AN EDGE-BASED GALERKIN FORMULATION FOR THERMO-CHEMICAL NON-EQUILIBRIUM FLOWS

Song Gao

Department of Mechanical Engineering
McGill University, Montreal
November 2018

A thesis submitted to McGill University
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Acknowledgements

I would like to express my sincerest gratitude to my supervisor Prof. Wagdi G. Habashi, for his expertise, support, and guidance throughout my studies. I would also like to thank my co-supervisor Prof. Marco Fossati, for many discussions. I would like to thank Drs. Dario Isola and Guido Baruzzi from ANSYS Canada for extensive technical discussions and Dario for code development/debugging and editorial assistance. Thanks to my colleagues Jory Seguin, Wenbo Zhang, and Marie-Ève Dumas, with whom I discussed many technical problems. I would like to thank my parents Haiyun Song and Xi Gao, who raised and educated me. This project was financially supported by the NSERC-Bell Helicopter-Lockheed Martin Industrial Research Chair at the McGill CFD Lab, and through Lockheed Martin grant G238199 to ANSYS. Compute Canada and CLUMEQ have given access to their computational resources that facilitated this work.

Table of Contents

List of Figures

List of Tables

List of Symbols

English letters

$\tilde{A}$	Roe matrix of flow solver
A	Jacobian for JFNK
$\hat{A}$	Linearized Jacobian of inviscid flux for flow solver
$A_{ij}^{V,S}$	Symmetric part of weak form of stress tensor
$A_{ij}^{V,A}$	Anti-symmetric part of weak form of stress tensor
A_r^f	Coefficient of modified Arrhenius equation
$C_{p,s}^{tr}$	Species translational-rotational specific heat at constant pressure
C_p	Mixture specific heat at constant pressure
$C_{v,s}^{r}$	Species rotational specific heat at constant volume
$C_{v,s}^{t}$	Species translational specific heat at constant volume
$C_{v,s}^{tr}$	Species translational-rotational specific heat at constant volume
C_v^{ve}	Mixture vibrational-electronic specific heat at constant volume
c_s	Species average molecular speed
D	Diagonal part of matrix A
D_{ii}^{A}	Inviscid part of D
D_{ii}^{V}	Viscous part of D
$\mathrm{D}_{ii}^{\partial}$	Boundary part of D
$D_{ii}^{A+\partial}$	Inviscid and boundary part of D
D	Mass diffusion coefficient
$\widetilde{D}_s$	Species mass diffusion coefficient
D_s	Species dissociation potential
D_s'	Species vibrational energy production
$\mathbf{d}_{ij}$	Edge coefficients
$\mathbf{d}_{ij}^{A}$	Anti-symmetric part of $\mathbf{d}_{ij}$
$\mathbf{d}_{ij}^{S}$	Symmetric part of $\mathbf{d}_{ij}$
$E_{a,r}$	Activation energy for reaction r

E_i	Set of elements sharing the i-th vertex
$\boldsymbol{e}$	Total energy per unit mass
$e_{e,s}$	Species electronic energy per unit mass
$e_{int,s}$	Species internal energy per unit mass
$e_{v,s}$	Species vibrational energy per unit mass
$e_{ve,s}$	Species vibrational-electronic energy per unit mass
e_{ve}	Total vibrational-electronic energy per unit mass
$error\_rel$	Square root of the machine precision
F_i	Set of boundary faces sharing the i-th vertex
$\mathbf{F}^{A,\partial}$	Inviscid flux vector of boundary nodes
$\mathbf{F}^{A}$	Inviscid flux vector
$F_{\boldsymbol{x}}^{A}$	Inviscid flux in x direction
$F_{\boldsymbol{y}}^{A}$	Inviscid flux in y direction
$F_{\boldsymbol{z}}^{A}$	Inviscid flux in z direction
$\tilde{F}_{\boldsymbol{x}}^{A}$	Inviscid flux of momentum equation in x direction
$\mathbf{F}^{V}$	Viscous flux vector
$F_{\boldsymbol{x}}^{V}$	Viscous flux vector in x direction
$F_{\boldsymbol{y}}^{V}$	Viscous flux vector in y direction
$F_{\boldsymbol{z}}^{V}$	Viscous flux vector in z direction
$\mathbf{F}_{ij}^{V}$	Viscous flux between node i and node j
$\mathbf{F}_{ij}^{F,M}$	Viscous flux for momentum equations in flow solver between node i and j
$\mathbf{F}_{ij}^{F,E}$	Viscous flux for energy equation in flow solver between node i and j
$\mathbf{F}_{ij}^{F,H}$	Heat flux for energy equation in flow solver between node i and j
$\mathbf{F}_{s,ij}^{C,D}$	Inter-diffusional mass flux for species s in chemistry solver between node i and j
$\mathbf{F}_{ij}^{F,D}$	Inter-diffusional heat flux for energy equation in flow solver between node i and j
$\mathbf{F}_{ij}^{VE,D}$	Inter-diffusional heat flux for vibrational-electronic energy equation in thermal non-equilibrium solver between node i and j

ΔG_r	Gibb's function
$h_{0,s}$	Species heat of formation
h_0	Total heat of formation
h	Total enthalpy per unit mass
$\tilde{h}_{int,s}$	NIST polynomial representation of enthalpy
$\mathbf{I}$	Identity tensor
$\mathbf{I}^{(c)}$	$(N_s + 5) \times 1$ matrix that is nil everywhere except at the position of species s continuity equation
$\mathbf{I}^{(e)}$	$(N_s + 5) \times 1$ matrix that is nil everywhere except at the position of energy equation
$\mathbf{I}^{(\boldsymbol{eve})}$	$(N_s + 5) \times 1$ matrix that is nil everywhere except at the position of vibrational-electronic energy equation
$\mathbf{I}^{(\boldsymbol{m})}$	$(N_s + 5) \times 3$ matrix that is nil everywhere except for an identity matrix at the position of momentum equations
$\boldsymbol{J}_s$	Species mass diffusion flux
$J_{i,s}$	Species mass diffusion flux in i direction
K_e	Set of nodes of the e-th element
K_i	Set of nodes connected to i-th vertex
K_r^{eq}	Equilibrium constant of reaction r
L	Lower parts of matrix $\boldsymbol{A}$
$\hat{L}$	Left eigenvectors of $\hat{A}$
L_{ij}^{A}	Inviscid part of L
L_{ij}^{V}	Viscous part of L
L	Lumped mass matrix
$\tilde{L}$	Left eigenvalue matrix of $\tilde{A}$
L_i	Lumped mass matrix for i-th vertex
M	Preconditioner of JFNK
$\boldsymbol{M}$	Absolute value of Hessian of solution
M_s	Species molecular weight

N_G	Total number of nodes in the grid
N	Mixture number density
N_i	Linear Lagrangian shape function
N_s	Number of species
N_r	Number of chemical reactions
n_{atoms}	Number of atoms
ΔP	Update of preconditioned conservative variables vector
p	Total pressure
p_s	Species partial pressure
Q	Conservative variables vector
ΔQ	Update of conservative variables vector
$\Delta \boldsymbol{Q}^*$	Intermediate update of conservative variables vector
Q_∞	Conservative variables vector at far field
$\boldsymbol{q}_{tr}$	Translation-rotational heat flux
$\boldsymbol{q}_{ve}$	Vibrational-electronic heat flux
$q_{tr,i}$	Translation-rotational heat fluxes in i direction
$q_{ve,i}$	Vibrational-electronic heat fluxes in i direction
$\tilde{R}$	Right eigenvalue matrix of $\tilde{A}$
R	RHS of linearized system of JFNK
$\hat{R}$	Right eigenvector of $\hat{A}$
R^A	Inviscid part of R
R^V	Viscous part of R
R^∂	Boundary part of R
R_r^b	Backward reaction rates of reaction r
R_r^f	Forward reaction rates of reaction r
R_s	Species gas constant
R_u	Universal gas constant
S	Source term vector
S_{c-v}	Coupling between vibrational-electronic energy and chemical reactions
S_{t-v}	Energy relaxation between translational and vibrational modes

$\bar{T}_c$	Corrected rate-controlling temperature
T_{tr}	Translation-rotational temperature
T_{ve}	Vibrational-electronic temperature
U	Upper parts of matrix $\boldsymbol{A}$
U_{ij}^{A}	Inviscid part of U
U_{ij}^{V}	Viscous part of U
u	Velocity component in x direction
$\tilde{V}_n$	Roe average velocity projected at $\boldsymbol{\eta}_{ij}$ direction
V^e	Volume of e-th element
V	Whole volume domain
$\boldsymbol{V}$	Velocity
V_n	Velocity along $\mathbf{n}$ direction
v	Velocity component in y direction
W_i	Linear Lagrangian test function
w	Velocity component in z direction
X_s	Species molar fraction
x_g	Coordinates of Gauss points
$\boldsymbol{x}_i$	Coordinates of node i
Y_s	Mass fraction of species s

Greek letters

γ	Gas heat capacity ratio
ε	Perturbation used in Fréchet derivative
$\boldsymbol{\eta}_{ij}$	Edge coefficients
η_r^f	Coefficient of modified Arrhenius equation
$\theta_{el,i,s}$	Species characteristic electronic temperature at level i
$\theta_{v,s}$	Species characteristic vibrational temperature
κ_s	Species heat conductivity
$\kappa_{tr,s}$	Species translation-rotational heat conductivity
$\kappa_{ve,s}$	Species vibrational-electronic heat conductivity

k_{tr}	Translation-rotational heat conductivity
k_{ve}	Vibrational-electronic heat conductivity
κ	Mixture conductivity
$\tilde{\Lambda}$	Eigenvalues matrix of $\tilde{A}$
$\hat{\Lambda}$	Eigenvalues matrix of $\hat{A}$
Λ_{ij}	Spectral radius of numerical flux of JFNK
μ_s	Species viscosity
μ	Mixture viscosity
$\nu'_{s,r}$	Reactant stoichiometric coefficients of species s and reaction r
$\nu''_{s,r}$	Product stoichiometric coefficients of species s and reaction r
$\boldsymbol{\xi}_i$	Nodal boundary coefficient
ρ	Mixture density
ρ_s	Species density
$\sum_{mol}$	Sum over all molecules
σ_s	Species limiting cross-section
τ_{ij}	Component of viscous stress tensor
$\tau_{p,s}$	Species Park's correction relaxation time of translational and vibrational energy exchange
$\tau_{s,i}$	Inter-species relaxation time of translational and vibrational energy exchange between species i and species s
τ_s	Species relaxation time of translational and vibrational energy exchange
$< \tau_s >$	Species Landau-Teller relaxation time of translational and vibrational energy exchange
τ	Relaxation time of translational and vibrational energy exchange
$\boldsymbol{\tau}$	Viscous stress tensor
$\Delta\tau^t$	Pseudo-time step
$\boldsymbol{\Phi}_{ij}^{\text{num,flow,density}}$	Numerical flux of density at edge ij midpoint
$\boldsymbol{\Phi}_{ij}^{\text{num,flow}}$	Numerical flux of flow solver at edge ij midpoint

$\mathbf{\Phi}_{ij}^{\text{num,thermal}}$	Numerical flux of thermal non-equilibrium solver at edge ij midpoint
$\mathbf{\Phi}^{\text{num}}$	Numerical flux at edge ij midpoint
$\mathbf{\Phi}_{s,ij}^{\text{num,chemistry}}$	Numerical flux of chemistry solver at edge ij midpoint for species s
ϕ_s	Scaling factor of computing mixture heat conductivity
$\boldsymbol{\chi}_{ij}$	Edge coefficients
$\dot{\omega}_s$	Species mass production or destruction due to reactions

Non-Dimensional Numbers

Re	Reynolds number
Kn	Knudsen number
Mach	Mach number
CFL	Courant-Friedrichs-Lewy number
Le	Lewis number

Parameters

$R_u = 8.3144621$	Universal Gas Constant [J/mol.K]
$N_A = 6.02 \cdot 10^{26}$	Avogadro's Number [1/mol]
$k \ = 1.38064852 \cdot 10^{-23}$	Boltzmann’s Constant [J/K]

Abbreviations

CFD	Computational Fluid Dynamics
COOL Fluid	Computational Object-Oriented Library for Fluid Dynamics
DPLR	Data-Parallel Line Relaxation
FEM	Finite Element Method
FGMRES	Flexible Generalized Minimal Residual Method
FIN-S	Fully-Implicit Navier-Stokes
FVM	Finite Volume Method
FAST20XX	Future High-Altitude High Speed Transport 20XX
GPS	Global Positioning System
HALO3D	High Altitude Low Orbit 3D Code
ILU	Incomplete LU Factorization
JFNK	Jacobian-Free Newton-Krylov

LAURA	Langley Aerothermodynamic Upwind Relaxation Algorithm
LeMANS	Michigan Aerothermodynamic Navier-Stokes Solver
LU-SGS	Lower-Upper Symmetric Gauss-Seidel
MUSCL	Monotonic Upstream-centered Scheme for Conservation Laws
PETSc	Portable, Extensible Toolkit for Scientific Computation
LHS	Left-hand side
RHS	Right-hand side
SUPG	Streamline Upwind Petrov-Galerkin
T-R	Translational-rotational
V-E	Vibrational-electronic

Abstract

The present work extends the capabilities of a compressible Navier-Stokes solver into a thermo-chemical non-equilibrium hypersonic flow solver. Finite-rate chemistry and two-temperature thermal non-equilibrium solvers are implemented to account for the additional non-equilibrium processes. The spatial discretization uses an edge-based Finite Element formulation with flow stabilization achieved using a Roe scheme. The governing equations are solved numerically on both structured and unstructured grids. The steady-state solution is obtained by using an implicit integration in time. The present code is comprised of flow, chemistry, and thermal non-equilibrium solvers, developed primarily by Dario Isola, Jory Seguin, and the author, respectively. A loosely-coupled strategy is used in which each of the systems is solved separately via a generalized minimal residual (GMRES) method with an incomplete LU factorization (ILU) preconditioner provided by the PETSc library. Numerical experiments consisting of flows past blunt cones, cylinders, and spheres are performed to assess the accuracy and efficiency of the approach, and good agreement is found with solutions available in the literature. It is observed that mesh distributions are crucial for simulations on unstructured meshes, and anisotropic mesh optimization is successfully applied.

A Jacobian-free Newton–Krylov (JFNK) solver with a lower-upper symmetric Gauss-Seidel (LU-SGS) preconditioner is developed for thermal equilibrium flows with frozen chemistry. The traditional LU-SGS formulation is enriched by including the contributions from viscous fluxes and boundary conditions. The performance of the JFNK solver is subsequently assessed and the enriched LU-SGS is found to be more robust and efficient than the Jacobi preconditioner and the original LU-SGS. Comparisons between JFNK with LU-SGS and GMRES with ILU are carried out and the results show that the present method, despite requiring more computation time, significantly reduces the memory footprint by half.

Résumé

Le présent travail étend les capacités d'un solveur Navier-Stokes compressible à un solveur hypersonique thermochimique hors-équilibre. Un solveur chimique à taux fini et un solveur thermique hors-équilibre à deux températures sont mis en œuvre pour tenir compte des processus supplémentaires hors-équilibre. La discrétisation spatiale est réalisée par la méthode des éléments finis avec un flux de stabilisation obtenu à l'aide du schéma de Roe. Les équations sont résolues numériquement sur des maillages structurés et non structurés. La solution à l'état d'équilibre est obtenue par une intégration temporelle implicite. Le présent logiciel comprend un solveur Navier-Stokes, un solveur chimique et un solveur thermique hors-équilibre, développés principalement et respectivement par Dario Isola, Jory Seguin et l'auteur. Une stratégie faiblement couplée est utilisée dans laquelle chacun des systèmes est résolu séparément via la généralisation de la méthode de minimisation du résidu (GMRES) avec un préconditionneur par factorisation incomplète (ILU) fourni par la bibliothèque PETSc. Une série d'expériences numériques sont effectuées pour évaluer la précision et l'efficacité de la méthode, y compris des écoulements autour d'un cône émoussé, d'un cylindre et d'une sphère. Une bonne concordance avec les solutions disponibles dans la littérature est obtenue. On constate que la distribution de mailles est cruciale pour les simulations sur des maillages non structurés et l'optimisation anisotrope du maillage est appliquée avec succès.

Un solveur du type Newton – Krylov sans Jacobien (JFNK) avec préconditionneur Gauss-Seidel (LU-SGS) est développé pour les écoulements à l'équilibre thermique sous une condition chimique fixée. La formulation LU-SGS traditionnelle est enrichie par les contributions des flux visqueux et des conditions aux limites. Les performances du solveur JFNK sont ensuite évaluées et la formulation LU-SGS modifiée se révèle plus robuste et efficace que le préconditionneur de Jacobi et la LU-SGS traditionnelle. Des comparaisons entre le JFNK avec la LU-SGS et le GMRES avec l'ILU sont effectuées et les résultats montrent que malgré une légère augmentation de temps de calcul, la méthode actuelle réduit considérablement la mémoire utilisée avec une baisse allant jusqu'à la moitié.

1 Introduction

1.1 Hypersonic Technology Applications

A long-term interest of the aerospace industry is moving towards designing vehicles that fly beyond the speed of sound, up to hypersonic speeds [1]. In operation from 1976 to 2003, the Concorde was the world's first commercial supersonic airliner to fly at a cruise speed of Mach 2.02, more than twice the speed of other subsonic civil airplanes. It flew regular transatlantic routes at a cruise altitude of 18,300 meters, about 6,000 meters higher than today's civil aircraft, with less than 3.5 flight hours from New York to Paris. The Concorde was retired in 2003, three years after the tragic accident of Air France Flight 4590 over Gonesse. The Tupolev Tu-144 was another commercial supersonic aircraft developed in Russia that started passenger service in 1977, ended in 1978, due to safety concerns. Later, it was used for cargo and as a supersonic research laboratory. In 1961, the US started its own supersonic transport project. The Boeing 2707 won the competition against the Lockheed L-2000 for a government-funded contract intended to deliver a much larger and faster airplane than the Concorde. Unfortunately, this project was cancelled in 1971, due to rising costs and lack of a clear market. Today, the development of next-generation civil supersonic aircraft is gaining more attention with Aerion announcing in December 2017 a collaboration with Lockheed Martin to design a new supersonic business jet operating at Mach 1.5 [2]. Boom Technology is another company working with Virgin to develop a Mach 2.2 commercial plane [3].

