36})$$

The derivative in the normal direction is approximated by

$$F_{vn} = M \frac{\Delta V_n}{\Delta n} = M_n (V_{nR} - V_{nL}) \quad (\text{B.37})$$

where $M_n = M/\Delta_n$ and the V_{nR} and V_{nL} are the property vectors on either side of the face. The Jacobian in the cartesian coordinate frame is obtained using the

rotation matrices as

$$F_v = R^{-1} M_n R (V_R - V_L) = R^{-1} M_n R N (Q_R - Q_L) \quad (\text{B.38})$$

and the viscous jacobian is finally given by

$$\frac{\partial F_v}{\partial Q_R} = R^{-1} M_n R N \quad \text{and} \quad \frac{\partial F_v}{\partial Q_L} = -R^{-1} M_n R N \quad (\text{B.39})$$

The matrix M_n is given by

$$M = \begin{bmatrix} \frac{\rho D_s}{\Delta_n} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{(2\mu+\lambda)}{\Delta_n} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\mu}{\Delta_n} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\mu}{\Delta_n} & 0 & 0 \\ \frac{\rho D_s h_s}{\Delta_n} & \frac{(2\mu+\lambda)u_n}{\Delta_n} & \frac{\mu u_t}{\Delta_n} & \frac{\mu u_r}{\Delta_n} & \frac{\kappa_{tv}}{\Delta_n} & \frac{\kappa_{ve}}{\Delta_n} \\ \frac{\rho D_s e_{ve,s}}{\Delta_n} & 0 & 0 & 0 & \frac{\kappa_{ve}}{\Delta_n} & 0 \end{bmatrix}. \quad (\text{B.40})$$

B.3.1 Implicit Boundary Conditions

For faces at boundaries, the viscous Jacobians are modified so that

$$\frac{\partial F_v}{\partial Q_R} = 0 \quad \text{and} \quad \frac{\partial F_v}{\partial Q_L} = R^{-1} M_n R (F - I) N \quad (\text{B.41})$$

where the folding matrices F are given in Chapter III and I is the identity matrix.

B.4 Chemistry Jacobians

The chemistry Jacobians in its matrix form are given by

$$\frac{\partial S_c}{\partial Q} = \begin{bmatrix} \frac{\partial \dot{w}_s}{\partial \rho_r} & \frac{\partial \dot{w}_s}{\partial u} & \frac{\partial \dot{w}_s}{\partial v} & \frac{\partial \dot{w}_s}{\partial w} & \frac{\partial \dot{w}_s}{\partial E} & \frac{\partial \dot{w}_s}{\partial E_v} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{B.42})$$

where

$$\frac{\partial \dot{w}_s}{\partial Q} = M_s \sum_k (\beta_{k,s} - \alpha_{k,s}) \left[\frac{\partial k_{f,k}}{\partial Q} \frac{R_{f,k}}{k_{f,k}} + k_{f,k} \frac{\partial}{\partial Q} \left(\frac{R_{f,k}}{k_{f,k}} \right) - \frac{\partial k_{b,k}}{\partial Q} \frac{R_{b,k}}{k_{b,k}} - k_{b,k} \frac{\partial}{\partial Q} \left(\frac{R_{b,k}}{k_{b,k}} \right) \right] \quad (\text{B.43})$$

The derivatives of the forward reaction rate coefficients, $\frac{\partial k_f}{\partial Q}$, are given by

$$\frac{\partial k_f}{\partial Q} = k_f \left(\frac{n_f}{T'_f} + \frac{E_f}{T'^2_f} \right) \frac{dT'_f}{dT_f} \left(a_f \frac{T_f}{T_{tr}} \frac{\partial T_{tr}}{\partial Q} + b_f \frac{T_f}{T_{ve}} \frac{\partial T_{ve}}{\partial Q} \right) \quad (\text{B.44})$$

where

$$\frac{dT'_f}{dT_f} = \frac{1}{2} + \frac{1}{2} \frac{T_f - T_{min}}{\sqrt{(T_f - T_{min})^2 + \epsilon^2}} \quad (\text{B.45})$$

The derivatives of the backward reaction rate coefficients, $\frac{\partial k_b}{\partial Q}$, are given by

$$\frac{\partial k_b}{\partial Q} = -k_b \left(\frac{1}{Kc} \frac{dKc}{dT'_b} \right) \left(a_b \frac{T_b}{T_{tr}} \frac{\partial T_{tr}}{\partial Q} + b_b \frac{T_b}{T_{ve}} \frac{\partial T_{ve}}{\partial Q} \right) + \frac{1}{Kc} \frac{\partial k_{fb}}{\partial Q} \quad (\text{B.46})$$