These projected commercial airliners will fly at supersonic speeds, defined as Mach numbers between 1 and 5. In this flow regime, the fluid is not chemically reacting and the internal energy modes of the air are in equilibrium, allowing the definition of a single temperature. As the Mach number increases beyond 5, the flow around the airliner is classified as hypersonic. Some civil transports have been designed in this flow region. cFASTT-1 is a concept reusable air-space vehicle designed at the University of Strathclyde that flies at Mach 22.7 at a 114 km altitude [4]. FAST20XX (Future High-Altitude High Speed Transport 20XX) is a European Space Agency program that focuses on the development of new technologies for future hypersonic suborbital flights [5]. Two concept vehicles have been generated in this program. The first, ALPHA is a vehicle that is drop-released from a carrier plane. Powered by rocket engines, it can fly out of the atmosphere up to 100 km altitude, and then glide back to Earth. A second vehicle, the Spaceliner, is designed to take 50 passengers and fly from Europe to Australia in 90 minutes, with a speed up to 7 km/s. In terms of engines, Sabre, under development in the UK, is a hybrid hypersonic engine that is designed to achieve single-stage-to-orbit capability [6]. Sabre can use an air-breathing mode to accelerate from still to Mach 5.5, after which a rocket mode can be activated to reach a speed of Mach 25. The pre-cooling system of Sabre will be tested in the US in 2019.

The development of commercial hypersonic airliners is still in the early stage, but hypersonic rockets already send cargoes, and soon passengers, to outer space [7] making space exploration been possible in the last few decades. Many important instruments have been launched into space such as the Hubble, one of the largest space telescopes, launched in 1990 and is still in operation [8]. It has helped study many fundamental problems in astronomy, such as the age of the Universe, formation of planets, and existence of dark energy. The International Space Station is a project that shows intergovernmental cooperation and peaceful uses of outer space [9]. Launched in 1998, it is currently the largest artificial body in low Earth orbit, the fruit of a collaboration between US, Russia, Europe, Japan, and Canada. Canada developed the Space Station Remote Manipulator System, known as Canadarm2, which is used to assist both docking and undocking the spacecraft [10]. Research in medicine, combustion, biology, education, superconductivity, and robotics have been conducted on the International Space Station. Nowadays, thousands of satellites have been launched into orbit. Most of them provide services, such as communication, weather forecast, and navigation. Perhaps the most widely-used application is the Global Positioning System (GPS) system [11]. Billions of GPS receivers have been installed in phones and cars, providing real-time location to the user. Moving into the 21st century, private companies, such as SpaceX, have joined space exploration [12]. SpaceX's remarkable achievements include the world's first reusable orbital rocket Falcon 9 in 2017, and the world's first partially reusable heavy-lift rocket Falcon Heavy in 2018. SpaceX plans to launch its first commercial tourist mission to the Moon in 2023.

Space debris is another interesting topic involving hypersonic flows [13]. In 2006, more than 9,000 man-made earth-orbiting objects were tracked. Due to the influence of gravity and solar activities some of them will gradually return to Earth at a high speed. Although most space debris are burned during re-entry, some large ones can reach the ground, which causes potential dangers to people. Studying the flow around debris will help determine whether they will hit the ground and, if so, where the impact area may be.

The above-mentioned applications typically involve complex physical phenomena due to high kinetic energy. To fully understand the hypersonic flow regime, it is necessary to perform a comprehensive study of the physics behind it.

1.2 Physical Modeling

Several outstanding challenges remain in hypersonic flows due to their complex physics. These flows have a very high energy content that is converted from kinetic to thermal energy through shock waves. The temperature increase across shock waves can be so high that the internal energy modes associated with molecular vibration and electronic excitation are activated [14]. At such elevated temperatures (8,000K) the gas is no longer inert and chemical reactions such as dissociation and ionization are initiated [15]. When the characteristic flow timescale becomes much shorter than the timescale required to complete energy exchanges between different modes or to complete the chemical reactions, thermo-chemical non-equilibrium exists [16]. Other physical phenomena such as radiation [17], couplings between flow and magnetic fields [18], catalytic effects and erosion of the ablative heat shield material [19] can also be present, and thus a multidisciplinary approach is necessary. Due to the complexity of the hypersonic regime, experimental campaigns are expensive, not always feasible or limited in the physics that they can accurately reproduce [20]. On the numerical side, despite being a long-investigated problem, computational hypersonic fluid dynamics to this day poses unique engineering challenges and therefore is still a very active field of research. Significant effort is currently invested in the development of numerical methods with advanced modelling capabilities, accurate solutions and the ability to mitigate the computational costs of the complex aerothermodynamics of high-Mach flows [21-25].

A molecular description of the thermodynamic behaviour of a gas allows for a complete and fundamental understanding of its macroscopic dynamics. Under the perspective of statistical thermodynamics, energy can be stored in each molecule in four different forms, referred to as energy modes. These are the translational, rotational, vibrational, and electronic modes and they respectively account for the kinetic and the angular kinetic energy of the molecules, the energy associated with the vibration of the atoms of a molecule and the energy associated with electrons orbiting around nuclei [26]. In the case of monatomic gases, the contributions associated with rotation and vibration are absent and the only two forms of energy storage are the ones associated with the translational kinetic energy and the electronic energy. For a gas at thermal equilibrium, energy is distributed equally among the different excited modes and a single macroscopic quantity, the temperature of the gas, can be used as a measure of its internal energy. However, for hypersonic flows, the energy modes of the gas may not be at equilibrium due to the very high temperature. A single temperature is therefore not adequate to describe the internal energy of the gas since each energy mode independently follows a Boltzmann distribution, possibly at different temperatures [16].

Candler [27] solves a set of equations that uses a multi-temperature model, including a translational rotational temperature T_{tr}, different vibrational temperatures $T_{v,s}$ for each diatomic species and an electronic and electron temperature T_e. Candler's findings demonstrate that for typical hypersonic flow conditions the vibrational temperatures of each molecule are similar, indicating that a common vibrational temperature is sufficient.

Lee [28] proposes a three-temperature model which first assumes that the translational and rotational energy of all heavy particles can be modelled by a single temperature T_{tr} since the translational energy equilibrates with the rotational one quickly. The second assumption is that the vibrational energy of all heavy species can be modeled by a single temperature T_v since the vibrational-vibrational energy transfer between molecules is sufficiently fast to ensure that the vibrational energy modes are in equilibrium. The final assumption is that the electronic energy of all species and the electron translational energy can be described by a single temperature T_e [29]. The presence of multiple temperatures results in a modification of the governing equations such that the conservation equations of total energy, vibrational energy, and electron energy should be addressed explicitly. The energy exchanges between different modes appear as source terms with expressions given in [28, 30, 31].

Park [32] makes additional simplifications to Lee's framework and proposes a two-temperature model which further assumes that the vibrational temperature T_v and electron temperature T_e are at equilibrium, denoted by T_{ve}. Park justified this assumption by stating that the strongest process affecting electron energy is the inelastic collision between electrons and molecules, which has the tendency to bring the vibrational temperature and electron temperature to equilibrium [33]. The two-temperature model simplifies the mathematical and computational complexity of the problem, and provides fairly accurate results for aerodynamic coefficients and convective heat transfer rates in the case of reentry problems [34]. Nevertheless, when it comes to address the problem of interaction between charged particles and free electrons, the two-temperature model may not be sufficient. Therefore, in cases where phenomena like radio blackout and/or attempts to control the flow via magnetic fields are the goal of the analysis, the three-temperature model is often preferred [31].

Thermal non-equilibrium and the resulting necessity of dealing with multiple temperatures have a strong impact on the thermochemical modelling of the gas. The three important aspects that must be addressed to accurately model hypersonic flows are the coupling between thermal non-equilibrium and chemical reaction rates, the effects of temperatures and gas compositions on transport properties, and the thermodynamic relations. The models proposed in the literature to address these three aspects are discussed in the following paragraphs.

To account for chemical non-equilibrium [16, 35], additional conservation equations for the mass of the species that define the gas are considered. The effects of chemical production and destruction are modelled via a finite-rate chemical kinetics model [15] that appears as a source term in the conservation equations of the mass of each species. Typically, the rate of a reaction is primarily driven by the temperature and as such higher temperatures lead to faster reactions. Thermal non-equilibrium strongly influences these chemical processes since the reaction rates are influenced differently by the translational and vibrational energy. The chemical reactions that typically occur in hypersonic flows are dissociation, neutral exchange, associative ionization, charge exchange and electron impact ionization reactions. Different chemical reactions are controlled by different temperatures in thermal non-equilibrium conditions. The neutral exchange, charge exchange and associative ionization reactions are controlled by the translational-rotational temperature T_{tr} while the electron impact ionization reactions are controlled by the electron temperature, T_{ve} (two-temperature model) or T_e (three-temperature model) [32]. The dissociation reactions are calculated from the control temperature T_c, defined as $T_{tr}^a T_{ve}^b$ (two-temperature model) or $T_{tr}^a T_v^b T_e^c$ (three-temperature model), where the coefficients are empirically proposed in [31, 32, 36]. Aside from the rate-controlling temperature in the Arrhenius law, the frequency factor and activation energy of the reaction are parameters that are determined from experiments. These detailed chemical reaction data are provided by Park in [37] for the Earth's atmosphere, and in [38] for the Martian atmosphere.

Thermal non-equilibrium also has significant consequences on the transport properties of a gas mixture such as the coefficients of diffusion, viscosity, and thermal conductivity. Typically, there are two ways to model the transport properties [30]. The first method uses Blottner's [39] curve fits of viscosity for each species, Eucken's [40] relation for the thermal conductivities of each energy mode for each species, and finally Wilke's [41] mixing rule to calculate the transport properties for the mixture. As discussed by Scalabrin [30], the first method is not designed for ionized flows and is only adequate for relatively low-temperature flows since the curve fits are only valid below 10,000 K. The second method is suggested by Lee [28] and is an extension of Yos' formula [42], as it uses Gupta's mixing rule [43] along with the up-to-date collisional cross-section data taken from [44]. Gupta's mixing rule is adequate for weakly ionized flows. The collisional cross-section is a function of the rate-controlling temperatures which can be obtained by either theoretical or experimental means.

The thermodynamic relation between temperatures and internal energy can be modelled in the framework of statistical mechanics [34]. Unlike a perfect gas, where energy and temperature are linearly related, the thermal non-equilibrium conditions require the consideration of the internal energy as a function of different temperatures. The internal energy is further divided into four parts associated with the translational, rotational, vibrational, and electronic modes. [45]. The analytical formulations of energies of each mode can be obtained in terms of the corresponding partition functions. In the temperature range above 200K, the translational and rotational modes are fully excited and thus the energies associated with these modes are linearly dependent on the corresponding temperature. The partition function of the vibrational mode is derived under the harmonic oscillator assumption, which is adequate since low vibrational energies can be modelled accurately, and the contribution of higher vibrational energies is negligible. The partition function of the electronic mode is derived by summing up over all the observed electronic energy levels for particles. Both vibrational and electronic modes are not fully excited, and therefore the energies of these two modes are nonlinear functions of the vibrational temperature T_v or electronic temperature T_e, respectively. Parameters for evaluating internal energy modes can be found in [30]. On the other hand, Gnoffo [34] proposes another method to obtain the thermodynamic relation for the two-temperature model. This method uses polynomial functions from the NIST-JANAF tables [46] to compute the internal energy at the vibrational-electronic temperature, followed by the vibrational-electronic energy obtained by subtracting the translational-rotational energy from the internal energy, given that the translational-rotational energy is fully activated. This method is relatively easy to implement, but care should be taken when using the NIST-JANAF tables as they are not designed for thermal non-equilibrium flows. Thus, the vibrational-electronic energy obtained using that method could be non-physical, particularly in the low-temperature range. In the present work, Gnoffo's original polynomial functions are adjusted to ensure that the vibrational-electronic energy remains positive.

The physical models presented above date back to the 1960s and have not changed much since then. More recent advances in the physico-chemical models of a mixture of gases can be found in the areas of quantum field theory [47], Rice-Ramsperger-Kassel-Marcus theory [48] and molecular reaction dynamics [49]. Nevertheless, these advanced models are hardly employed in the context of CFD simulations as they are too complex to implement and require a significant amount of processed information, such as spectroscopic data, which may not be easily or accurately available. It should also be noted that the accuracy of physical models not only depends on the theoretical understanding of physics but also on quantitative data obtained from experiments. Moreover, the results obtained with the present models provide simulations in reasonably good agreement with experimental data and the change towards a more fundamental representation of these thermochemical properties may not be justified from a purely computational point of view.

1.3 Existing CFD Codes

The numerical simulation of hypersonic flow begins with the discretization of the governing equations. The temporal discretization may be either explicit or implicit. Explicit approaches were used in the 1980s due to their moderate computational requirements, but the stiffness resulting from the chemical reaction terms may restrict the permissible time step to an extremely small number, resulting in long simulation times. Nowadays, implicit methods are preferred because they are more stable and, as a result, the allowable time step is much larger than for explicit methods.

The spatial discretization can use either a finite volume method (FVM) or a finite element method (FEM). Most numerical approaches for simulating hypersonic flows in the literature use a FVM approach. Candler [27] was the first to perform two-dimensional simulations of thermo-chemical non-equilibrium flows with ionization. A multi-temperature model is used that includes one translational temperature, one electron temperature, and multiple vibrational temperatures, in which a different vibrational temperature is defined for each diatomic species. The inviscid flux is computed via a flux-splitting scheme developed by MacCormack and Candler, except in regions of large pressure gradients, where the Steger-Warming scheme is used. The system is fully-coupled and solved by an implicit Gauss-Seidel line relaxation method. The Langley Aerothermodynamic Upwind Relaxation Algorithm (LAURA) is another CFD code developed by Gnoffo [34] that implements Lee's three-temperature model. This code is often used for the design of reentry capsules. The Data-Parallel Line Relaxation (DPLR) code is a hypersonic flow solver developed by NASA [50] in which the inviscid fluxes are computed using a modified form of the Steger-Warming flux splitting technique, combined with the monotonic upstream-centered scheme for conservation laws (MUSCL) for third order spatial accuracy. The viscous fluxes are second-order accurate via central differences. To overcome the numerical stiffness arising from chemical source terms, an implicit time stepping method is used. DPLR uses line relaxation methods to obtain improved convergence of the linear system. The idea of line relaxation is to exactly solve the portion of the linear system that consists of a line in the direction normal to the vehicle (where the gradients are strong), while relaxing the solution of the linear system in the other directions. This strategy, while easily done on a structured grid, puts limitations on geometries and grid sizes. An alternative strategy shown in this thesis is the anisotropic mesh optimization on unstructured grids. The Michigan Aerothermodynamic Navier-Stokes Solver (LeMANS) [30] is another hypersonic FVM code designed for unstructured meshes. The calculations of numerical fluxes are similar to those of DPLR and the linear system is solved using a line relaxation method, with an algorithm to find the line direction in unstructured meshes. hy2Foam is an open-source CFD hypersonic flow solver implemented in the OpenFoam framework [51]. This code also uses an implicit time integration and the inviscid flux is computed using a central-upwind interpolation scheme. The code has been verified against LeMANS and has achieved reasonable agreement for many test cases. The Computational Object-Oriented Library for Fluid Dynamics (COOLFluiD) [52] is an object-oriented code developed in the Von Karman Institute, capable of solving all Mach number flows, from incompressible to hypersonic flows. It is highly flexible and extensible due to its kernel-module structure. The kernel defines all the essential functionalities and abstract interfaces to the modules, while the modules implement all the details, such as physical models and numerical algorithms.

Within the FEM community, a fully implicit hypersonic FEM code was proposed by Ait-Ali-Yahia and Habashi [53] for two-dimensional flows. To reduce the computation to a manageable level, this code utilizes a loosely coupled approach that segregates the hypersonic system into gasdynamic, thermo-chemical non-equilibrium sub-systems which are solved sequentially. In addition, an anisotropic mesh optimization technique was introduced to accurately capture shock waves. The Fully-Implicit Navier-Stokes (FIN-S) [54] solver is another FEM hypersonic code. The system is discretized by a Petrov-Galerkin formulation with Streamline Upwind Petrov-Galerkin (SUPG) [55] stabilization. The time stepping method is fully implicit to overcome the stiffness due to chemical reactions. A large sparse linear system is assembled and solved with an incomplete LU factorization (ILU) preconditioner provided by the PETSc [56] library.

Most of the codes [23, 24, 27, 34, 50, 51] introduced above use a fully-coupled strategy where the flow, chemistry, and vibrational energy equation are assembled and solved simultaneously. In this way, a strong consistency with the solution variables is secured and a better convergence rate obtained, but memory requirements increase due to a much larger matrix, with memory storage growing quadratically with the number of equations. However, in a recent paper, Candler et al. [35] proposed a strategy for decoupling the total mass, momentum, and internal energy equations from species mass and vibrational-electronic energy equations. Their test cases showed that the decoupled system has no loss of convergence rate and the memory cost only grows linearly. Typically, decoupled approaches suffer from a deterioration of the convergence rate unless some additional constraints are introduced. Nonetheless, it is still worth starting the development with a decoupled approach due to its simplicity.

Despite the existence of several hypersonic codes, there is still room for improving the numerical approach. The edge-based FEM formulation used in this work blurs the boundaries of FVM and FEM, and combines the advantages of both. The basic framework of the solver uses the Galerkin FEM, resulting in the natural application of Neumann boundary conditions and the accurate computation of surface and viscous fluxes. However, the inviscid fluxes are assembled using an edge-based scheme that permits the application of FVM compressible flux schemes that ensure adequate upwinding, as well as second-order reconstruction via MUSCL. It has been shown that this scheme is also accurate and robust on grids having high aspect ratio elements (up to 14,000 [57]), which is crucial for resolving shocks and boundary layers on unstructured grids.