The last expression use the derivatives of the “forward-backward” reaction rate coefficient divided by the equilibrium constant $\frac{1}{Kc} \frac{\partial k_{fb}}{\partial Q}$ as

$$\frac{1}{Kc} \frac{\partial k_{fb}}{\partial Q} = k_b \left(\frac{n_f}{T'_b} + \frac{E_f}{T'^2_b} \right) \frac{dT'_b}{dT_b} \left(a_b \frac{T_b}{T_{tr}} \frac{\partial T_{tr}}{\partial Q} + b_b \frac{T_b}{T_{ve}} \frac{\partial T_{ve}}{\partial Q} \right) \quad (\text{B.47})$$

and the derivative of the equilibrium constants, for Park’s 1990 air model, given by

$$\frac{1}{Kc} \frac{dKc}{dT_b} = - \left(-\frac{B_0}{Z} + B_2 + B_3 Z + 2B_4 Z^2 \right) \frac{1}{T'_b} \frac{dT'_b}{dT_b} \quad (\text{B.48})$$

The expression for dT'_b/dT_b is similar to dT'_f/dT_f .

The derivatives of the forward reaction rate divided by the forward reaction rate coefficient, $\frac{\partial}{\partial Q} \left(\frac{R_{f,k}}{k_{f,k}} \right)$, are given by

$$\begin{aligned} \frac{\partial}{\partial \rho_l} \left(\frac{R_{f,k}}{k_{f,k}} \right) &= 1000 \left[0.001 \frac{\alpha_{k,l}}{M_l} \left(\frac{0.001 \rho_l}{M_l} \right)^{\max(0, \alpha_{k,l}-1)} \prod_{j \neq l} \left(\frac{0.001 \rho_l}{M_l} \right)^{\alpha_{k,j}} \right] \\ \frac{\partial}{\partial \rho_u} \left(\frac{R_{f,k}}{k_{f,k}} \right) &= 0 \\ \frac{\partial}{\partial \rho_v} \left(\frac{R_{f,k}}{k_{f,k}} \right) &= 0 \\ \frac{\partial}{\partial \rho_w} \left(\frac{R_{f,k}}{k_{f,k}} \right) &= 0 \\ \frac{\partial}{\partial E} \left(\frac{R_{f,k}}{k_{f,k}} \right) &= 0 \\ \frac{\partial}{\partial E_v} \left(\frac{R_{f,k}}{k_{f,k}} \right) &= 0 \end{aligned} . \quad (\text{B.49})$$

The derivatives of the translational-rotational temperature are given by

$$\begin{aligned} \frac{\partial T_{tr}}{\partial \rho_s} &= \frac{1}{\rho C v_{tr,mix}} \left[-C v_{tr,s} T_{tr} + \frac{1}{2}(u^2 + v^2 + w^2) - h_s^o \right] \\ \frac{\partial T_{tr}}{\partial \rho_u} &= -\frac{u}{\rho C v_{tr,mix}} \\ \frac{\partial T_{tr}}{\partial \rho_v} &= -\frac{v}{\rho C v_{tr,mix}} \\ \frac{\partial T_{tr}}{\partial \rho_w} &= -\frac{w}{\rho C v_{tr,mix}} \\ \frac{\partial T_{tr}}{\partial E} &= \frac{1}{\rho C v_{tr,mix}} \\ \frac{\partial T_{tr}}{\partial E_v} &= -\frac{1}{\rho C v_{tr,mix}} \end{aligned} . \quad (\text{B.50})$$

and the derivatives for the vibrational-electron-electronic temperature are given by

$$\begin{aligned} \frac{\partial T_{ve}}{\partial \rho_s} &= \frac{e_{ve,s}}{\rho C v_{ve,mix}} \\ \frac{\partial T_{ve}}{\partial \rho_u} &= 0 \\ \frac{\partial T_{ve}}{\partial \rho_v} &= 0 \\ \frac{\partial T_{ve}}{\partial \rho_w} &= 0 \\ \frac{\partial T_{ve}}{\partial E} &= 0 \\ \frac{\partial T_{ve}}{\partial E_v} &= \frac{1}{\rho C v_{ve,mix}} \end{aligned} . \quad (\text{B.51})$$

The derivatives of the backward reaction rate divided by the backward reaction rate coefficient are similar.

B.5 Non-Equilibrium Jacobians

The non-equilibrium Jacobians in matrix form are given by

$$\frac{\partial S_c}{\partial Q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\partial \dot{w}_v}{\partial \rho_r} & \frac{\partial \dot{w}_v}{\partial u} & \frac{\partial \dot{w}_v}{\partial v} & \frac{\partial \dot{w}_v}{\partial w} & \frac{\partial \dot{w}_v}{\partial E} & \frac{\partial \dot{w}_v}{\partial E_v} \end{bmatrix} \quad (\text{B.52})$$

where $\frac{\partial \dot{w}_v}{\partial Q}$ is composed of many different terms

$$\frac{\partial \dot{S}_v}{\partial Q} = \frac{\partial S_{epg}}{\partial Q} + \frac{\partial S_{c-v}}{\partial Q} + \frac{\partial S_{t-v}}{\partial Q} + \frac{\partial S_{h-e}}{\partial Q} + \frac{\partial S_{e-i}}{\partial Q} \quad (\text{B.53})$$

which will be detailed below. The derivatives of S_{epg} and S_{h-e} are neglected in this work, or

$$\frac{\partial S_{epg}}{\partial Q} = 0 \quad (\text{B.54})$$

and

$$\frac{\partial S_{h-e}}{\partial Q} = 0 \quad (\text{B.55})$$

This is an approximation that does not seem to have an impact on the stability of the problems simulated.

The derivatives of the translational-vibrational energy transfer term are given by

$$\begin{aligned} \frac{\partial S_{t-v}}{\partial \rho_s} &= \frac{\rho C v_{ve,mix}}{\tau} \left(\frac{\partial T_{tr}}{\partial \rho_s} - \frac{\partial T_{ve}}{\partial \rho_s} \right) + \frac{C v_{ve,s}}{\tau} (T_{tr} - T_{ve}) \\ \frac{\partial S_{t-v}}{\partial \rho_u} &= \frac{\rho C v_{ve,mix}}{\tau} \left(\frac{\partial T_{tr}}{\partial \rho_u} - \frac{\partial T_{ve}}{\partial \rho_u} \right) \\ \frac{\partial S_{t-v}}{\partial \rho_v} &= \frac{\rho C v_{ve,mix}}{\tau} \left(\frac{\partial T_{tr}}{\partial \rho_v} - \frac{\partial T_{ve}}{\partial \rho_v} \right) \\ \frac{\partial S_{t-v}}{\partial \rho_w} &= \frac{\rho C v_{ve,mix}}{\tau} \left(\frac{\partial T_{tr}}{\partial \rho_w} - \frac{\partial T_{ve}}{\partial \rho_w} \right) \\ \frac{\partial S_{t-v}}{\partial E} &= \frac{\rho C v_{ve,mix}}{\tau} \left(\frac{\partial T_{tr}}{\partial E} - \frac{\partial T_{ve}}{\partial E} \right) \\ \frac{\partial S_{t-v}}{\partial E_v} &= \frac{\rho C v_{ve,mix}}{\tau} \left(\frac{\partial T_{tr}}{\partial E_v} - \frac{\partial T_{ve}}{\partial E_v} \right) \end{aligned} \quad (\text{B.56})$$

It should be observed that in the previous derivatives the relaxation time τ is assumed constant. This is another approximation that is used in the derivation of non-equilibrium Jacobians[24] that does not impact the stability of the numerical code.

The derivatives of the chemistry term are given by

$$\frac{\partial S_{c-v}}{\partial Q} = \sum_s \left[\frac{\partial \dot{w}_s}{\partial Q} (D_s + e_{el,s}) + \dot{w} \left(\frac{\partial D_s}{\partial Q} + Cv_{el,s} \frac{\partial T_{ve}}{\partial Q} \right) \right] \quad (\text{B.57})$$

where

$$\frac{\partial D_s}{\partial Q} = \begin{cases} 0 & \text{for the preferential model,} \\ Cv_{v,s} \frac{\partial T_{ve}}{\partial Q} & \text{for the non-preferential model.} \end{cases} \quad (\text{B.58})$$

Finally, the derivatives for the electron impact ionization term are given by

$$\frac{\partial S_{e-i}}{\partial Q} = M_{N^+} \hat{I}_N \frac{\partial \dot{w}_{N^+,eii}}{\partial Q} + M_{O^+} \hat{I}_O \frac{\partial \dot{w}_{O^+,eii}}{\partial Q} \quad . \quad (\text{B.59})$$