1.4 Jacobian-Free Newton-Krylov Method

Solving a complex system of conservation equations that models thermo-chemical non-equilibrium demands a computational framework capable of handling stiff problems characterized by vastly different relaxation times. For steady-state problems, the governing equations are often solved implicitly through an exact or inexact Newton-Raphson procedure. At every Newton step, the residual function is linearized, and the update is computed by solving a linear system. Compared to explicit methods, implicit algorithms require three additional operations: the analytical differentiation of the residual, the assembly of the Jacobian matrix, and the solution of a linear system. For high enthalpy flows, numerical problems arise that can potentially weaken the efficiency of implicit strategies. The analytical differentiation of the residual becomes increasingly difficult as the physics are enriched to include high-temperature thermodynamic effects. The second operation may become memory-prohibitive as the number of species increases. An analytical differentiation of the Jacobian is difficult and is often derived with many physical approximations. Although a numerical approximation of the Jacobian can overcome some of the aforementioned problems, it comes at the expense of increased computational time. Lani et al. [52] have developed a solver that numerically computes each column of the Jacobian via forward finite differences and subsequently stores the Jacobian. The resulting scheme is slower than the one obtained using analytic differentiation and still requires significant memory to store the Jacobian. In the numerical solution of high-Mach flows using implicit methods, the memory footprint of the Jacobian can be very large as the number of degrees of freedom per node increases with the number of species, thus making it more important to limit memory requirements. One possible solution is the line-implicit relaxation method used in LeMANS [24]. It adopts a line search method that finds the normal direction of the wall, uses a renumbering algorithm that reorders the elements, and then the off-tridiagonal terms of the linearized system are moved to the right-hand side, whereas the left-hand side tridiagonal system is solved by a point-iterative method without storing the Jacobian. Another possibility is the Jacobian-Free Newton-Krylov (JFNK) methods, where the Jacobian-vector product is approximated by a Fréchet derivative, which circumvents the need to store the full Jacobian, and a Krylov subspace method is used to solve the linear system [58]. In this work, JFNK is chosen as the linear solver to achieve the Jacobian-free property. Implementing a JFNK method from an existing code that uses an explicit Jacobian is straightforward, and the existing Jacobian assembly routine can be used as an adequate preconditioner. Nevertheless, a new matrix-free preconditioner should be subsequently developed.

The effectiveness of Krylov subspace linear system solvers depends on the adoption of a good preconditioner. The widely-used Incomplete Lower-Upper (ILU) factorization not only requires the storage of the system matrix but also its factorization, thus doubling the memory footprint. To the best of the author's knowledge, three alternate matrix-free options are available. The first one is the Jacobi preconditioner, which can be implemented in a Jacobian-free fashion. This method only requires the diagonal of the Jacobian. While it seems numerically advantageous, many linear iterations might be required for convergence. The second alternative is the multiplicative-additive Schwarz preconditioned inexact Newton [59] approach, which is a nonlinear Jacobian-free preconditioning method designed to overcome unbalanced nonlinearities stemming from different ranges of time and spatial scales, such as shock waves and reaction fronts. The third alternative is the Lower-upper Symmetric Gauss-Seidel (LU-SGS) [60] preconditioner, which was originally developed as a solver for inviscid flows on structured grids, but has recently been extended to viscous flows and unstructured meshes [61, 62]. In LU-SGS, the implicit operator is simplified by introducing a Roe-type flux approximation that replaces the Roe matrix with its spectral radius. Consequently, only the diagonal part of the Jacobian is stored, and the products of the off-diagonal terms and the solution update are approximated by a Fréchet derivative through a forward and a backward sweep. The matrix-free preconditioner developed in this work is LU-SGS, as many authors [60, 63] have reported its remarkable robustness, efficiency, and low memory cost.

1.5 Thesis Objectives

The CFD Lab at McGill University is developing a hypersonic code named High Altitude Low Orbit (HALO3D) with collaborations from ANSYS and Lockheed Martin. This thesis focuses on the development of a numerical approach to simulate multi-species, chemically reacting non-equilibrium flows, with special emphasis on the thermal non-equilibrium condition. It aims to provide an approach that preserves good computational efficiency, robustness, and accuracy. This work focuses on thermo-chemical non-equilibrium. The effects of ionization, ablation, catalysis, radiation, and magnetic field control are not considered but left for parallel efforts at the CFD Lab. In addition to the numerical modelling of hypersonic flows, this thesis also aims at studying novel technologies of solving linear systems. Therefore, a JFNK solver is developed for compressible flows as a preliminary work before hypersonic flows are introduced.

1.6 Thesis Contributions

The engineering contribution is to provide an easy-to-use software toolkit that can simulate hypersonic flows in a cost-effective manner. A graphical user interface is provided that can set up flow configurations efficiently and user-friendly. Inside the toolkit, there are specific tools to monitor the convergence file, to visualize the solutions, and to edit grids and solutions. Installment guides and tutorials are also provided. Most of the above-mentioned functionalities were developed by former students at the CFD Lab and experts from ANSYS. Thanks to their contributions, it is possible to update the numerical technologies and develop new physical models in the HALO3D framework without reinventing the wheel.

The author's contributions are summarized as follows:

1. Development of a JFNK solver with LU-SGS matrix-free preconditioner for compressible flows
2. Improvement of the original LU-SGS by accounting for the viscous Jacobian on the fly during the two sweeps and including Riemann, supersonic outlet, slip wall, and non-slip wall boundary conditions in the Jacobian
3. Development of an edge-based FEM two-temperature thermal non-equilibrium solver for hypersonic flows
4. Study of anisotropic mesh optimization for hypersonic flows.

1.7 Thesis Outline

This thesis is organized as follows: Chapter 2 illustrates the governing equations and the physical models. Chapter 3 presents the edge-based FEM numerical formulations. Chapter 3.8 presents the numerical results of hypersonic flows. Chapter 5 describes a Jacobian-free solver for compressible flows, a necessary step to assess the characteristics of JFNK and LU-SGS before introducing more complex non-equilibrium effects. Chapter 6 gives conclusions and future developments.

2 Mathematical Formulation

2.1 Governing Equations

The equations governing unsteady compressible viscous two-temperature hypersonic flows in conservative form are [30]

$$\frac{\partial Q}{\partial t} + \nabla \cdot (\mathbf{F}^A(Q) - \mathbf{F}^V(Q, \nabla Q)) = S \tag{2.1}$$

where Q is the vector of conservative variables, given as

$$Q = \begin{Bmatrix} \rho Y_s \\ \rho u \\ \rho v \\ \rho w \\ \rho e \\ \rho e_{ve} \end{Bmatrix} \tag{2.2}$$

ρ is the density, Y_s is the mass fraction of the s-th species, u, v, w are the velocities components in the x, y, z directions, e is the total energy per unit mass and e_{ve} is the vibrational-electronic energy per unit mass.

The inviscid fluxes $\mathbf{F}^A$ in the x, y, z directions are

$$F_x^A = \begin{Bmatrix} \rho Y_s u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(\rho e + p) \\ \rho e_{ve} u \end{Bmatrix}, \quad F_y^A = \begin{Bmatrix} \rho Y_s v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ v(\rho e + p) \\ \rho e_{ve} v \end{Bmatrix}, \quad F_z^A = \begin{Bmatrix} \rho Y_s w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ w(\rho e + p) \\ \rho e_{ve} w \end{Bmatrix} \tag{2.3}$$

where p is the pressure. The viscous fluxes $\mathbf{F}^V$ in the x, y, z directions are given as

$$F_x^V = \begin{Bmatrix} -J_{x,s} \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{xx} u + \tau_{xy} v + \tau_{xz} w - \left(q_{tr,x} + q_{ve,x}\right) - \sum_s J_{x,s} h_s \\ -q_{ve,x} - \sum_s J_{x,s} e_{ve,s} \end{Bmatrix}$$

$$F_y^V = \begin{Bmatrix} -J_{y,s} \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ \tau_{yx}u + \tau_{yy}v + \tau_{yz}w - \left(q_{tr,y} + q_{ve,y}\right) - \sum_s J_{y,s}h_s \\ -q_{ve,y} - \sum_s J_{y,s}e_{ve,s} \end{Bmatrix}$$

and

$$F_z^V = \begin{Bmatrix} -J_{z,s} \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ \tau_{zx}u + \tau_{zy}v + \tau_{zz}w - \left(q_{tr,z} + q_{ve,z}\right) - \sum_s J_{z,s}h_s \\ -q_{ve,z} - \sum_s J_{z,s}e_{ve,s} \end{Bmatrix} \tag{2.4}$$

where $J_{i,s}$ is the s-th species mass diffusion flux in the i direction, τ_{ij} is the component of the viscous stress tensor $\boldsymbol{\tau}$, h_s is the s-th species enthalpy, $q_{tr,i}$ and $q_{ve,i}$ are the translation-rotational and vibrational-electronic heat fluxes in the i direction, respectively.

2.2 Viscous Stresses and Heat Fluxes

The Newtonian fluid assumption and the Stokes hypothesis are used to define the viscous stress

$$\boldsymbol{\tau} = \mu(\nabla \boldsymbol{V} + \nabla \boldsymbol{V}^T) + \lambda(\nabla \cdot \boldsymbol{V})\mathbf{I} \quad \text{with} \quad \lambda = -\frac{2}{3}\mu \tag{2.5}$$

where μ is the mixture dynamic viscosity coefficient, $\boldsymbol{\tau}$ is the viscous stress tensor, $\boldsymbol{V}$ is the velocity vector and $\mathbf{I}$ is the identity tensor. The heat fluxes are modeled from Fourier's law which states that the direction of the heat flux is along the negative temperature gradient

$$\boldsymbol{q}_{tr} = -k_{tr}\nabla T_{tr} \quad \text{and} \quad \boldsymbol{q}_{ve} = -k_{ve}\nabla T_{ve} \tag{2.6}$$

where $\boldsymbol{q}$ is the heat flux, k is the heat conductivity coefficient and T is the temperature. The subscripts tr and ve represent translational-rotational and vibrational-electronic parts of the quantities, respectively.

The species mass diffusion flux is modelled using Fick's law which states that the species mass flux goes from high concentration regions to low concentration regions

$$\boldsymbol{J}_s = -\rho\widetilde{D}_s \nabla Y_s \tag{2.7}$$

where $\widetilde{D}_s$ is the species mass diffusion coefficient.

2.3 Thermodynamic Relations

The medium is assumed to be a mixture of N_s species that follows the idea gas law. This assumption is valid for the typical hypersonic flow conditions, where the temperature is above a hundred Kelvins and the pressure is below a thousand atmospheres. The equation of state links the translational-rotational temperature T_{tr}, partial density ρ_s and partial pressure p_s together as

$$p_s = \rho_s R_s T_{tr} \tag{2.8}$$

where that R_s is the gas constant for species s, defined as

$$R_s = \frac{R_u}{M_s} \tag{2.9}$$

The mixture pressure and the mixture density are obtained from Dalton's law for perfect gases

$$p = \sum_{s=1}^{N_s} p_s, \qquad \rho = \sum_{s=1}^{N_s} \rho_s \tag{2.10}$$

Y_s is the species mass fractions defined as the ratio between the mass of the species and the mass of the mixture

$$Y_s = \frac{\rho_s}{\rho} \tag{2.11}$$

Y_s satisfies the condition

$$\sum_{s=1}^{N_s} Y_s = 1 \tag{2.12}$$

The total energy per unit mass, e, is given by the sum of kinetic and internal energies, defined as

$$e = \frac{1}{2}(u^2 + v^2 + w^2) + \sum_{s=1}^{N_s} Y_s e_{int,s} \tag{2.13}$$

where $e_{int,s}$ is the internal energy per mass for species s. The total enthalpy per unit mass h related to the total energy e by the following relation

$$h = e + \frac{p}{\rho} \tag{2.14}$$

The internal energy can be stored in atoms and molecules in different forms, referred to as energy modes. For the atoms, there are two energy modes, which are the translational energy associated with the motion of the particle and the electronic energy associated with electrons orbiting around the nucleus. For the molecules, there are two additional energy modes, which are the rotational energy that is associated with the angular kinetic energy and the vibrational energy that is associated with the vibration of the atoms of a molecule. Each energy mode independently follows a Boltzmann distribution, possibly at a different temperature. The two-temperature thermal non-equilibrium model used in this work assumes that the translational and rotational energy of all species are in equilibrium with one temperature T_{tr}, and the vibrational and electronic energy of all species are in equilibrium with another temperature T_{ve}. The species s internal energy $e_{int,s}$ in thermal non-equilibrium can then be obtained from the following expression

$$e_{int,s} = e_{t,s}(T_{tr}) + e_{r,s}(T_{tr}) + e_{v,s}(T_{ve}) + e_{e,s}(T_{ve}) + h_{0,s} \tag{2.15}$$

Where $e_{t,s}$, $e_{r,s}$, $e_{v,s}$, and $e_{e,s}$ are the translational, rotational, vibrational, and electronic energies for the s-th species, respectively. $h_{0,s}$ is the species heat of formation assuming zero K as the reference temperature, given in Appendix B. The expressions of the energy modes can be derived from partition functions in quantum mechanics. Under the rigid-rotator-harmonic oscillator [64] molecule model, each energy mode is independent of the other. The translational energy and the rotational energy are assumed to be fully excited, which is valid in the typical hypersonic flow conditions. Therefore, those two energy modes are combined as translational-rotational energy

$$e_{tr,s}(T_{tr}) = e_{t,s}(T_{tr}) + e_{r,s}(T_{tr}) = C_{v,s}^{tr} T_{tr} \tag{2.16}$$

$C_{v,s}^{tr}$ is the species translational-rotational specific heat at constant volume, which is fully excited, thus

$$C_{v,s}^{t} = \frac{3}{2} R_s, \quad C_{v,s}^{r} = \begin{cases} R_s & n_{atoms} > 1 \\ 0 & n_{atoms} = 1 \end{cases}, \quad C_{v,s}^{tr} = C_{v,s}^{t} + C_{v,s}^{r} \tag{2.17}$$

where $C_{v,s}^{t}$ and $C_{v,s}^{r}$ are the species translational and rotational specific heat at constant volume, n_{atoms} is the number of atoms in the species. The species vibrational energy per unit mass $e_{v,s}$ is obtained by assuming a quantum harmonic oscillator, i.e.

$$e_{v,s} = \begin{cases} R_s \dfrac{\theta_{v,s}}{\exp(\theta_{v,s}/T_{ve}) - 1} & n_{atoms} > 1 \\ 0 & n_{atoms} = 1 \end{cases} \tag{2.18}$$

where $\theta_{v,s}$ is the characteristic vibrational temperature of species s, given in Appendix B. The harmonic oscillator assumption is valid for low vibrational energy levels, the contributions from high vibrational energy levels are negligible in calculating the total energy [28]. The species electronic energy per unit mass $e_{e,s}$ is modeled by summing over all the electronic energy levels

$$e_{e,s} = R_s \frac{\sum_{i=1}^{\infty} g_{i,s}\theta_{el,i,s}\exp(-\theta_{el,i,s}/T_{ve})}{\sum_{i=0}^{\infty} g_{i,s}\exp(-\theta_{el,i,s}/T_{ve})} \tag{2.19}$$

where $\theta_{el,i,s}$ is the characteristic electronic temperature at level i of species s. In practice, the summation is over a finite number of levels because the energy contributions from high levels are negligible.

2.3.1 Internal Energy Representation using NASA Polynomials

For the two-temperature thermal non-equilibrium model, it is possible to use the NASA polynomials [65] to compute the species internal enthalpy $h_{int,s}$, and therefore avoiding using Eq. (2.18) and (2.19). The NASA polynomials are polynomial fits of experimental data for the heat capacity of every species in thermodynamic equilibrium, where $T_{tr} = T_{ve} = T$. Therefore, a tilde is added to indicate that $\tilde{h}_{int,s}$ is only for thermodynamic equilibrium

$$\frac{\tilde{h}_{int,s}(T)}{R_u T} = -a_1 T^{-2} + a_2 \frac{lnT}{T} + a_3 + a_4 \frac{T}{2} + a_5 \frac{T^2}{3} + a_6 \frac{T^3}{4} + a_7 \frac{T^4}{5} + b_1 T^{-1} \tag{2.20}$$

In the case of the two-temperature model, the internal enthalpy $h_{int,s}$ is a function of both T_{tr} and T_{ve}. At reasonably high temperatures (>200K) the translational-rotational modes can be assumed to be fully excited. This allows expressing the energies of the translational-rotational and vibrational-electronic modes as

$$e_{tr,s}(T_{tr}) = h_{0,s} + C_{v,s}^{tr} T_{tr} \qquad \text{and} \qquad e_{ve,s}(T_{ve}) = \tilde{h}_{int,s}(T_{ve}) - R_s T_{ve} - \tilde{e}_{tr,s}(T_{ve}) \tag{2.21}$$

where $C_{v,s}^{tr} = C_{p,s}^{tr} - R_s$ is the species translational-rotational specific heat at constant volume.

2.4 Transport Properties

Transport properties such as viscosity and conductivity are required in the simulations. At high temperatures, the transport properties are dependent on temperatures and mixture compositions, and they can be theoretically computed by the kinetic theory of gases [66]. The results of the calculations are presented in this section, which is adequate for non-ionized flows with flow temperatures less than around 10,000K.

2.4.1 Viscosity

Gordon and McBride [67] provide NIST least-squares polynomials for the species viscosity μ_s , shown below

$$\ln \mu_s = A_s \ln T_{tr} + \frac{B_s}{T_{tr}} + \frac{C_s}{{T_{tr}}^2} + D_s \tag{2.22}$$

Another representation of viscosity is given by Blottner's curve fit [39], expressed as follows

$$\mu_s = 0.1 \exp[(A_s \ln T_{tr} + B_s) \ln T_{tr} + C_s] \tag{2.23}$$

The values of the coefficients are given in Appendix B.

2.4.2 Thermal Conductivity

In the simulation of thermal non-equilibrium flows, the translational-rotational and vibrational-electronic contributions to the thermal conductivities must be computed separately. Eucken's relation [40] is used to compute the translational-rotational and vibrational-electronic components of thermal conductivity

$$\kappa_{tr,s} = \mu_s \left(\frac{5}{2} C_{v,s}^t + C_{v,s}^r\right) \qquad \kappa_{ve,s} = \mu_s C_{v,s}^{ve} \tag{2.24}$$

2.4.3 Mixing Rule

To calculate the viscosity and thermal conductivity for the mixture, the Wilke's mixing rule [68] for viscosity and the Mason's mixing rule [69] for thermal conductivity is used. Both are based on a simplification of the first order Chapman-Enskog solution of the Boltzmann equation [66]. The transport properties of the mixture are computed as a molar weighted summation of the transport properties of the individual species.

$$\mu = \sum_{s=1}^{N_s} \frac{X_s \mu_s}{\phi_s} \qquad \text{and} \qquad \kappa = \sum_{s=1}^{N_s} \frac{X_s \kappa_s}{\phi_s} \tag{2.25}$$

where μ_s and κ_s are the species viscosity and thermal conductivity, respectively, X_s is the species molar fraction, defined as

$$X_s = \frac{\rho_s / M_s}{\rho / M} \qquad \text{with} \qquad M = \frac{1}{\sum_{s=1}^{N_s} Y_s M_s} \tag{2.26}$$

and ϕ_s is the scaling factor, given by

$$\phi_s = \sum_{i=1}^{N_s} X_i \left[1 + \sqrt{\frac{\mu_s}{\mu_i}} \left(\frac{M_i}{M_s}\right)^{\frac{1}{4}}\right]^2 \left[\sqrt{8\left(1 + \frac{M_s}{M_i}\right)}\right]^{-1} \tag{2.27}$$

2.4.4 Mass Diffusion Coefficient

The species mass diffusion coefficient $\widetilde{D}_s$ is replaces with a single coefficient $\widetilde{D}$ to ensure the sum of mass diffusion fluxes is zero in the domain. $\widetilde{D}$ is given by

$$\widetilde{D} = \frac{Le\ \kappa_{tr}}{\rho\ C_p^{tr}} \tag{2.28}$$

where Le is the Lewis number [70], taken as 1.4 in this work, κ_{tr} is the mixture conductivity of translational-rotational modes and C_p^{tr} is the mixture specific heat of translational-rotational modes at constant pressure.

2.5 Source Terms

The source terms appear in Eq. (2.1) are

$$S = \left\{ \begin{matrix} \dot{\omega}_s \\ 0 \\ 0 \\ 0 \\ 0 \\ S_{c-v} + S_{t-v} \end{matrix} \right\} \tag{2.29}$$

where $\dot{\omega}_s$ is the mass production or destruction of species s due to reactions, S_{c-v} is the coupling between vibrational-electronic energy and chemical reactions, S_{t-v} is the energy relaxation between translational and vibrational energy modes. A detailed description of the three terms is presented in the next two sections.

2.5.1 Chemical Kinetic Model

In this work, a laminar finite-rate chemistry model [15] is used, assuming that the reactions follow Arrhenius kinetics. A general system of N_r chemical reactions involving N_s chemical species can be represented as

$$\sum_{s=1}^{N_s} \nu'_{s,r}\,[A_s] \underset{k_r^b}{\overset{k_r^f}{\rightleftharpoons}} \sum_{s=1}^{N_s} \nu''_{s,r}\,[A_s], \qquad \forall r \in [1, N_r] \tag{2.30}$$

where A_s are the s-th species symbol and $\nu'_{s,r}$ and $\nu''_{s,r}$ are the reactant and product stoichiometric coefficients of species s and reaction r, respectively. According to the law of mass action, the production of mass per unit volume per second for the s-th species is thus expressed as

$$\dot{\omega}_s = M_s \sum_{r=1}^{N_r} \left(\nu''_{s,r} - \nu'_{s,r}\right)\left(R_r^f - R_r^b\right) \tag{2.31}$$

where the summation is over N_r chemical reactions, R_r^f and R_r^b are the forward and backward reaction rates of reaction r, defined as

$$R_r^f = k_r^f \prod_{s=1}^{N_s} [X_s]^{\nu'_{s,r}} \qquad \text{and} \qquad R_r^b = k_r^b \prod_{i=1}^{N_s} [X_i]^{\nu''_{i,r}} \tag{2.32}$$

The forward reaction rate coefficient k_r^f is modelled using the modified Arrhenius equation as

$$k_r^f(\bar{T}_c) = A_r^f \bar{T}_c^{\eta_r^f} \exp\left(-\frac{E_{a,r}}{R_u \bar{T}_c}\right) \tag{2.33}$$

Where A_r^f , η_r^f, and $E_{a,r}$ are reaction-specific parameters which are commonly given in the CGI unit in the literature. $\bar{T}_c$ is the corrected rate-controlling temperature. In Park's two-temperature model, to account for the fact that dissociative reactions are greatly influenced by the vibrational modes, the forward dissociative reactions are characterized by a combination of the translational-rotational and vibrational-electronic temperatures as [34]

$$\bar{T}_c = {T_{tr}}^{q} {T_{ve}}^{1-q} \tag{2.34}$$

In the literature, the most commonly used values for the exponent q are 0.5 and 0.7. Other reactions are characterized by the translation-rotational temperature T_{tr}. The backward reaction rates coefficient is given by the ratio between the forward reaction rates coefficient and the equilibrium constant

$$k_r^b = \frac{k_r^f}{K_r^{eq}} \tag{2.35}$$

The equilibrium constant can be obtained from curve fitting. Many experiments have been conducted which measured reaction rate coefficients for various temperature ranges, in the present work, a general polynomial is implemented in the code as [37]

$$K_r^{eq}(\bar{T}) = \exp\left(\begin{array}{c} B_{r,1}\left(\frac{1}{z^2}\right) + B_{r,2}\left(\frac{1}{z}\right) + B_{r,3} + B_{r,4}z + B_{r,5}z^2 \\ +B_{r,6}z^3 + B_{r,7}z^4 + B_{r,8}\log(z) + B_{r,9}\log\left(\frac{1}{z}\right) \end{array}\right) \tag{2.36}$$

where $z = 10000/\bar{T}$. Park's 1993 chemical mechanism is used in this work, and the reaction data are listed in Appendix C.

2.5.2 Vibrational-electronic Energy Equation Source Terms

Two source terms appear in the vibrational-electronic energy equation. The first term is the vibrational-electronic energy produced or removed by chemical reactions, given by

$$S_{c-v} = \sum_{s=1}^{N_s} \dot{\omega}_s D_s' \tag{2.37}$$

where D_s' is the produced vibrational energy, which can be modeled using either a preferential or non-preferential model. In the non-preferential model, it is assumed that the molecules are created or destroyed at the average vibrational energy level, therefore, D_s' equals the vibrational-electronic energy

$$D_s' = e_{ve,s}(T_{ve}) \tag{2.38}$$

In the preferential model, it is assumed that a molecule is more likely to dissociate in a higher vibrational state, therefore it is often assumed that the vibrational energy created or destroyed by chemical reactions equals a fraction of the dissociation energy of the molecule.

$$D_s' = \alpha_s D_s \tag{2.39}$$

where D_s is the dissociation potential of the chemical species, given in Appendix B and α_s is a fractional number typically taken as 0.3.

The second source term is the relaxation between translational and vibrational modes, which has the tendency to bring the translational energy and the vibrational energy to equilibrium. The relaxation source term is modelled using the Landau-Teller theory as [28, 34, 40]

$$S_{t-v} = \frac{\rho C_v^{ve}}{\tau}(T_{tr} - T_{ve}), \qquad \tau = \frac{\sum_{mol} \frac{\rho_s}{M_s}}{\sum_{mol} \frac{\rho_s}{M_s \tau_s}} \tag{2.40}$$

where the sum is over all the molecules, and τ_s is the relaxation time given by

$$\tau_s =< \tau_s > + \tau_{p,s} \tag{2.41}$$

The first term is the Landau-Teller relaxation time, given by

$$< \tau_s > = \sum_i X_i \Big/ \sum_i \frac{X_i}{\tau_{s,i}} \tag{2.42}$$

The inter-species relaxation time $\tau_{s,i}$ is obtained from Millikan and White's curve fits [71] as

$$\tau_{s,i} = \frac{101325}{p} \exp\left[A_{s,i}\left(T^{-\frac{1}{3}} - B_{s,i}\right) - 18.42\right] \tag{2.43}$$

where

$$A_{s,i} = 1.16 \cdot 10^{-3} \mu_{s,i}^{\frac{1}{2}} \theta_{v,s}^{\frac{4}{3}}, \qquad B_{s,i} = 0.015 \mu_{s,i}^{\frac{1}{4}} \qquad \text{and} \qquad \mu_{s,i} = \frac{M_s M_i}{M_s + M_i} \tag{2.44}$$

It is known that at high temperatures the Millikan-white curve fits underpredicts the vibrational relaxation time. Therefore, Park's correction time [72] is added as follows

$$\tau_{p,s} = \frac{1}{\sigma_s c_s N}, \qquad \text{with} \qquad c_s = \sqrt{\frac{8R_u T_{tr}}{\pi M_s}} \qquad \text{and} \qquad \sigma_s = 10^{-21}\left(\frac{50000}{T_{tr}}\right)^2 \tag{2.45}$$

where N is the mixture number density, c_s is the average molecular speed of the species and σ_s is the limiting cross-section.

3 Numerical Modeling

The weak-Galerkin formulation of Eq. (2.1) is obtained by multiplying the linear test function W_i and then integrate by parts [73]

$$\int_{\Omega} W_i \frac{\partial Q}{\partial t} - \int_{\Omega} \nabla W_i \cdot (\mathbf{F}^A - \mathbf{F}^V) + \int_{\partial\Omega} W_i \mathbf{n} \cdot (\mathbf{F}^A - \mathbf{F}^V) = \int_{\Omega} W_i S \tag{3.1}$$

The inner product between vectors is defined in Appendix A. A linear shape function is used in this work since a smooth higher-order shape function will not incorporate discontinuities, such as shock waves. Interpolating the conservative variables Q and the source term S by the linear shape function N_i at node i. The inviscid fluxes $\boldsymbol{F}^A$ is discretized by the group representation which yields a fourth-order truncation error on a uniform grid with linear elements in 2D [74]

$$Q(x,t) = \sum_{j=1}^{N_G} N_j(x) Q_j(t), \quad \mathbf{F}^A(x,t) = \sum_{j=1}^{N_G} N_j(x) \mathbf{F}^A{}_j(t), \quad S(x,t) = \sum_{j=1}^{N_G} N_j(x) S_j(t) \tag{3.2}$$

where N_G is the total number of nodes in the grid, Q_j, $\mathbf{F}^A{}_j$ and S_j are the nodal values which are only functions of time. Substituting Eq. (3.2) into Eq. (3.1) gets

$$\begin{gathered} \sum_{e \in E_i} \sum_{j \in K_e} \int_{V^e} W_i N_j \frac{dQ_j}{\partial t} dV - \sum_{e \in E_i} \sum_{j \in K_e} \int_{V^e} N_j \nabla W_i \cdot \mathbf{F}^A{}_j dV + \sum_{e \in E_i} \int_V \nabla W_i \cdot \mathbf{F}^V dV \\ + \sum_{e \in F_i} \int_A W_i \mathbf{n} \cdot (\mathbf{F}^A - \mathbf{F}^V)\, dA = \sum_{e \in E_i} \sum_{j \in K_e} \int_{V^e} W_i N_j S_j dV \end{gathered} \tag{3.3}$$

where E_i is the set of elements sharing the i-th vertex, F_i is the set of boundary faces sharing the i-th vertex, K_e is the set of nodes of the e-th element, V is the whole volume domain, and V^e is the volume of e-th element.

3.1 Lumped Mass Matrix

The lumped mass matrix is introduced for its benefit on easily finding the inverse

$$L_i = \sum_{e \in E_i} \sum_{j \in K_e} \int_{V^e} W_i N_j dV \tag{3.4}$$

The temporal term and the source term in Eq. (3.3) can then be written as

$$\sum_{e\in E_i}\sum_{j\in K_e}\int_{V^e} W_i N_j \frac{dQ_j}{\partial t} dV = L_i \frac{dQ_i}{\partial t} \quad and \quad \sum_{e\in E_i}\sum_{j\in K_e}\int_{V^e} W_i N_j S_j dV = L_i S_i \tag{3.5}$$

3.2 Inviscid Fluxes

An edge-based FE formulation is used to discretize the inviscid and viscous Fluxes. The edge-based assembly can easily handle hybrid meshes with a unique data structure and it is computationally more efficient than the element-based one [75]. It also allows the applications of Total Variation Diminishing (TVD) schemes that were originally developed in the FV context to provide stabilization for the advection terms. Furthermore, a one-to-one relation has been found between the edge-based FEM and FVM for simplicial [76] and non-simplicial [57] meshes.

The second term in Eq. (3.3) is the domain term of the inviscid fluxes $\mathbf{F}^A$. It can be reformulated in an edge-based fashion [77]

$$-\sum_{e\in E_i}\sum_{j\in K_e}\int_{V^e} N_j \nabla W_i \cdot \mathbf{F}^A{}_j dV = \sum_{j\in K_i} \boldsymbol{\eta}_{ij} \cdot \frac{\mathbf{F}^A{}_i + \mathbf{F}^A{}_j}{2} - \sum_{j\in K_i} \boldsymbol{\chi}_{ij} \cdot \frac{\mathbf{F}^A{}_j - \mathbf{F}^A{}_i}{2} \tag{3.6}$$

where K_i is the set of nodes connected to i-th vertex via an element, $\boldsymbol{\eta}_{ij}$ and $\boldsymbol{\chi}_{ij}$ are edge coefficients, defined as

$$\boldsymbol{\eta}_{ij} = \sum_{e\in E_i}\int_{V^e} \left(W_i \nabla N_j - N_j \nabla W_i\right) dV \qquad \boldsymbol{\chi}_{ij} = \sum_{e\in E_i}\int_{A} W_i N_j \mathbf{n}\, dA \tag{3.7}$$

Note that the coefficient $\boldsymbol{\chi}_{ij}$ is defined only for boundary edges. Those two edge coefficients only depend on elements sharing the i-th vertex. In the case of a stationary mesh, they can be computed at the beginning of the calculation. In Eq. (3.6), the first term on the right-hand side (RHS) is an arithmetic average of inviscid fluxes between node i and node j. To provide stabilization for advection-dominated flows, the vector of inviscid fluxes $\mathbf{F}^A$ is replaced with a numerical counterpart, $\boldsymbol{\Phi}^{\mathrm{num}}$ evaluated at the edge's midpoint, i.e.

$$\boldsymbol{\eta}_{ij} \cdot \frac{\mathbf{F}^A{}_i + \mathbf{F}^A{}_j}{2} = \boldsymbol{\Phi}^{\mathrm{num}}(Q_i, Q_j, \boldsymbol{\eta}_{ij}) \tag{3.8}$$

The numerical inviscid fluxes used in this work are the Roe fluxes [78], which are nonlinear functions of the nodal variables and of the edge coefficients. The second term in RHS of Eq. (3.6) can be seen as a "correction" factor, since it is proportional to the difference of fluxes along the edge. Following [76], it is left untouched and no additional dissipation is introduced for boundary edges. To introduce optimal numerical diffusion along stretched elements [57] the Galerkin edge coefficients are replaced by the so-called split-Galerkin coefficients. For tetrahedral elements, the two sets of coefficients are identical, whereas for hexahedral and prismatic elements the Galerkin edge coefficients along the diagonals are split into the principal edge directions, which eliminates the spurious artificial diffusion in the discretization of the advection term [57].

3.2.1 Viscous Fluxes

The third term in Eq. (3.3) is the domain terms of the viscous fluxes $\mathbf{F}^V$, which appear as the stress tensor, the inner product of stress tensor and velocity, heat flux, inter-diffusional mass flux, and inter-diffusional heat flux. The numerical formulations of these terms are addressed separately below.

After interpolating velocity by the shape functions, the stress tensor can be assembled in an edge-based fashion [76]

$$\sum_{e\in E_i}\int_V \nabla W_i \cdot \boldsymbol{\tau} dV = \sum_{j\in K_i}\left[\left(\mu_{ij}\mathrm{tr}(\mathbf{d}_{ij})\mathbf{I} + (\mu_{ij}+\lambda_{ij})\mathbf{d}_{ij}^{s} + (\lambda_{ij}-\mu_{ij})\mathbf{d}_{ij}^{A}\right)(\boldsymbol{V}_j - \boldsymbol{V}_i)\right] \tag{3.9}$$

The inner product between a vector and a second-order tensor is defined in Appendix A. The edge mid-point quantities, μ_{ij} and λ_{ij}, are calculated by the arithmetic average of the node values. The edge coefficient $\mathbf{d}_{ij}$ is a second order tensor defined as

$$\mathbf{d}_{ij} = \sum_{e\in E_i}\int_{V^e} \left(\nabla W_i \nabla N_j\right) dV \tag{3.10}$$

$\mathrm{tr}(\,)$ is the trace operator and $\mathbf{d}_{ij}^{S}$ and $\mathbf{d}_{ij}^{A}$ are the symmetric and anti-symmetric part of $\mathbf{d}_{ij}$, i.e.

$$\mathbf{d}_{ij}^{S} = \frac{\mathbf{d}_{ij} + \left(\mathbf{d}_{ij}\right)^T}{2} \quad \mathbf{d}_{ij}^{A} = \frac{\mathbf{d}_{ij} - \left(\mathbf{d}_{ij}\right)^T}{2} \tag{3.11}$$

Similarly, the inner product of stress tensor and velocity can be assembled in an edge-based fashion as

$$\sum_{e\in E_i}\int_V \nabla W_i \cdot (\boldsymbol{\tau}\cdot\boldsymbol{V}) dV = \sum_{j\in K_i}\left[\left(\mu_{ij}\mathrm{tr}(\mathbf{d}_{ij})\mathbf{I} + (\mu_{ij}+\lambda_{ij})\mathbf{d}_{ij}^{s} + (\lambda_{ij}-\mu_{ij})\mathbf{d}_{ij}^{A}\right) : \boldsymbol{V}_{ij}(\boldsymbol{V}_j - \boldsymbol{V}_i)\right] \tag{3.12}$$

The column product between two second-order tensors is defined in Appendix A.

The heat fluxes of translational-rotational modes can be assembled as

$$\sum_{e\in E_i}\int_V \nabla W_i \cdot \boldsymbol{q}_{tr} dV = -\sum_{j\in K_i}\left[k_{tr,ij}\mathrm{tr}(\mathbf{d}_{ij})(T_{tr,j} - T_{tr,i})\right] \tag{3.13}$$

The heat fluxes of vibrational-electronic mode are assembled in the same way.