APPENDIX C

Axisymmetric Formulation and Axisymmetric Jacobian

The implemented numerical code is able to solve the set of equations assuming axisymmetry. In that case, the spatially integrated equations using the finite-volume approach are given by[65]

$$y_i A_i \frac{\partial Q_i}{\partial t} = - \sum_j (\vec{F}_j - \vec{F}_{dj}) \cdot \vec{n}_j y_j s_j + y_i A_i S_{cv,i} + A_i S_{a,i} = R_i . \quad (C.1)$$

where $\vec{F} = E\vec{i} + F\vec{j}$ is the inviscid flux at the face, $\vec{F}_d = E_d\vec{i} + F_d\vec{j}$ is the diffusive flux at the face, $S_{cv,i}$ is the chemistry and non-equilibrium source term, $S_{a,i}$ is the source term that arises from the integration being performed assuming axisymmetry, A_i is the area of the i-th cell, \vec{n} is the normal vector to the j-th face and s_j is the length of the j-th face. The radial coordinate of the i-th cell, y_i , is measured from its centroid. The radial coordinate of the j-th face, y_j , is measured from its mid-point. The term

S_{ai} can be written as

$$S_{ai} = \begin{Bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ p - 2\mu \frac{v}{y_i} - \lambda \vec{\nabla} \cdot \vec{u} \\ 0 \\ 0 \end{Bmatrix} . \quad (\text{C.2})$$

In the axisymmetric formulation, $\vec{\nabla} \cdot \vec{u}$ in the previous expression and in the stress tensor is given by

$$\vec{\nabla} \cdot \vec{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{v}{y} . \quad (\text{C.3})$$

The inviscid, viscous, chemistry and non-equilibrium Jacobians are the same as in the three-dimensional case excluding the momentum equation in the z direction. It is indicated to divide the only non zero term of axisymmetric source term in two parts for the calculation of the jacobian as $S_{ai,v} = A + B$

$$A = \begin{Bmatrix} 0 \\ 0 \\ p - (2\mu + \lambda) \frac{v}{y} \\ 0 \\ 0 \end{Bmatrix} \quad (\text{C.4})$$

and

$$B = \begin{Bmatrix} 0 \\ 0 \\ -\lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \\ 0 \\ 0 \end{Bmatrix} . \quad (\text{C.5})$$

The Jacobian of A is straightforward to calculate and is given by

$$\frac{\partial A}{\partial Q} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \frac{\partial A}{\partial \rho_s} & \frac{\partial A}{\partial \rho u} & \frac{\partial A}{\partial \rho v} & \frac{\partial A}{\partial E} & \frac{\partial A}{\partial E_v} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{C.6})$$

where

$$\begin{aligned} \frac{\partial A}{\partial \rho_s} &= \frac{\partial p}{\partial \rho_s} + (2\mu + \lambda) \frac{v}{y\rho} \\ \frac{\partial A}{\partial \rho u} &= \frac{\partial p}{\partial \rho u} \\ \frac{\partial A}{\partial \rho v} &= \frac{\partial p}{\partial \rho v} - (2\mu + \lambda) \frac{1}{y\rho} \\ \frac{\partial A}{\partial E} &= \frac{\partial p}{\partial E} \\ \frac{\partial A}{\partial E_v} &= \frac{\partial p}{\partial E_v} \end{aligned} . \quad (\text{C.7})$$

The Jacobian for B requires some approximation. This Jacobian is not a function of the properties at the i -th cell only because it involves derivatives. Using Gauss' theorem, the derivatives in the cell can be substituted by

$$B_v = -\lambda \frac{1}{V_i} \sum_f \vec{u} \cdot \vec{n} S . \quad (\text{C.8})$$

Then, B can be written as $B = \sum_f M_{a,f} V_f$ where

$$M_a = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{\lambda n_x S}{V_i} & -\frac{\lambda n_y S}{V_i} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{C.9})$$

and $V_f = 0.5(V_L + V_R)$. Now, using conserved variables,

$$B = \sum_f M_{a,f} N_f \frac{Q_i + Q_{nb}}{2} \quad (\text{C.10})$$

and then

$$\frac{\partial B}{\partial Q_i} = \frac{1}{2} \sum_f M_{a,f} N_f \quad . \quad (\text{C.11})$$