The inter-diffusional mass flux can be assembled as

$$\sum_{e\in E_i}\int_V \nabla W_i\cdot \boldsymbol{J}_s dV = -\sum_{j\in K_i}\left[\rho_{ij}\widetilde{D}_{ij}\mathrm{tr}(\mathbf{d}_{ij})(Y_{s,j}-Y_{s,i})\right] \qquad (3.14)$$

The inter-diffusional heat flux in the total energy equation can be written as

$$\sum_{e\in E_i}\int_V \nabla W_i\cdot \sum_{s=1}^{N_s}(J_s h_s)\, dV = -\sum_{j\in K_i}\sum_{s=1}^{N_s}\left[\rho_{ij}h_{s,ij}\widetilde{D}_{ij}\mathrm{tr}(\mathbf{d}_{ij})(Y_{s,j}-Y_{s,i})\right] \qquad (3.15)$$

The inter-diffusional heat flux in the vibrational-electronic energy equation can be written as

$$\sum_{e\in E_i}\int_V \nabla W_i\cdot \sum_{s=1}^{N_s}(J_s e_{ve,s})\, dV = -\sum_{j\in K_i}\sum_{s=1}^{N_s}\left[\rho_{ij}e_{ve,s}\widetilde{D}_{ij}\mathrm{tr}(\mathbf{d}_{ij})(Y_{s,j}-Y_{s,i})\right] \qquad (3.16)$$

3.3 Loosely-coupled Strategy

The solution of a fully-coupled system may be prohibitive due to its potentially large dimension, depending on the number of species considered. In the present work, the system of $N_s + 6$ equations is split into three smaller subsystems which are solved in a loosely-coupled fashion. These are the flow equations for the conservation of the mixture quantities (mass, momentum, and total energy), the chemistry equations for the conservation of mass of the species, and the thermal non-equilibrium equations for the conservation of the vibrational-electronic energy of the mixture. A diagram of the iterative strategy is shown in Figure 1. The conservative variables Q at the $n-1$ time level are given as an input to the flow solver and the mixture quantities $\rho\boldsymbol{V}$, ρe, and ρ are updated, at the new time level n. This new set of variables, together with part of the old ones, is then used as an input for the chemistry solver which outputs the species density ρY_s at the time n. The current codes solve for both the density of the mixture and the density of every species. This introduces an extra constraint $\rho = \sum_s \rho Y_s$ that may be violated after the chemistry solver updates the species density. Therefore, a correction is made to ensure this constraint. Supposing $(\rho Y_s)^{\tilde{n}}$ is the s-th species density output by the chemistry solver, the correction is given by

$$(\rho Y_s)^n = \rho^n \frac{(\rho Y_s)^{\tilde{n}}}{\sum_s (\rho Y_s)^{\tilde{n}}} \qquad (3.17)$$

$(\rho Y_s)^n$ is used to update species density and to make sure the constraint $\rho = \sum_s \rho Y_s$ is satisfied.

In the literature, some codes [79] apply this constraint by solving for the density of the mixture and the density of $N_s - 1$ species, allowing the density of the N_sth species to be computed by resorting to the constraint $\rho Y_{N_s} = \rho - \sum_{s=1}^{N_s-1} \rho Y_s$. However, in the present work, the thermodynamic state of the gas is only updated after the solution of each individual solver, resulting in a Gauss-Seidel type update. Although this is believed to improve convergence, it eliminates the possibility of using the species mass constraint, since the change in mixture density and the change in species densities are effectively being computed at different time steps. Therefore, using this constraint would in practice result in instabilities at the beginning of the simulation. Finally, the updated set of variables and the "old" vibrational energy are input into the thermal non-equilibrium solver and the vibrational-electronic energy ρe_{ve} is updated. In addition, a consistent set of thermochemical states (pressure, temperatures, velocity, species mass fraction, vibrational-electronic energy) is updated after the solution of each individual solver. The loosely-coupled approach significantly reduces the implementation burden.

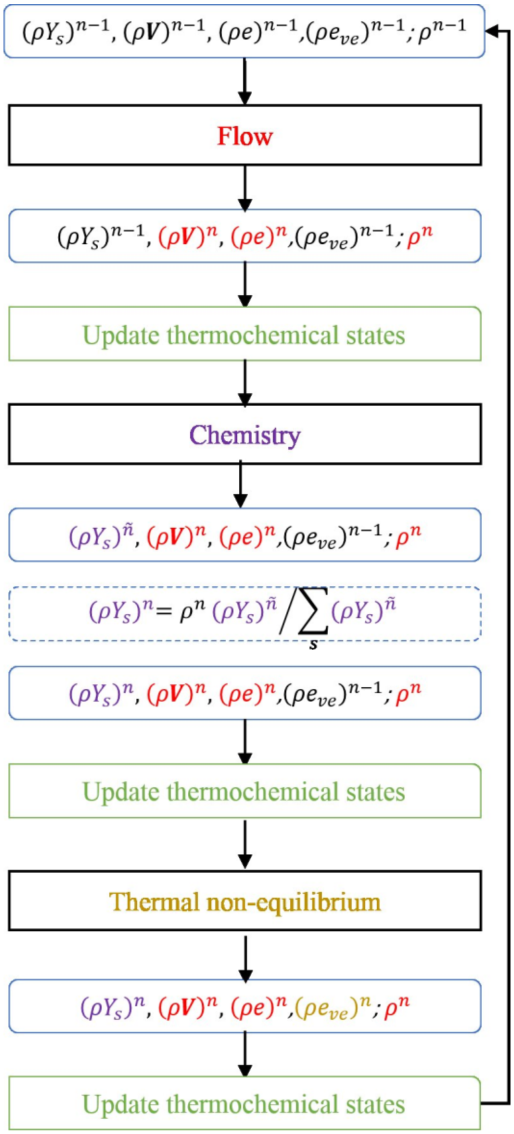

Figure 1. Illustration of the loosely-coupled strategy

3.4 Roe's Solver for Segregated Systems of Equations

In Eq. (3.8), the average of inviscid fluxes at node i and node j is replaced by a numerical flux evaluated at the edge midpoint, that is obtained by approximately solving the Riemann problem between node i and node j. In the flow solver, the Roe scheme [78] is used, which is

$$\boldsymbol{\Phi}_{ij}^{\text{num,flow}} \triangleq \boldsymbol{\Phi}^{\text{num,flow}}(Q_i, Q_j, \boldsymbol{\eta}_{ij}) = \boldsymbol{\eta}_{ij} \cdot \frac{\mathbf{F}^A{}_i + \mathbf{F}^A{}_j}{2} - \frac{|\tilde{A}(Q_i, Q_j, \boldsymbol{\eta}_{ij})|}{2}(Q_j - Q_i) \tag{3.18}$$

where $\tilde{A}$ is the Roe matrix with the following properties

1. $\tilde{A}$ is diagonalizable with real eigenvalues.
2. $\tilde{A}(Q_i, Q_j) \rightarrow A(Q_i)$ as $Q_i - Q_j \rightarrow 0$

3. $\tilde{A}(Q_j - Q_i) = \mathbf{F}^A(Q_j) - \mathbf{F}^A(Q_i)$

where Q is the set of conservative variables for the flow solver, A is the Jacobian of $\mathbf{F}^A$ with respect to Q , Q_i and Q_j are the left state and right state of the Riemann problem, respectively. The matrix $\tilde{A}$ is evaluated at the Roe average state, defined as

$$\begin{cases} \tilde{\rho} = \sqrt{\rho_i \rho_j} \\ \widetilde{\boldsymbol{V}} = \dfrac{\sqrt{\rho_i}\boldsymbol{V}_i + \sqrt{\rho_j}\boldsymbol{V}_j}{\sqrt{\rho_i} + \sqrt{\rho_j}} \\ \tilde{h} = \dfrac{\sqrt{\rho_i}h_i + \sqrt{\rho_j}h_j}{\sqrt{\rho_i} + \sqrt{\rho_j}} \end{cases} \tag{3.19}$$

and

$$\tilde{a}^2 = (\gamma - 1)\left[\tilde{h} - \frac{1}{2}\widetilde{\boldsymbol{V}} \cdot \widetilde{\boldsymbol{V}}\right]$$

Using the spectral decomposition, $\tilde{A}$ can be written as

$$\tilde{A} = \tilde{L}\tilde{\Lambda}\tilde{R} \tag{3.20}$$

where $\tilde{\Lambda}$ is the eigenvalues matrix given by

$$\tilde{\Lambda} = [\tilde{V}_n - \tilde{a}, \tilde{V}_n, \tilde{V}_n + \tilde{a}, \tilde{V}_n, \tilde{V}_n]\mathrm{I}_{5\times 5} \tag{3.21}$$

$\tilde{L}$ and $\tilde{R}$ are the left and right eigenvector matrices with definitions given in Appendix D. $\tilde{V}_n$ is the Roe average velocity projected at $\boldsymbol{\eta}_{ij}$ direction, i.e.

$$\tilde{V}_n = \widetilde{\boldsymbol{V}} \cdot \frac{\boldsymbol{\eta}_{ij}}{|\boldsymbol{\eta}_{ij}|} \tag{3.22}$$

It is well known that the Roe scheme cannot distinguish between a compression shock and an expansion shock [80]. This happens because one of the eigenvalues decreases toward zero and consequently the numerical dissipation associated with that specific wave vanishes. To remedy this problem, an entropy fix is used that applying the following modification to eigenvalues

$$\hat{\lambda}_i = \begin{cases} \dfrac{\tilde{\lambda}_i^2 + \delta^2}{2\delta} & \text{for } |\tilde{\lambda}_i| < \delta \\ |\tilde{\lambda}_i| & \text{for } |\tilde{\lambda}_i| \geq \delta \end{cases} \tag{3.23}$$

where δ is 0.3 in the present work.

For the chemistry solver, it is unsuitable to apply the Roe scheme directly since the decoupled chemistry system contains only degenerated eigenvalues. Following the work of [81], the numerical flux of each species density is computed by multiplying the total mass flux from the flow solver by the species mass fraction from the upwind node i.e.

$$\mathbf{\Phi}_{s,ij}^{\text{num,chemistry}} \triangleq \mathbf{\Phi}_{s}^{\text{num,chemistry}}(Q_i, Q_j, \boldsymbol{\eta}_{ij}) = \mathbf{\Phi}_{i,j}^{\text{num,flow,density}} \times \begin{cases} Y_{s,i} & \text{if } \mathbf{\Phi}^{\text{num,flow,density}} > 0 \\ Y_{s,j} & \text{if } \mathbf{\Phi}^{\text{num,flow,density}} \leq 0 \end{cases} \tag{3.24}$$

where $\mathbf{\Phi}_{s}^{\text{num,chemistry}}$ is the numerical flux in chemistry solver for species s and $\mathbf{\Phi}^{\text{num,flow,density}}$ is the density component of the numerical flux in the flow solver. This scheme automatically satisfies the consistent constraint that $\mathbf{\Phi}_{i,j}^{\text{num,flow,density}} = \sum_s \mathbf{\Phi}_{s,ij}^{\text{num,chemistry}}$.

In the thermal non-equilibrium solver, the Roe flux is computed by advecting the fluxes based on the sign of the eigenvalue, which is the Roe-averaged velocity at the edge midpoint, i.e.

$$\mathbf{\Phi}_{ij}^{\text{num,thermal}} \triangleq \mathbf{\Phi}^{\text{num,thermal}}(Q_i, Q_j, \boldsymbol{\eta}_{ij}) = \begin{cases} \rho_i e_{ve,i} \boldsymbol{V}_i & \text{if } \widetilde{\boldsymbol{V}} \cdot \boldsymbol{\eta}_{ij} > 0 \\ \rho_j e_{ve,j} \boldsymbol{V}_j & \text{if } \widetilde{\boldsymbol{V}} \cdot \boldsymbol{\eta}_{ij} \leq 0 \end{cases} \tag{3.25}$$

3.4.1 MUSCL Reconstruction

Approximate Riemann solvers can be made second-order accurate in space by resorting to the so-called Monotonic Upstream-Centered Scheme for Conservation Laws (MUSCL) [82]. The conservative variables are reconstructed to the edge midpoint as

$$Q_{i,ij}^{u} = Q_i + \nabla Q_i \cdot (\boldsymbol{x}_j - \boldsymbol{x}_i)/2 \quad \textit{and} \quad Q_{j,ij}^{u} = Q_j - \nabla Q_j \cdot (\boldsymbol{x}_j - \boldsymbol{x}_i)/2 \tag{3.26}$$

As per [83], the gradients are computed by constructing two upwind triangles for node i and node j, as illustrated in Figure 2. The upwind triangles are found aligned with the edge and the gradients are computed by the continuous Galerkin method.

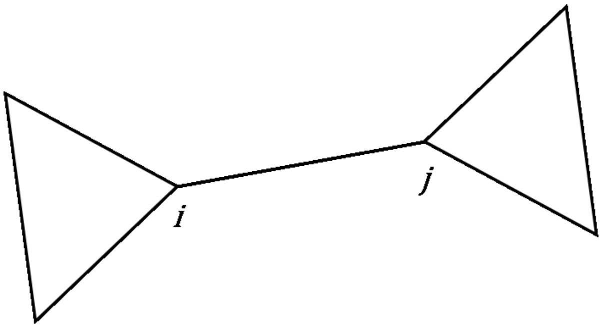

Figure 2. Illustration of upwind triangles

To avoid spurious oscillations near the shock, a 1D van Albada slope limiter [84] is introduced

$$Q_{i,ij} = Q_i - \frac{1}{2} lim\big(2(Q_i - Q_{i,ij}^{u}), Q_i - Q_j\big), \quad Q_{j,ij} = Q_j - \frac{1}{2} lim(2(Q_j - Q_{j,ij}^{u}), Q_j - Q_i) \tag{3.27}$$

where the limiter function is

$$lim(a,b) = \frac{(a^2+\varepsilon)b + (b^2+\varepsilon)a}{a^2+b^2+\varepsilon} \tag{3.28}$$

where ε is a small number which is 0.05 in this work.

3.5 Time Discretization

For each solver, the numerical formulation Eq. (3.3) can be written in a unified form as

$$L\frac{dQ}{dt} = R(Q) \tag{3.29}$$

where Q is the solution vector, L is the diagonal mass matrix, and R is the residual vector obtained by moving all the terms (except the first one) in LHS of Eq. (3.3) to RHS.

A Newton-Raphson procedure is then introduced to linearize the system

$$A(Q^{n-1})\varDelta Q^n = -R(Q^n) \tag{3.30}$$

where

$$\varDelta Q^n = Q^n - Q^{n-1} \qquad A = \frac{L}{\varDelta\tau^n}\mathbf{I} + \frac{\partial R}{\partial Q}\bigg|^n \tag{3.31}$$

Since only steady solutions are interested, the pseudo-time step $\varDelta\tau^n$ [85] is used to locally satisfy the CFL stability condition for linear advection problems and is in general increased as the solution progresses. Note that small values of $\varDelta\tau^n$ increase the diagonal dominance of the system matrix, making it easier to invert. As $\varDelta\tau^n$ increases, the first term of A gradually vanishes and the pseudo-unsteady problem reverts to the standard Newton's method.

3.6 Boundary Conditions

The fourth term in Eq. (3.3) is the boundary term and is treated differently for inviscid and viscous flows.

3.6.1 Inviscid Flow

For inviscid flows, inviscid fluxes are discretized in a nodal-based fashion.

$$\sum_{e\in F_i}\int_A W_i\mathbf{n}\cdot\mathbf{F}^A\,dA = \boldsymbol{\xi}_i\cdot\mathbf{F}^A(Q_i) + \sum_{j\in K_i}\left[\mathbf{F}^A\left(Q_j\right) - \mathbf{F}^A(Q_i)\right]\cdot\boldsymbol{\chi}_{ij} \tag{3.32}$$

where $\boldsymbol{\xi}_i$ is the nodal boundary coefficient, defined as

$$\boldsymbol{\xi}_i = \sum_{e\in F_i}\int_A W_i\mathbf{n}\,dA \tag{3.33}$$

In Eq. (3.32), the first term is the contribution from face nodes and the second term is the contribution from face edges. Since the second term is proportional to the difference of the fluxes of an edge, it is smaller than the first term. Therefore, the contribution from face edges is discarded. A ∂ is added to the superscript of $\mathbf{F}^A$ to indicate that the actual expression of $\mathbf{F}^A$ depends on the type of boundary conditions. Eq. (3.32) can be rewritten as

$$\sum_{e \in F_i} \int_A W_i \mathbf{n} \cdot \mathbf{F}^{A,\partial}\, dA = \boldsymbol{\xi}_i \cdot \mathbf{F}^{A,\partial} \tag{3.34}$$

Note that in this expression, the inviscid term can be assembled by simply looping over boundary nodes. The expressions of $\mathbf{F}^{A,\partial}$ are given below, depending on the boundary types.

3.6.1.1 Slip Wall Boundary Condition

For the flow solver, the slip boundary condition is of the form

$$\sum_{e \in F_i} \int_A W_i \mathbf{n} \cdot \mathbf{F}^{A,\partial}\, dA = [0, \boldsymbol{\xi}_i p_i, 0] \tag{3.35}$$

The chemistry and thermal non-equilibrium solvers correspond to mass and energy conservation equations, respectively. Therefore, the contribution of wall faces to the inviscid flux surface integral vanishes, since the velocity in the wall-normal direction is zero.

3.6.1.2 Riemann Boundary Condition

The Riemann boundary condition is used to specify inlet/outlet conditions based on the sign of the incoming/outgoing characteristics. Let Q_r be the Riemann state, Q_i be the internal state at the nodal point and Q_∞ be the boundary state. The Riemann state can be calculated by

$$Q_r = Q_i - [R(Q_i)N(Q_i)L(Q_i)](Q_\infty - Q_i) \tag{3.36}$$

where $R(Q_i)$ and $L(Q_i)$ are the right eigenvectors and left eigenvectors of the Jacobian of $\mathbf{F}^A$, respectively. N is a diagonal matrix that is one if the corresponding eigenvalue is negative and zero if it is positive, i.e.

$$N_i = -\max\big(0, \operatorname{sgn}(\lambda_i)\big) \tag{3.37}$$

where λ_i is the eigenvalue of the Jacobian of $\mathbf{F}^A$.

For the flow solver, the eigenvalues are given by

$$\Lambda = \{u_n - a, u_n, u_n + a, u_n, u_n\}^T \tag{3.38}$$

where u_n is the normal direction velocity and a is the speed of sound, given as

$$u_n = \boldsymbol{V} \cdot \mathbf{n} \quad a = \sqrt{\gamma \frac{p}{\rho}} \quad \text{with } \gamma = \frac{c_p^{tr}}{c_v^{tr}} \tag{3.39}$$

For the chemistry and thermal non-equilibrium solvers, the expressions of eigenvectors and eigenvalues are simpler, namely

$$\lambda_i = u_n \,, \quad R(Q_i) = L(Q_i) = \mathbf{I} \tag{3.40}$$

3.6.2 Viscous Flow

Since in 3D, the boundary elements are much fewer than the volume elements, the inviscid fluxes are discretized with the standard continuous Galerkin approximation and assembled by looping over the faces Gauss points, i.e.

$$\sum_{e \in F_i} \int_A W_i \mathbf{n} \cdot \mathbf{F}^{A,\partial} \, dA = \sum_{e \in F_i} \sum_{g=1}^{N_g} \left[w_g W_i(x_g) \boldsymbol{n}(\boldsymbol{x}_g) \cdot \mathbf{F}^{A,\partial} \right] \tag{3.41}$$

where N_g is the number of Gauss points for the face e and $\boldsymbol{x}_g$ is the Gauss point. Similar to Section 3.6.1, the actual expression of $\mathbf{F}^{A,\partial}$ depends on the type of boundary conditions. For the inlet/outlet, Riemann boundary condition is imposed, $\mathbf{F}^{A,\partial}$ is computed the same as in Section 3.6.1.2, except that the internal state at the nodal point Q_i is replaced by the internal state at the Gauss point $Q(\boldsymbol{x}_g)$. For the no slip wall boundary condition, $\mathbf{F}^{A,\partial}$ is taken as its internal state evaluated at the Gauss point, i.e. $\mathbf{F}^{A,\partial} = \mathbf{F}^{A}\big(Q(\boldsymbol{x}_g)\big)$.

The viscous fluxes in the boundary are also discretized with the standard continuous Galerkin approximation and assembled in an element-based fashion,

$$-\sum_{e \in F_i} \int_A W_i \mathbf{n} \cdot \mathbf{F}^{V,\partial} \, dA = -\sum_{e \in F_i} \sum_{g=1}^{N_g} \left[w_g W_i(\boldsymbol{x}_g) \boldsymbol{n}(\boldsymbol{x}_g) \cdot \mathbf{F}^{V,\partial} \left(Q(\boldsymbol{x}_g), \nabla Q(\boldsymbol{x}_g) \right) \right] \tag{3.42}$$

3.6.2.1 Dirichlet Boundary Conditions

The primitive variables such as velocity, pressure, temperatures, mass fractions are often imposed on the walls. Dirichlet boundary conditions are used to specify a value of one or more of the above quantities. In the present work, a Newton-Raphson procedure is used in which the residuals are linearized with respect to the conservative variables. In order to enforce a Dirichlet boundary condition on one of the conservative variables, the row corresponding to the variable is zeroed out from the system matrix and RHS and a value of one is placed on the diagonal. For example, to enforce the non-slip boundary condition, a zero momentum value is specified at the wall. If the velocity is already zero, this translates into the requirement that $\Delta \rho \boldsymbol{V} = 0$ for a Dirichlet node. Therefore the momentum equations for this node are multiplied by zero (RHS and LHS) and a 3-by-3 diagonal is placed on the momentum diagonal block. When primitive variables are imposed, e.g. the temperature, it is no longer possible to easily enforce the condition since the linear system is expressed in term of update in the conservative variables. It is thus necessary to convert a constraint expressed for the primitive variables into one expressed in terms of conservative variables.

3.6.2.1.1 Imposing Pressure

In the flow solver, imposing a fixed value of pressure is equivalent to imposing $\Delta p = 0$ for a given node. Since pressure is a primitive variable, it is necessary to express this constraint in terms of conservative variables as

$$\frac{\boldsymbol{V} \cdot \boldsymbol{V}}{2} \Delta\rho - \boldsymbol{V} \cdot \Delta(\rho \boldsymbol{V}) + \Delta e = 0 \tag{3.43}$$

This Dirichlet boundary condition is imposed by inserting the above relationship in the row corresponding to the variables in the LHS matrix and placing zeros in the corresponding row in the RHS, i.e.

$$\begin{bmatrix} \frac{\boldsymbol{V} \cdot \boldsymbol{V}}{2} & -\boldsymbol{V} & 1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} \Delta\rho \\ \Delta(\rho \boldsymbol{V}) \\ \Delta e \end{bmatrix} = \begin{bmatrix} 0 \\ \cdot \\ \cdot \end{bmatrix} \tag{3.44}$$

The imposition of other Dirichlet boundary conditions is done in a similar manner.

3.6.2.1.2 Imposing Temperatures

In the flow solver, imposing a fixed value of the translational-rotational temperature is equivalent to imposing $\Delta T_{tr} = 0$, which gives

$$\left(\frac{\boldsymbol{V} \cdot \boldsymbol{V}}{2} - e_{int}\right) \Delta\rho - \boldsymbol{V} \cdot \Delta(\rho \boldsymbol{V}) + \Delta e = 0 \tag{3.45}$$

In the thermal non-equilibrium solver, imposing a fixed value of the vibrational-electronic temperature is equivalent to imposing $\Delta T_{ve} = 0$, which gives

$$\Delta(\rho e_{ve}) = 0 \tag{3.46}$$

3.6.2.1.3 Imposing Mass Fractions

In the chemistry solver imposing a fixed value of mass fractions $\Delta Y_s = 0$ gives

$$(1 - Y_s)\Delta(\rho Y_s) - Y_s \sum_{\boldsymbol{i \neq s}} \Delta(\rho Y_i) = 0 \tag{3.47}$$

3.7 Parallelization

Parallelization is realized by the standard message passing interface (MPI) [86]. ParMETIS [87] is used to partition the domain that tries to minimize the communications between processors. The domain is divided into partitions with one partition per processor. The overlapping regions between partitions are updated once the solution is updated. A sparse parallel block AIJ format matrix is assembled in the PETSc framework [56]. The linear solver is GMRES [88] which is an iterative method for solving nonsymmetric linear systems. Block Jacobi preconditioning [89] is used that partitions the linear system into different blocks. By default, the number of blocks is the same as the number of processors. Since the blocks are used as preconditioners, it is not necessary to solve them exactly. Each block is solved approximately by applying its ILU [90] preconditioner of fill-in level zero with only one iteration. This procedure can be illustrated in Eq. (3.48), where n is the number of blocks, A_1, A_2, … A_n are the block matrices from the linear system, b_1, b_2, … b_n are the corresponding block vectors in the intermediate steps of GMRES, and P_1, P_2, … P_n are the ILU preconditioners of A_1, A_2, … A_n, respectively. Note that in the last step, the diagonal block A_i is approximately inverted by replacing it with the ILU preconditioner P_i.

$$\begin{bmatrix} A_1 & 0 & 0 & 0 \\ 0 & A_2 & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_n \end{bmatrix}^{-1} \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix} = \begin{bmatrix} {A_1}^{-1} b_1 \\ {A_2}^{-1} b_2 \\ \dots \\ {A_n}^{-1} b_n \end{bmatrix} \cong \begin{bmatrix} {P_1}^{-1} b_1 \\ {P_2}^{-1} b_2 \\ \dots \\ {P_n}^{-1} b_n \end{bmatrix} \tag{3.48}$$

3.8 Computing heat fluxes and shear stresses on walls

Following [91], the calculation of heat fluxes and shear stresses is performed via a consistent Galerkin approach. The weak form of the momentum equation in the x direction can be written as

$$\int_{\Omega} W_i \frac{\partial \rho u}{\partial t} dV - \int_{\Omega} \nabla W_i \cdot \left(\tilde{F}_x^A - \boldsymbol{\tau}_x\right) dV + \int_{\partial\Omega} W_i \mathbf{n} \cdot \left(\tilde{F}_x^A - \boldsymbol{\tau}_x\right) dA = 0 \tag{3.49}$$

where $\tilde{F}_x^A$ is the inviscid flux of the momentum equation in the x direction, and $\boldsymbol{\tau}_x$ is the stress tensor in the x direction

$$\tilde{F}_x^A = [\rho u^2 + p \quad \rho uv \quad \rho uw]^T \quad \text{and} \quad \boldsymbol{\tau}_x = [\tau_{xx} \quad \tau_{xy} \quad \tau_{xz}]^T \tag{3.50}$$

Since only the steady solution is of interest, the temporal term is neglected, and the equation becomes

$$\int_{\partial\Omega} W_i(\mathbf{n}\cdot\boldsymbol{\tau})_x\,dA = \int_{\partial\Omega} W_i\mathbf{n}\cdot\tilde{F}_x^A\,dA - \int_{\Omega} \nabla W_i\cdot\left(\tilde{F}_x^A - \boldsymbol{\tau}_x\right)dV \tag{3.51}$$

RHS of Eq. (3.51) has been constructed during the assembly of the momentum equations and LHS of Eq. (3.51) can be replaced by a lumped mass matrix, as in Section 3.1

$$L_i(\mathbf{n}\cdot\boldsymbol{\tau})_{x,i} = \int_{\partial\Omega} W_i\mathbf{n}\cdot\tilde{F}_x^A\,dA - \int_{\Omega} \nabla W_i\cdot\left(\tilde{F}_x^A - \boldsymbol{\tau}_x\right)dV \tag{3.52}$$

The solution of Eq. (3.52) is then the nodal value of the normal shear stress in the x direction. A similar approach can be applied to the y and z directions. For the heat fluxes, considering the weak form of the energy equation

$$\begin{aligned}\int_{\Omega} W_i\frac{\partial\rho e}{\partial t}dV - \int_{\Omega}\nabla W_i\cdot\left(\rho h\boldsymbol{V} - \boldsymbol{V}\cdot\boldsymbol{\tau} - k_{tr}\nabla T_{tr} - k_{ve}\nabla T_{ve} - \sum_s \rho\tilde{D}_s\nabla Y_s h_s\right)dV&\\ + \int_{\partial\Omega} W_i\mathbf{n}\cdot\left(\rho h\boldsymbol{V} - \boldsymbol{V}\cdot\boldsymbol{\tau} - k_{tr}\nabla T_{tr} - k_{ve}\nabla T_{ve} - \sum_s \rho\tilde{D}_s\nabla Y_s h_s\right)dA&\\ = 0&\end{aligned} \tag{3.53}$$

A similar result can be obtained after neglecting the temporal term

$$\begin{aligned}-\int_{\partial\Omega} W_i\mathbf{n}\cdot\left(k_{tr}\nabla T_{tr} + k_{ve}\nabla T_{ve} + \sum_s \rho\tilde{D}_s\nabla Y_s h_s\right)dA&\\ = \int_{\Omega}\nabla W_i\cdot\left(\rho h\boldsymbol{V} - \boldsymbol{V}\cdot\boldsymbol{\tau} - k_{tr}\nabla T_{tr} - k_{ve}\nabla T_{ve} - \sum_s \rho\tilde{D}_s\nabla Y_s h_s\right)dV&\\ - \int_{\partial\Omega} W_i\mathbf{n}\cdot(\rho h\boldsymbol{V} - \boldsymbol{V}\cdot\boldsymbol{\tau})\,dA&\end{aligned} \tag{3.54}$$

where LHS of Eq. (3.54) is the heat flux due to the translational-rotational, vibrational-electronic, and inter-diffusional heat conduction.

4 Thermo-chemical Non-equilibrium Flows

In this chapter, the results from selected simulations of thermo-chemical non-equilibrium flows are presented. The development of the thermo-chemical non-equilibrium solver is a CFD Lab team effort, to which many people took part. Although it is not always possible to rigorously separate the contributions of the different authors, it is fair to say a significant part of the development work for the chemistry solver has been carried out by Jory Seguin, while the author of this thesis focused on the development of the two-temperature thermal non-equilibrium solver. The two solvers are inherently intertwined, but for the sake of detailed results from chemical non-equilibrium, which can be found in Seguin's thesis [92], are not presented here.

The flow, chemistry, and thermal non-equilibrium solvers are loosely-coupled. In all of the solvers, the Jacobian is assembled and stored explicitly. Four test cases are presented. The first one is a cube filled with Oxygen at rest, which is used to validate the source term in the vibration-electronic energy equation that models the energy exchange between the translational and vibrational modes. The second one is a Mach 11.3 flow past a 3D blunt cone in thermal non-equilibrium with frozen chemistry. The third test case is a Mach 20 flow past a 2D cylinder. Two conditions are considered: thermal non-equilibrium with frozen chemistry and thermo-chemical non-equilibrium. The three above test cases all use structured girds. The fourth test case is a Mach 15.3 flow past a 3D sphere in thermo-chemical non-equilibrium. An unstructured grid is used, and an anisotropic mesh optimization technique is performed to improve the meshes and solutions. These cases are used to examine the accuracy and performance of the proposed formulation.

4.1 Zero-dimensional Cube

To validate the numerical formulation of the source term in the vibration-electronic energy equation, only the thermal non-equilibrium solver is activated. The test case is a cube filled with Oxygen at rest. Consequently, the convection term in the vibration-electronic energy equation vanishes, and each grid node is decoupled from others. The governing equation can be simplified as

$$\frac{d\rho e_{ve}}{dt} = S_{t-v} \tag{4.1}$$

Initially, the pressure is 0.063 atm, the translational-rotational temperature is 300K, and the vibrational-electronic temperature is 20,000 K. A backward-Euler method is used to discretize the temporal term and ensure unconditional stability. Since the solution of this problem is not spatiallydependent, the CFL number is not meaningful. A constant time step of 10^{-10}s is used to achieve adequate temporal accuracy and to compare to Casseau's results [93]. Electronic energy modes are not considered.

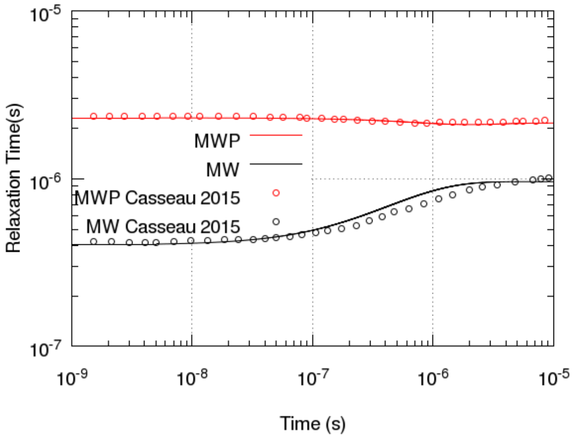

Figure 3. Zero-dimensional cube: translational-vibrational relaxation time versus physical time

Figure 3 plots the translational-vibrational relaxation time versus physical time. In the legend, MW corresponds to the relaxation time computed by Millikan-White formula and MWP stands for the relaxation time computed by Millikan-White's formula with Park's correction. Millikan-White's formula is a semi-empirical curve valid in the temperature range from 300K to 8000K. At higher temperatures, it is known that the Millikan-White formula under-predicts the relaxation time. Park corrected this by adding an extra term that more accurate estimates collision cross-sections. In Figure 4 (left), the translational-rotational (T-R) temperatures and the vibrational-electronic (V-E) temperatures are plotted against the reference, and it can be seen that the T-R and V-E temperatures eventually reach equilibrium. MW reaches equilibrium earlier than MWP, indicating that Park's correction increases the relaxation time at higher temperatures. Figure 4 (right) plots the pressure distributions versus time. The solid lines are results from this work and the circle points are reference results from [93]. Good agreement is found for all the results.

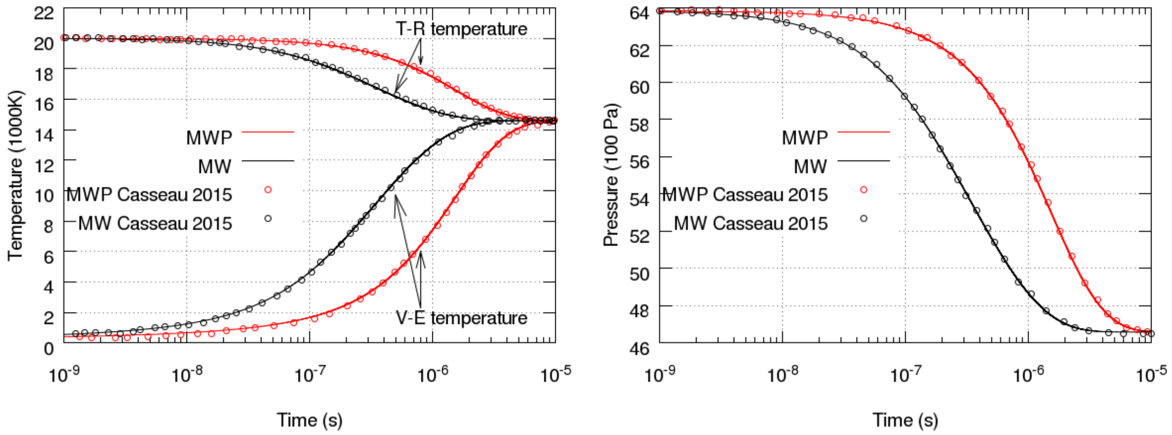

Figure 4. Zero-dimensional cube: translational-rotational temperature and vibrational-electronic temperature versus physical time (left). Pressure distribution versus physical time (right).

4.2 Mach 11.3 Flow past a Blunt Cone

This test case is a Mach 11.3 laminar flow of Nitrogen past a blunt cone in 3D. It is in thermal non-equilibrium but with frozen chemistry. The nose radius is 6.35 mm. The flat plane is 5 cm long in the streamwise direction and has a 25-degree angle with the stream direction. The Reynolds number based on the length of the blunt cone is 8,284. Since this problem is axisymmetric, it is possible to use a 2D mesh and solve the governing equations in cylindrical coordinates. Nevertheless, as HALO3D targets fully 3D large size industrial applications, a 3D structured mesh is used, as shown in Figure 5, consisting of 8,611,400 nodes and 8,568,144 elements. The height of the first layer of elements near the wall is 2 μm. The solution is obtained using the Roe scheme and extended to second order by the van Albada slope limiter. The initial conditions are: Mach number 11.3, velocity 2764.5 m/s, pressure 21.9 Pa, freestream T-R and V-E temperatures are 144.4 K. Both temperatures on the wall are 297.2 K. The far-field Kn number is 2.05×10^{-3}. Electronic energy modes are not considered. The initial CFL number is 0.1 and is exponentially increased to 3 within 1000 iterations.