The terms $\frac{\partial B}{\partial Q_{nb}}$ are neglected in this work. Those terms could be stored at the faces, similarly to what is done with the inviscid and viscous Jacobians, and used in the iterative procedure. That approach is not used in this work so that all source term Jacobians are treated locally.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] ATLAS homepage. <http://math-atlas.sourceforge.net/>.
- [2] BLAS homepage. <http://www.netlib.org/blas/>.
- [3] LAPACK homepage. <http://www.netlib.org/lapack/>.
- [4] PETSc homepage. <http://www.mcs.anl.gov/petsc>.
- [5] *FLUENT 5 User's Guide*. Fluent, 2004.
- [6] *GAMBIT 2.2 User's Guide*. Fluent, 2004.
- [7] ANDERSON, E., BAI, Z., BISCHOF, C., BLACKFORD, S., DEMMEL, J., DON-GARRA, J., CROZ, J. D., GREENBAUM, A., HAMMARLING, S., MCKENNEY, A., AND SORENSEN, D. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, 1999.
- [8] BALAY, S., BUSCHELMAN, K., EIJKHOUT, V., GROPP, W. D., KAUSHIK, D., KNEPLEY, M. G., MCINNES, L. C., SMITH, B. F., AND ZHANG, H. PETSc users manual. Tech. Rep. ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.
- [9] BARTH, T. J., AND JESPERSEN, D. C. The design and application of upwind schemes on unstructured meshes. AIAA Paper 1989-0366, 1989.
- [10] BATINA, J. Three-dimensional flux-split Euler schemes involving unstructured dynamic meshes. AIAA Paper 1990-1649, 1990.
- [11] BERTIN, J. J. *Hypersonic Aerothermodynamics*. AIAA Education Series, 1994.
- [12] BIBB, K. L., PERAIRE, J., AND RILEY, C. J. Hypersonic flow computations on unstructured meshes. AIAA Paper 1997-0625, 1997.
- [13] BLACKFORD, L. S., DEMMEL, J., DONGARRA, J., DUFF, I., HAMMARLING, S., HENRY, G., HEROUX, M., KAUFMAN, L., LUMSDAINE, A., PETITET, A., POZO, R., REMINGTON, K., AND WHALEY, R. C. An updated set of basic linear algebra subprograms (BLAS). *ACM Trans. Math. Soft.* 28, 135–151.

- [14] BLOTTNER, F. G., JOHNSON, M., AND ELLIS, M. Chemically reacting viscous flow program for multi-component gas mixtures. SC-RR-70-754, Sandia Laboratories, Albuquerque, New Mexico, 1971.
- [15] BOYLAND, D. E., AND GRIFFITH, B. J. Simulation of the Apollo command module aerodynamics at re-entry altitudes. In *3rd National Conference on Aerospace Meteorology*. New Orleans, Louisiana, 1968.
- [16] BUNING, P. G., AND STEGER, J. L. Solution of the two-dimensional Euler equations with generalized coordinate transformations using flux vector splitting. AIAA Paper 1982-0971, 1982.
- [17] CANDLER, G. Computation of thermo-chemical nonequilibrium Martian atmospheric entry flows. AIAA Paper 1990-1695, 1990.
- [18] CANDLER, G. V. *The Computational of Weakly Ionized Flow in Nonequilibrium*. PhD thesis, Stanford University, California, June 1988.
- [19] CHEN, Y. K., HENLINE, W. D., AND TAUBER, M. E. Trajectory based heating and ablation calculation for Mars Pathfinder aeroshell. *Journal of Spacecraft and Rockets* 32 (1995), 225–230.
- [20] DIETRICH, S., AND BOYD, I. D. Scalar and parallel optimized implementation of the direct simulation Monte Carlo method. *Journal of Computational Physics*. 126 (1996), 328–342.
- [21] DRUGUET, M., CANDLER, G. V., AND NOMPESIS, I. Effect of numerics on Navier-Stokes computations of hypersonic double-cone flows. *AIAA Journal* 43, 3 (2005), 616–623.
- [22] EBERHARDT, S., AND BROWN, K. A shock capturing technique for hypersonic, chemically relaxing flows. AIAA Paper 1986-0231, 1986.
- [23] GNOFFO, P. A. Computational aerothermodynamics in aeroassist applications. *Journal of Spacecraft and Rockets* 40, 3 (2003), 305–312.
- [24] GNOFFO, P. A., GUPTA, R. N., AND SHINN, J. L. Conservation equations and physical models for hypersonic air flows in thermal and chemical nonequilibrium. NASA-TP-2867, NASA Langley, Hampton, Virginia, 1989.
- [25] GNOFFO, P. A., AND MCCANDLESS, R. S. Three-dimensional AOTV flow-fields in chemical nonequilibrium. AIAA Paper 1986-0230, 1986.
- [26] GNOFFO, P. A., WEILMUENSTER, K. J., AND ALTER, S. J. A multiblock analysis for Shuttle orbiter re-entry heating from Mach 24 to Mach 12. AIAA Paper 1993-2813, 1993.
- [27] GNOFFO, P. A., WEILMUENSTER, K. J., BRAUN, R. D., AND CRUZ, C. I. Influence of sonic-line location on Mars pathfinder probe aerothermodynamics. *Journal of Spacecraft and Rockets* 33 (1996), 169–177.