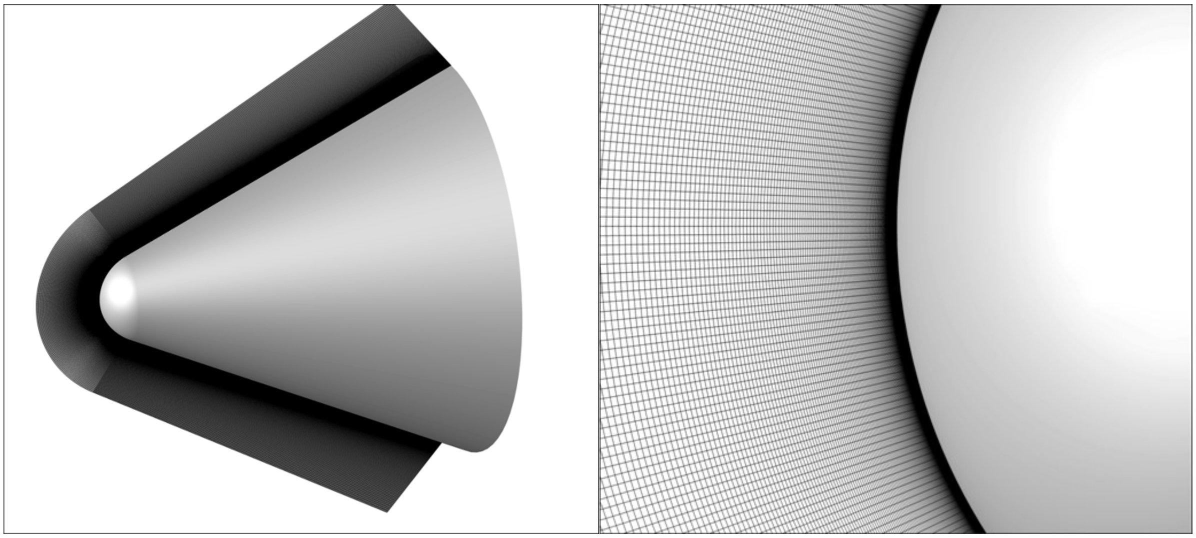

Figure 5. Mach 11.3 flow past a blunt cone: mesh (left) and mesh near the wall (right)

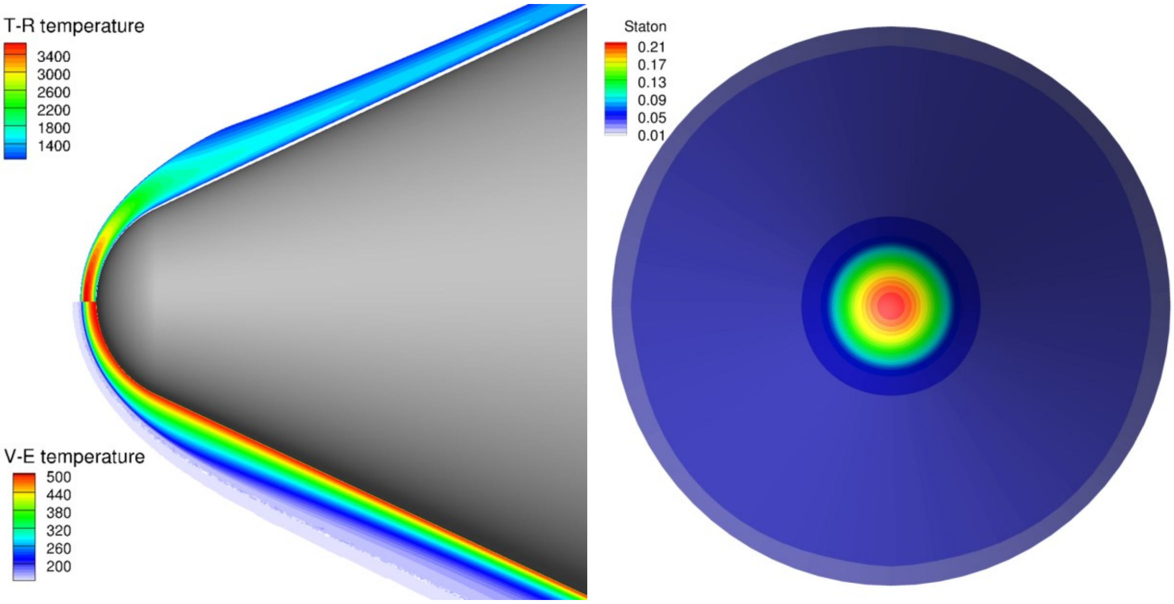

Figure 6. Mach 11.3 flow past a blunt cone: translational-rotational and vibrational-electronic temperature contours (left) and Stanton number contour (right).

Figure 6 plots the T-R and V-E temperature contours on the left, and Stanton number contours on the right. Figure 7 plots the convergence curves for flow solver and thermal non-equilibrium solver. The residual is defined as the L_2-norm of R in Eq. (3.29), with the definition of the L_2-norm given in Appendix A. The flow solver achieves roughly four orders of magnitude reduction in the residual norm, while the thermal non-equilibrium solver achieves roughly two orders of magnitude reduction in 2000 iteration. Figure 8 plots the skin friction coefficient, pressure coefficient, and Stanton numbers on the wall. The CFD results with the no-slip boundary condition are compared with Casseau's results where the Smoluchowski temperature jump condition and the Maxwell velocity slip jump boundary conditions are employed [51]. The far-field Kn number is still in the continuum regime and little difference is expected to be found between the profiles obtained from the two boundary conditions. Figure 9 plots the T-R temperature, V-E temperature, density, and velocity profiles along the stagnation line. The agreement of profiles along the stagnation line is satisfactory.

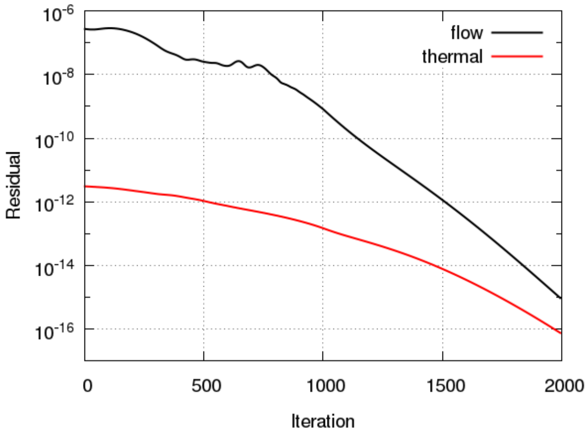

Figure 7. Mach 11.3 flow past a blunt cone: convergence curves

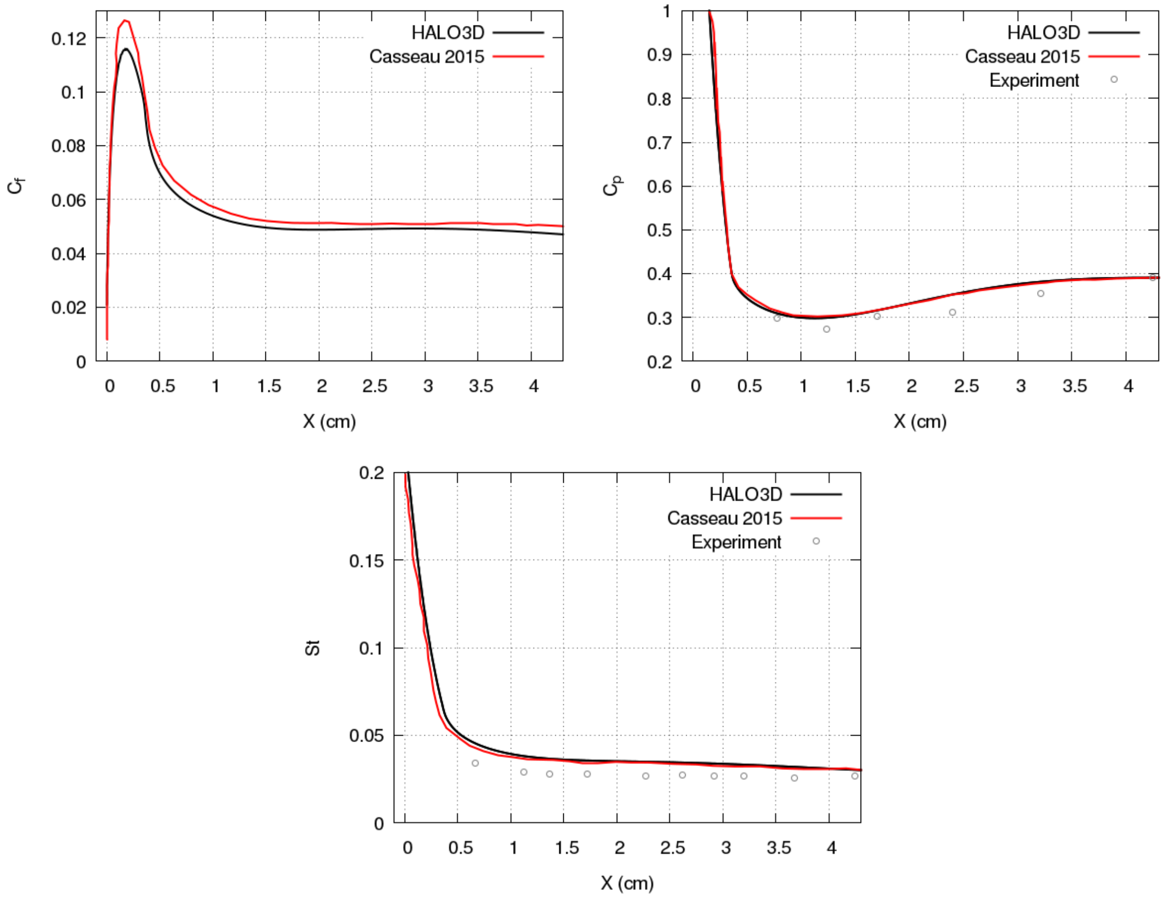

Figure 8. Mach 11.3 flow past a blunt cone: skin friction coefficient, pressure coefficient, and Stanton number on the wall.

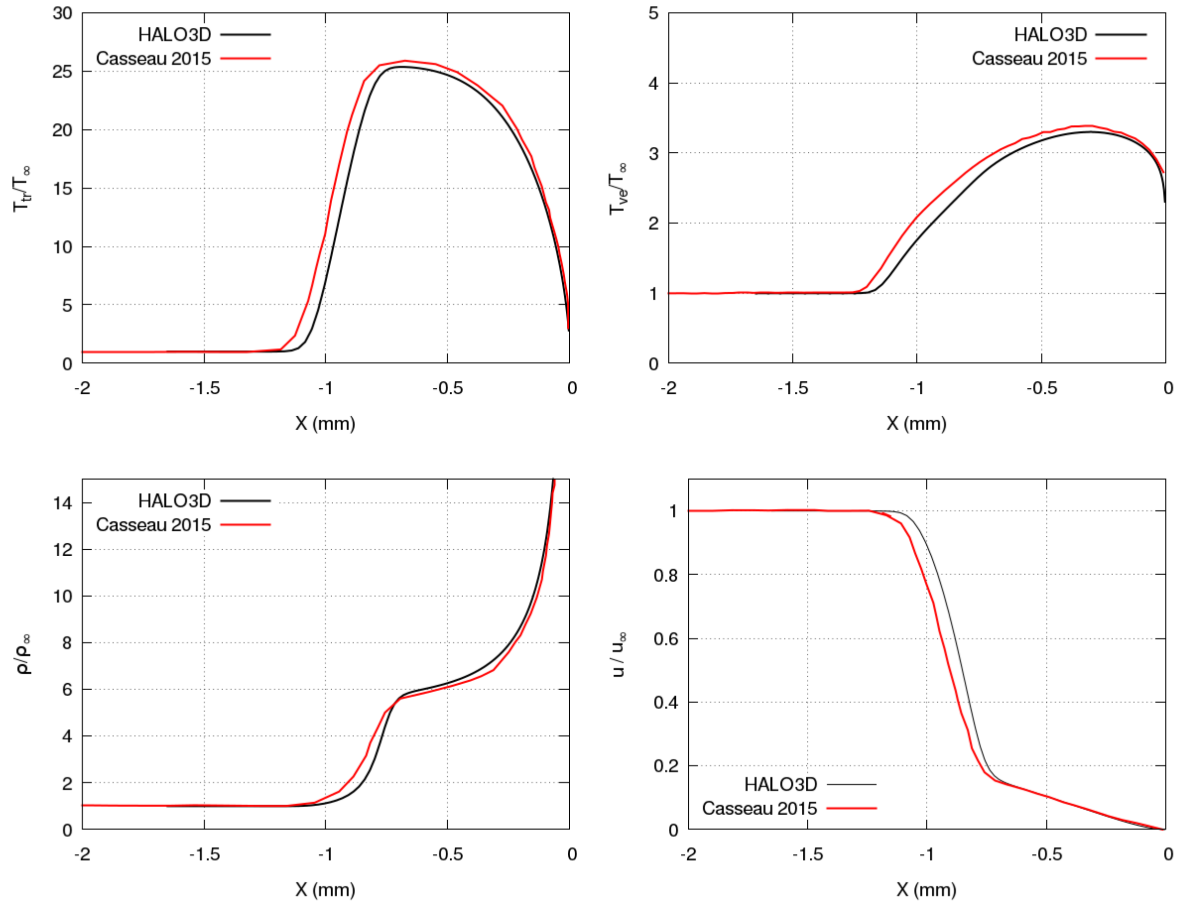

Figure 9. Mach 11.3 flow past a blunt cone: translational-rotational temperature, vibrational-electronic temperature, density, and velocity along the stagnation line.

4.3 Mach 20 flow past a Cylinder

This test case is a Mach 20 laminar flow of nitrogen past a cylinder with a radius of 1 m. The Reynolds number based on cylinder radius is 5,913. A 2D structured mesh is shown in Figure 10, consisting of 78,400 nodes and 78,204 elements. The height of the first layer of elements near the wall is 2 µm. The solution is obtained with the Roe scheme and extended to second order by the van Albada slope limiter. The initial conditions are: Mach number 20, velocity 6047 m/s, pressure 0.89 Pa, both freestream T-R and V-E temperatures are 220 K. An isothermal wall is used, with both temperatures set to 1,000 K. The far-field Kn is 5.1×10^{-3}. Two configurations are considered. The first one is thermal non-equilibrium with frozen chemistry, while the second one is in thermo-chemical non-equilibrium. Electronic energy modes are not considered.

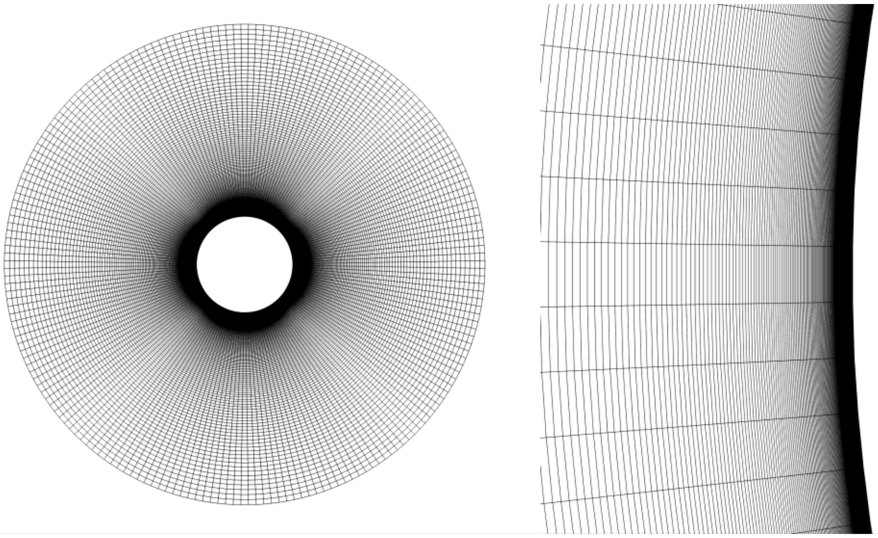

Figure 10. Mach 20 flow past a cylinder: mesh (left) and mesh near the wall (right)

4.3.1 Non-reacting Results

For the non-reacting flow, the initial CFL number is 0.1 and is exponentially increased to 10 within 1000 iterations. Figure 11 plots the T-R and V-E temperature contours on the left, and pressure contours on the right. Figure 12 plots the convergence curves for flow and thermal non-equilibrium solvers. The flow solver achieves roughly three orders of magnitude reduction in the residual norm, with both solvers stalling after 5,000 Newton iterations. The non-monotonic convergence at around 1,000 Newton iterations is due to the movement of the shock at the early stages of iteration. Once the shock is near its the correct position, the residual starts to decrease monotonically. Figure 12 (right) plots the convergence of integrated heat flux. Despite the residuals of flow and thermal non-equilibrium solvers stall, the heat flux converges after 5,000 Newton iterations. Figure 13 plots the skin friction coefficient, pressure coefficient, and heat fluxes on the wall. Figure 14 plots the T-R temperature, V-E temperature, density, and Mach number profiles along the stagnation line. The numerical results with the no-slip boundary condition (labelled HALO3D) are compared with Casseau's results [93] (labelled Casseau 2015) where the Smoluchowski temperature jump condition and the Maxwell velocity slip jump boundary conditions are employed [51]. Since the far-field's Knudsen number is still in the continuum regime, minor differences are expected between the profiles obtained from the two boundary conditions. The comparisons of all quantities are in good agreement.