- [28] GNOFFO, P. A., WEILMUNSTER, K. J., HAMILTON II, H. H., OLYNICK, D. R., AND VENKATAPATHY, E. Computational aerothermodynamic design issues for hypersonic vehicles. *Journal of Spacecraft and Rockets* 36 (1999), 21–43.
- [29] GOSSE, R., AND CANDLER, G. Diffusion flux modeling: Application to direct entry problems. AIAA Paper 2005-0389, 2005.
- [30] GRASSO, F., AND CAPANO, G. Modeling of ionizing hypersonic flows in nonequilibrium. *Journal of Spacecraft and Rockets* 32, 2 (1995), 217–224.
- [31] GRASSO, F., AND GNOFFO, P. A. A numerical study of hypersonic stagnation heat transfer predictions at a coordinate singularity. In *Proceedings of the Eighth GAMM-Conference on Numerical Methods in Fluid Mechanics*, vol. 29. Vieweg, Delft, Netherlands, 1989, pp. 179–188.
- [32] GRIFFITH, B. J., AND BOYLAND, D. E. Reynolds and Mach number simulations of Apollo and Gemini re-entry and comparisons with flight. In *NATO Advisory Group for Aerospace Research and Development*. NATO, Paris, 1968, pp. 8–1–8–21.
- [33] GROPP, W. D., KAUSHIK, D. K., KEYES, D. E., AND SMITH, B. F. High performance parallel implicit CFD. *Parallel Computing* 27, 4 (2001), 337–362.
- [34] GUPTA, R. N., YOS, J. M., THOMPSON, R. A., AND LEE, K. P. A review of reaction rates and thermodynamic and transport properties for an 11-species air model for chemical and thermal nonequilibrium calculations to 30,000 k. NASA-RP-1232, 1990.
- [35] HARTEN, A., ENGQUIST, B., OSHER, S., AND CHAKRAVARTHY, S. Uniformly high order essentially non-oscillatory schemes, III. *Journal of Computational Physics* 71 (1987), 231–303.
- [36] HARTEN, A., LAX, P. D., AND LEER, B. V. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review* 25, 35–61.
- [37] HASH, D., OLEJNICZAK, J., WRIGHT, M., PRABHU, D., PULSONETTI, M., HOLLIS, B., GNOFFO, P., BARNHARDT, M., NOMPELIS, I., AND CANDLER, G. FIRE II calculations for hypersonic nonequilibrium aerothermodynamics code verification: DPLR, LAURA, and US3D. AIAA Paper 2007-0605, 2007.
- [38] HASSAN, B., KUNTZ, D. W., AND POTTER, D. L. Coupled fluid/thermal prediction of ablating hypersonic vehicles. AIAA Paper 1998-0168, 1998.
- [39] HASSAN, O., MORGAN, K., AND PERAIRE, J. An adaptive implicit/explicit finite element scheme for compressible high speed flows. AIAA Paper 1989-0363, 1989.

- [40] HIRSCH, C. *Numerical Computation of Internal and External Flows*. John Wiley & Sons, 1991.
- [41] HOLLIS, B. R. Experimental and computational aerothermodynamics of a Mars entry vehicle. NASA-CR-201633, Dec. 1996.
- [42] HOLLIS, B. R., AND PERKINS, J. N. High-enthalpy aerothermodynamics of a Mars entry vehicle Part 1: Experimental results. *Journal of Spacecraft and Rockets* 34, 4 (1997), 449–456.
- [43] HOLLIS, B. R., AND PERKINS, J. N. High-enthalpy aerothermodynamics of a Mars entry vehicle Part 2: Computational results. *Journal of Spacecraft and Rockets* 34, 4 (1997), 457–463.
- [44] JAWAHAR, P., AND KAMATH, H. A high-resolution procedure for Euler and Navier-Stokes computations on unstructured grids. *Journal of Computational Physics* 164 (2000), 165–203.
- [45] JOSYULA, E., AND BAILEY, W. Governing equations for weakly ionized plasma flowfields of aerospace vehicles. *Journal of Spacecraft and Rockets* 40, 6 (2003), 845–857.
- [46] KARYPIS, G., AND KUMAR, V. *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*. University of Minnesota, MN, 1998.
- [47] KIM, M. D., BHUTTA, B. A., AND LEWIS, C. H. Three-dimensional effects upon real gas flows past the Space Shuttle. AIAA Paper 1984-0225, 1984.
- [48] KIM, S., MAKAROV, B., AND CARAENI, D. A multi-dimensional linear reconstruction scheme for arbitrary unstructured grids. AIAA Paper 2003-3990, 2003.
- [49] LEE, J. H. Basic governing equations for the flight regimes of aeroassisted orbital transfer vehicles. In *Thermal Design of Aeroassisted Orbital Transfer Vehicles*, vol. 96. AIAA, New York, 1985, pp. 3–53.
- [50] LEER, B. V. Flux vector splitting for the Euler equations. *Lecture Notes in Physics* 170, 507–512.
- [51] LEER, B. V. Towards the ultimate conservative scheme IV: A new approach to numerical convection. *Journal of Computational Physics* 23, 276–299.
- [52] LEER, B. V. Towards the ultimate conservative scheme V: A second order sequel to godunov's method. *Journal of Computational Physics* 32, 101–136.
- [53] LEER, B. V., AND MULDER, W. A. Relaxation methods for hyperbolic conservation laws. In *Numerical Methods for the Euler Equations of Fluid Dynamics*, vol. 21 of *Proceedings in Applied Mathematics*. SIAM, Philadelphia, 1985, pp. 312–333.

- [54] LEVY, D. W., POWELL, K. G., AND LEER, B. V. Use of a rotated Riemann solver for the two-dimensional Euler equations. *Journal of Computational Physics* 106, 201–214.
- [55] LI, C. P. Computation of three-dimensional flow about aerobrake configurations. AIAA Paper 1986-0566, 1986.
- [56] LIOU, M.-S. Ten years in the making – AUSM-family. NASA TM-2001-210977, 2001.
- [57] LIU, X. D., OSHER, S., AND CHAN, T. Weighted essentially nonoscillatory schemes. *Journal of Computational Physics* 115 (1994), 200–212.
- [58] LOFTHOUSE, A., SCALABRIN, L., AND BOYD, I. Hypersonic aerothermodynamics analysis across nonequilibrium regimes using continuum and particle methods. AIAA Paper 2007-3903, 2007.
- [59] LOFTHOUSE, A., SCALABRIN, L., AND BOYD, I. Velocity slip and temperature jump in hypersonic aerothermodynamics. AIAA Paper 2007-0208, 2007.
- [60] MACCORMACK, R. W., AND CANDLER, G. V. The solution of the Navier-Stokes equations using Gauss-Seidel line relaxation. *Computers and Fluids* 17 (1989), 135–150.
- [61] MARTIN, D., AND LOHNER, R. An implicit linelet-based solver for incompressible flows. AIAA Paper 1992-0668, 1992.
- [62] MILLIKAN, R. C., AND WHITE, D. R. Systematics of vibrational relaxation. *J. of Chem. Phys.* 39 (1963), 3209–3213.
- [63] MITCHELTREE, R. A., AND GNOFFO, P. A. Wake flow about a MESUR Mars entry vehicle. AIAA Paper 1994- 1958, 1994.
- [64] MOSS, J., AND SIMMONDS, A. Galileo probe forebody flowfield predictions during Jupiter entry. AIAA Paper 1982-0874, 1982.
- [65] NOMPELIS, I. *Computational Study of Hypersonic Double-Cone Experiments for Code Validation*. PhD thesis, University of Minnesota, Minnesota, May 2004.
- [66] NOMPELIS, I., DRAYNA, T. W., AND CANDLER, G. V. Development of a hybrid unstructured implicit solver for the simulation of reacting flows over complex geometries. AIAA Paper 2004-2227, 2004.
- [67] NOMPELIS, I., DRAYNA, T. W., AND CANDLER, G. V. A parallel unstructured implicit solver for reacting flow simulation. AIAA Paper 2005-4867, 2005.
- [68] OLYNICK, D. R. Aerothermodynamics of the Stardust sample return capsule. AIAA Paper 1998-0167, 1998.

- [69] OLYNICK, D. R., CHEN, Y. K., AND TAUBER, M. Forebody TPS sizing with radiation and ablation for the Stardust sample return capsule. AIAA Paper 1997-2474, 1997.
- [70] OLYNICK, D. R., CHEN, Y. K., AND TAUBER, M. Wake flow calculations with radiation and ablation for the Stardust sample return capsule. AIAA Paper 1997-2477, 1997.
- [71] OLYNICK, D. R., HENLINE, W. D., CHAMBERS, L. H., AND CANDLER, G. V. Comparison of coupled radiative flow solutions with project FIRE II flight data. *Journal of Thermophysics and Heat Transfer* 9 (1995), 586–594.
- [72] PALMER, G., AND WRIGHT, M. Comparisons of methods to compute high-temperature gas viscosity. *Journal of Thermophysics and Heat Transfer* 17, 2 (2003), 232–239.
- [73] PARK, C. On convergence of computation of chemically reacting flows. AIAA Paper 85-0247, 1985.
- [74] PARK, C. *Nonequilibrium Hypersonic Aerothermodynamics*. John Wiley & Sons, 1990.
- [75] PARK, C. Chemical-kinetic parameters of hyperbolic Earth entry. *Journal of Thermophysics and Heat Transfer* 15 (2001), 76–90.
- [76] PARK, C., HOWE, J., JAFFE, R., AND CANDLER, G. Review of chemical-kinetic problems of future NASA missions, II: Mars entries. *Journal of Thermophysics and Heat Transfer* 8 (1994), 9–23.
- [77] PORWITZKY, A., SCALABRIN, L., KEIDAR, M., AND BOYD, I. Chemically reacting plasma jet expansion simulation for application to electrothermal chemical guns. AIAA Paper 2007-4600, 2007.
- [78] ROE, P. L. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics* 43, 357–372.
- [79] ROE, P. L. The use of the Riemann problem in finite difference schemes. *Lecture Notes in Physics* 141, 354–359.
- [80] RUSANOV, V. V. Calculation of interaction of non-steady shock waves with obstacles. *J. Compt. Math. Phys. USSR* 1, 267–279.
- [81] SCALABRIN, L. C., AND BOYD, I. D. Numerical simulations of the FIRE-II convective and radiative heating rates. AIAA Paper 2007-4044, 2007.
- [82] SCHWARTZENTRUBER, T., SCALABRIN, L., AND BOYD, I. Hybrid particle-continuum simulations of non-equilibrium hypersonic blunt body flow fields. AIAA Paper 2006-3602, 2006.

- [83] SCHWARTZENTRUBER, T., SCALABRIN, L., AND BOYD, I. Hybrid particle-continuum simulations of low Knudsen number hypersonic flows. AIAA Paper 2007-3892, 2007.
- [84] SCHWARTZENTRUBER, T., SCALABRIN, L., AND BOYD, I. Modular implementation of a hybrid DSMC-NS algorithm for hypersonic non-equilibrium flows. AIAA Paper 2007-0613, 2007.
- [85] SHU, C. W., AND OSHER, S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics* 77, 2 (1988), 439–471.
- [86] STEGER, J. L., AND WARMING, R. F. Flux vector splitting for the inviscid gasdynamic equations with applications to finite difference methods. *Journal of Computational Physics*. 40 (1981), 263–293.
- [87] SUTTON, K., AND GNOFFO, P. A. Multi-component diffusion with application to computational aerothermodynamics. AIAA Paper 1998-2575, 1998.
- [88] VENKATAKRISHNAN, V. Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *Journal of Computational Physics* 118 (1995), 120–130.
- [89] VENKATAKRISHNAN, V. Implicit schemes and parallel computing in unstructured grid CFD. VKI-LS-1995-02, VKI Lecture Series, 1995.
- [90] VICTORIA, K. J., AND WIDHOPF, G. F. Numerical solution of the unsteady Navier-Stokes equations in curvilinear coordinates: The hypersonic blunt body merged layer problem. In *Proceedings of the 3rd International Conference on Numerical Methods in Fluid Dynamics*. Paris, France, 1972, pp. 179–188.
- [91] VINCENTI, W. G., AND KRUGER, C. H. *Introduction to Physical Gas Dynamics*. Krieger Publishing Company, 1982.
- [92] WHALEY, R. C., PETITET, A., AND DONGARRA, J. J. Automated empirical optimization of software and the ATLAS project. *Parallel Computing* 27 (2001), 3–35.
- [93] WHITING, E. E., PARK, C., LIU, Y., ARNOLDAND, J. O., AND PATERSON, J. A. NEQAIR96 nonequilibrium and equilibrium radiative transport and spectra program: User's manual. NASA-RP-1389, 1996.
- [94] WILKE, C. R. A viscosity equation for gas mixtures. *J. of Chem. Phys.* 18 (1950), 517–519.
- [95] WRIGHT, M. J. *A Family of Data-Parallel Relaxation Methods for the Navier-Stokes Equations*. PhD thesis, University of Minnesota, Minnesota, June 1997.

- [96] WRIGHT, M. J., BOSE, D., PALMER, G. E., AND LEVIN, E. Recommended collision integrals for transport property computations, Part 1: Air species. *AIAA Journal* 43, 12 (2005), 2558–2564.