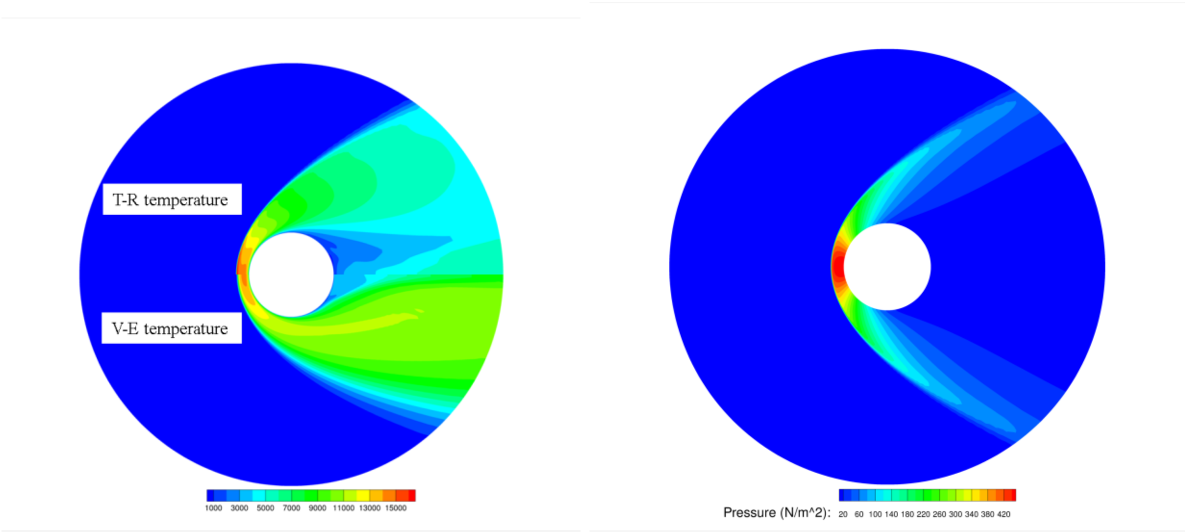

Figure 11. Mach 20 flow past a cylinder (non-reacting): translational-rotational and vibrational-electronic temperature contours (left) and pressure contour (right).

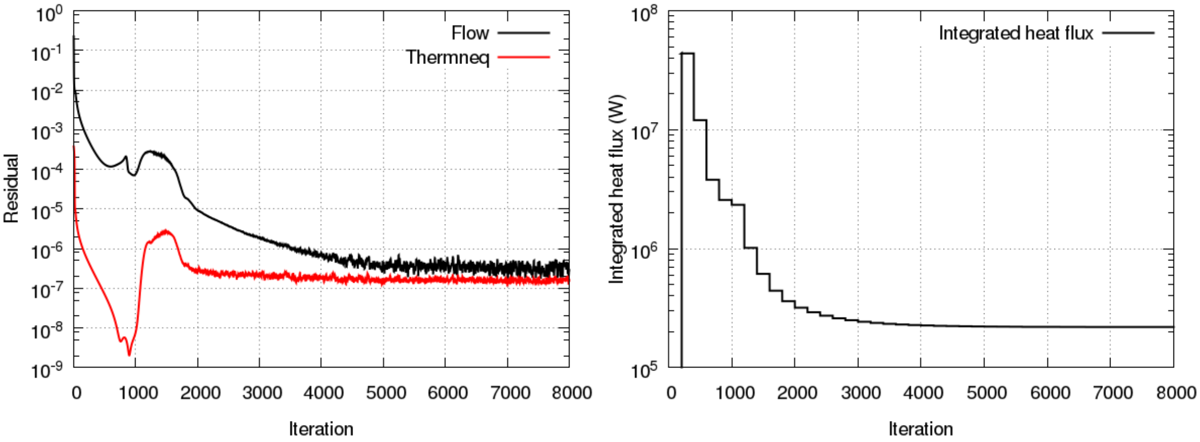

Figure 12. Mach 20 flow past a cylinder (non-reacting): convergence curves of solvers (left) and integrated heat flux (right)

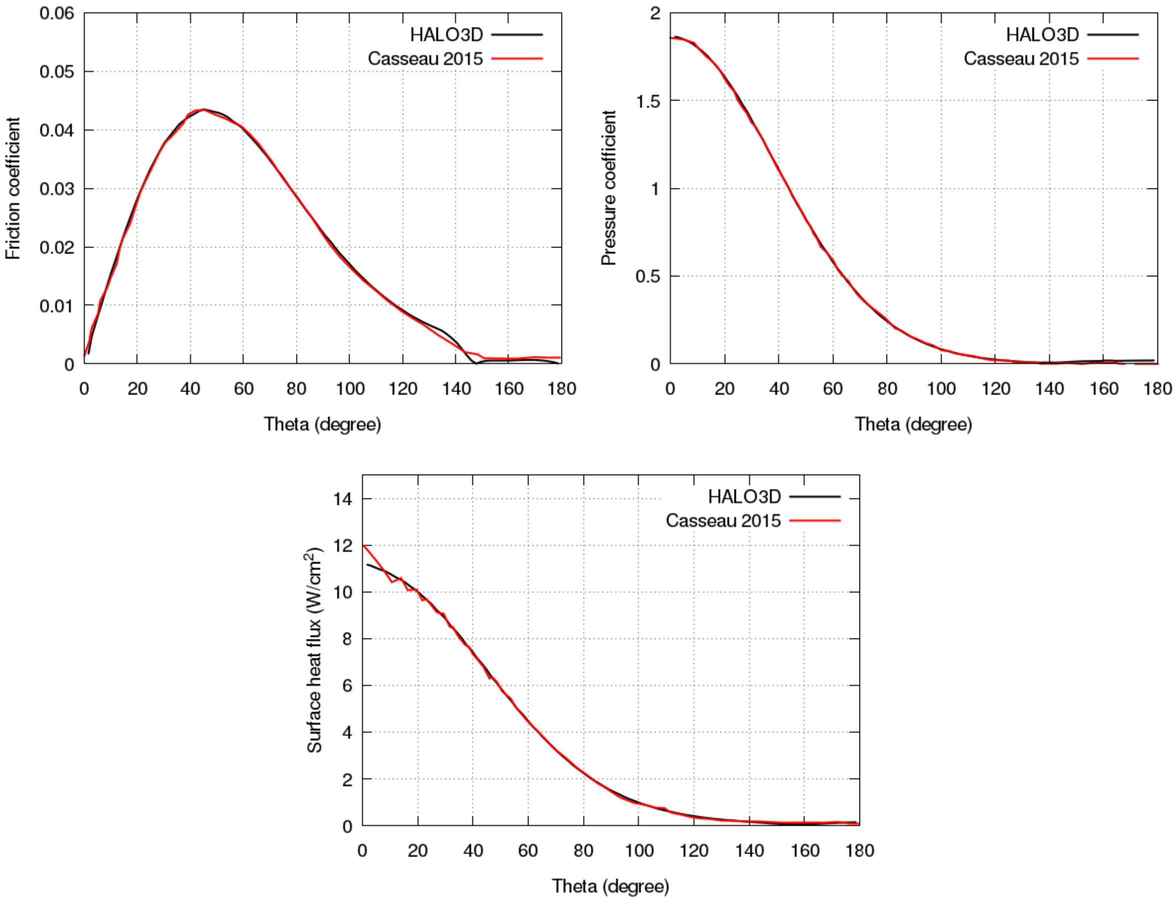

Figure 13. Mach 20 flow past a cylinder (non-reacting): skin friction coefficient, pressure coefficient, and surface heat flux on the wall.

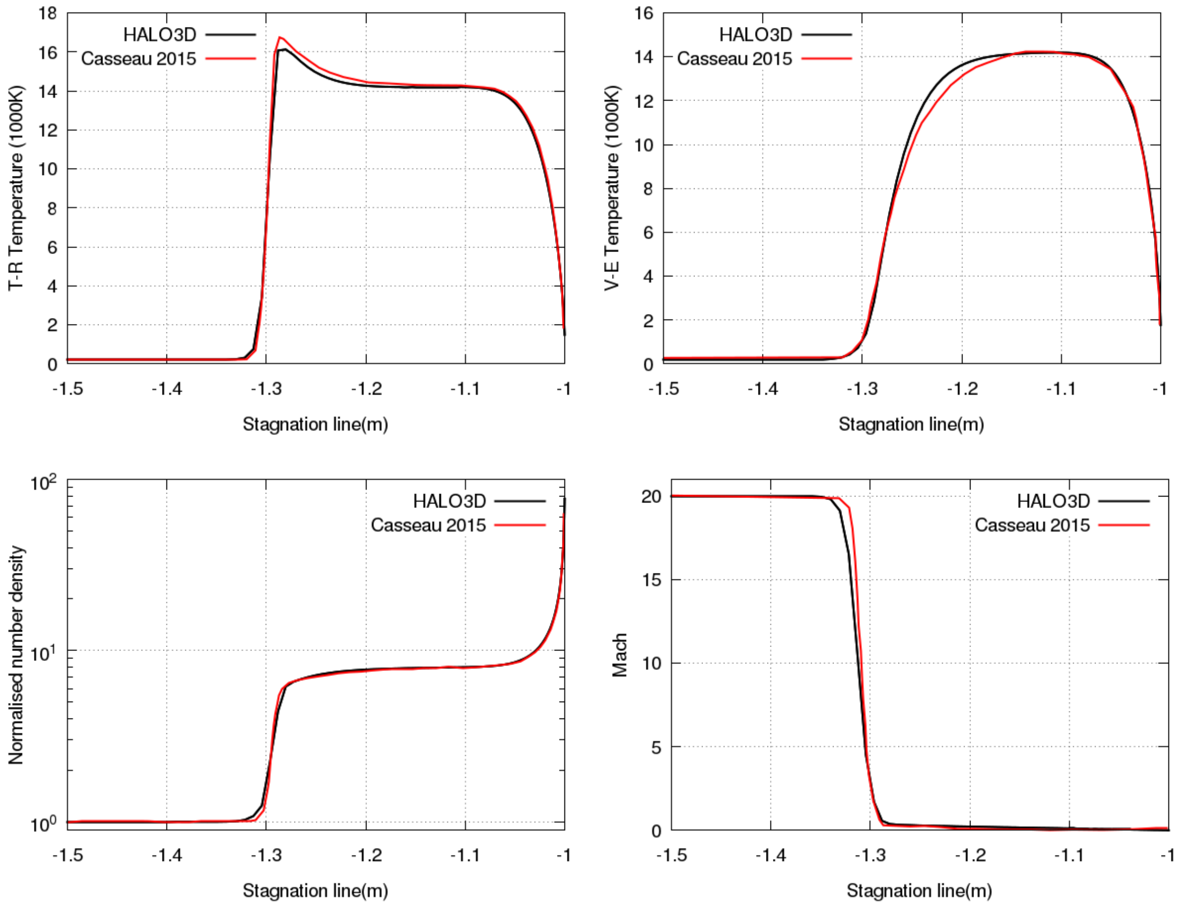

Figure 14. Mach 20 flow past a cylinder (non-reacting): translational-rotational temperature, vibrational-electronic temperature, density, and Mach number along the stagnation line.

4.3.2 Reacting Results

For the reacting case, two irreversible reactions are considered:

$$\begin{aligned} N_2 + N_2 &\rightarrow 2N + N_2 \\ N_2 + N &\rightarrow 2N + N \end{aligned} \tag{4.2}$$

The Arrhenius constants in Eq. (2.33) are given as

Reaction	Arrhenius constants		
	A_r^f	η_r^f	E_a
$N_2 + N_2 \rightarrow 2N + N_2$	7.0×10^{21}	-1.6	113,200
$N_2 + N \rightarrow 2N + N$	3.0×10^{22}	-1.6	113,200

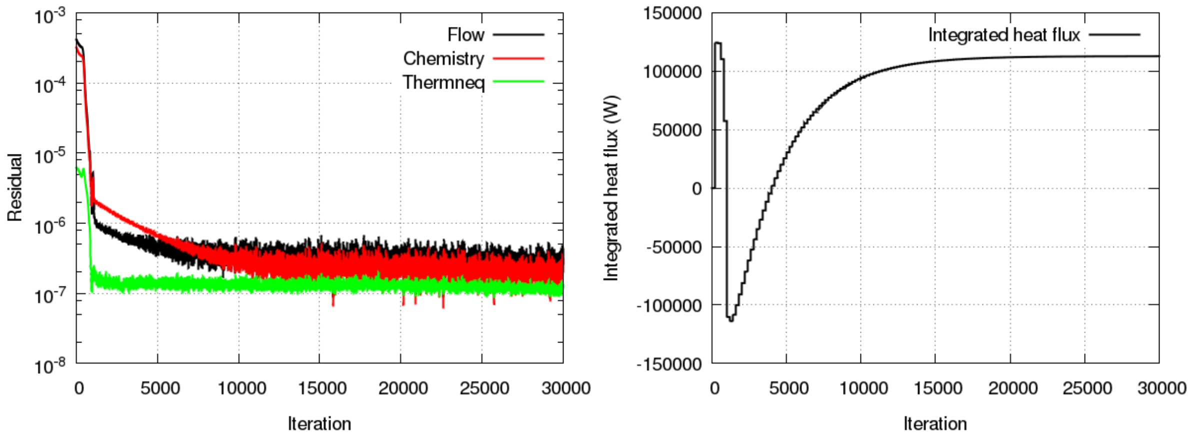

Figure 15. Mach 20 flow past a cylinder (reacting): convergence curves of solvers (left) and integrated heat flux (right)

The initial CFL number is 0.1 and is exponentially increased to 10 within 1000 iterations. Figure 15 (left) plots the convergence curves for the flow, chemistry, and thermal non-equilibrium solvers. The various solvers achieve roughly two to three orders of magnitude reduction in the residual norm, stalling after 10,000 Newton iterations. Figure 15 (right) shows the convergence of integrated heat flux. It is found that at 10,000 Newton iterations, the integrated heat flux is still changing and does not converge until 20,000 iterations. This suggests that when the solvers slow down, the physical quantities, such as integrated heat flux, may require more Newton iterations to reach convergence. Figure 16 and Figure 17 compare the temperature contours and profiles between non-reacting and reacting cases. The reacting case has a smaller shock standoff distance and a lower V-E temperature compared with the non-reacting case, which indicates internal energy is consumed in the chemical dissociations. Figure 18 plots the skin friction coefficient, pressure coefficient, and heat fluxes on the wall. Figure 19 plots the T-R temperature, V-E temperature, species density, and Mach number profiles along the stagnation line. The CFD results (labelled HALO3D) are compared with Casseau's results [51] (labelled Casseau 2016). The comparisons of stagnation line quantities are in good agreement, although slight differences can be seen. The friction coefficient on the wall is slightly under-predicted. For the heat flux on the wall, both CFD and DSMC results from Vincent are plotted. The results from HALO3D is in better agreement with the DSMC results. For the pressure coefficient on the wall, a good agreement is achieved. The differences between HALO3D and Casseau 2016 are probably due to different physical models used. HALO3D uses the isothermal wall boundary conditions, Wilke's mixing rule and the standard Eucken's relation, while Casseau uses the jump boundary conditions, Armaly and Sutton's mixing rule and the Eucken's relation with a coefficient of 1.2.

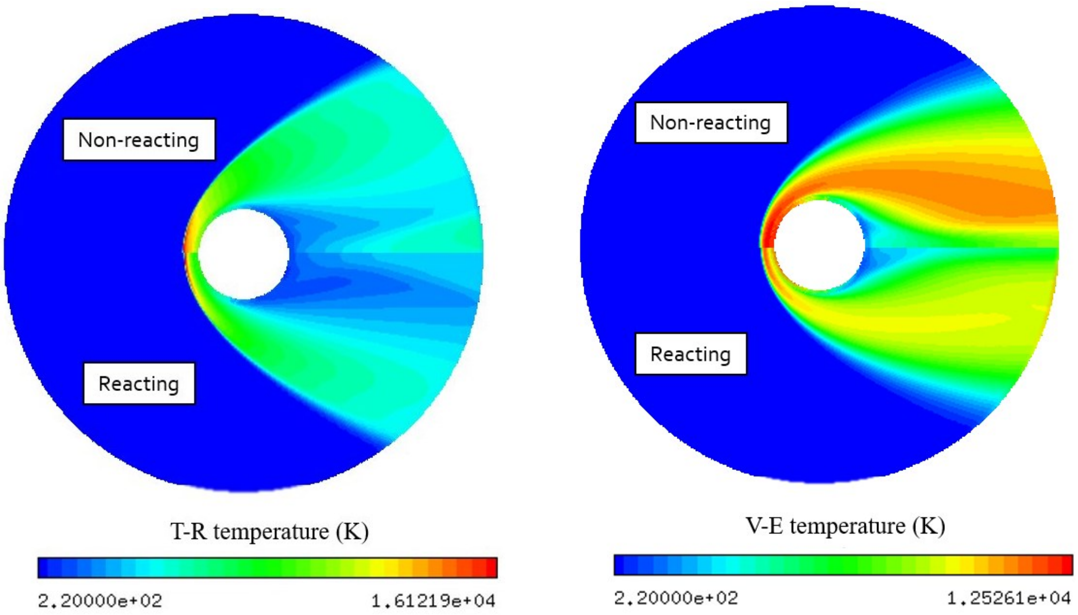

Figure 16. Mach 20 flow past a cylinder (comparison between non-reacting and reacting results): translational-rotational temperature contour (left) and vibrational-electronic temperature contour (right), the upper half is non-reacting contours and the lower half is reacting contours.

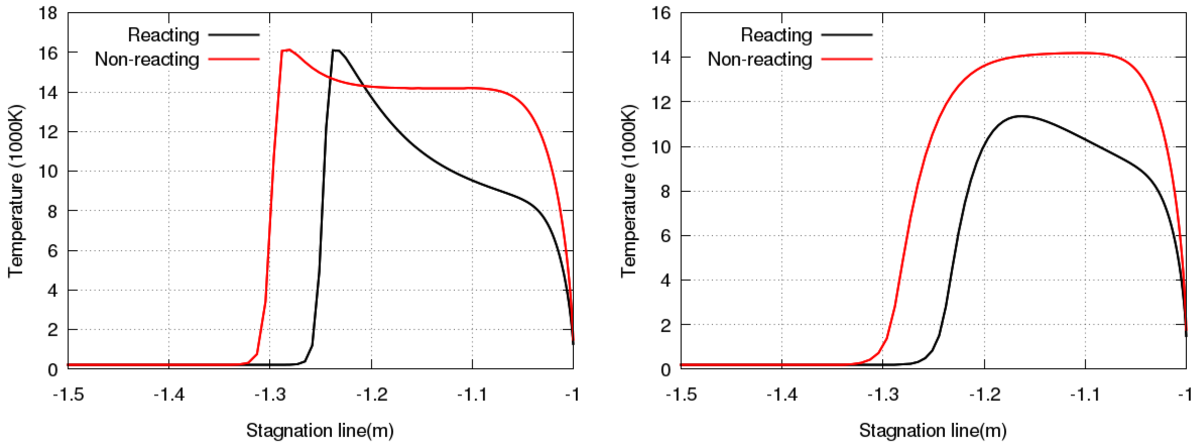

Figure 17. Mach 20 flow past a cylinder (comparison between non-reacting and reacting results): translational-rotational temperature (left) and vibrational-electronic temperature (right) profiles along stagnation line

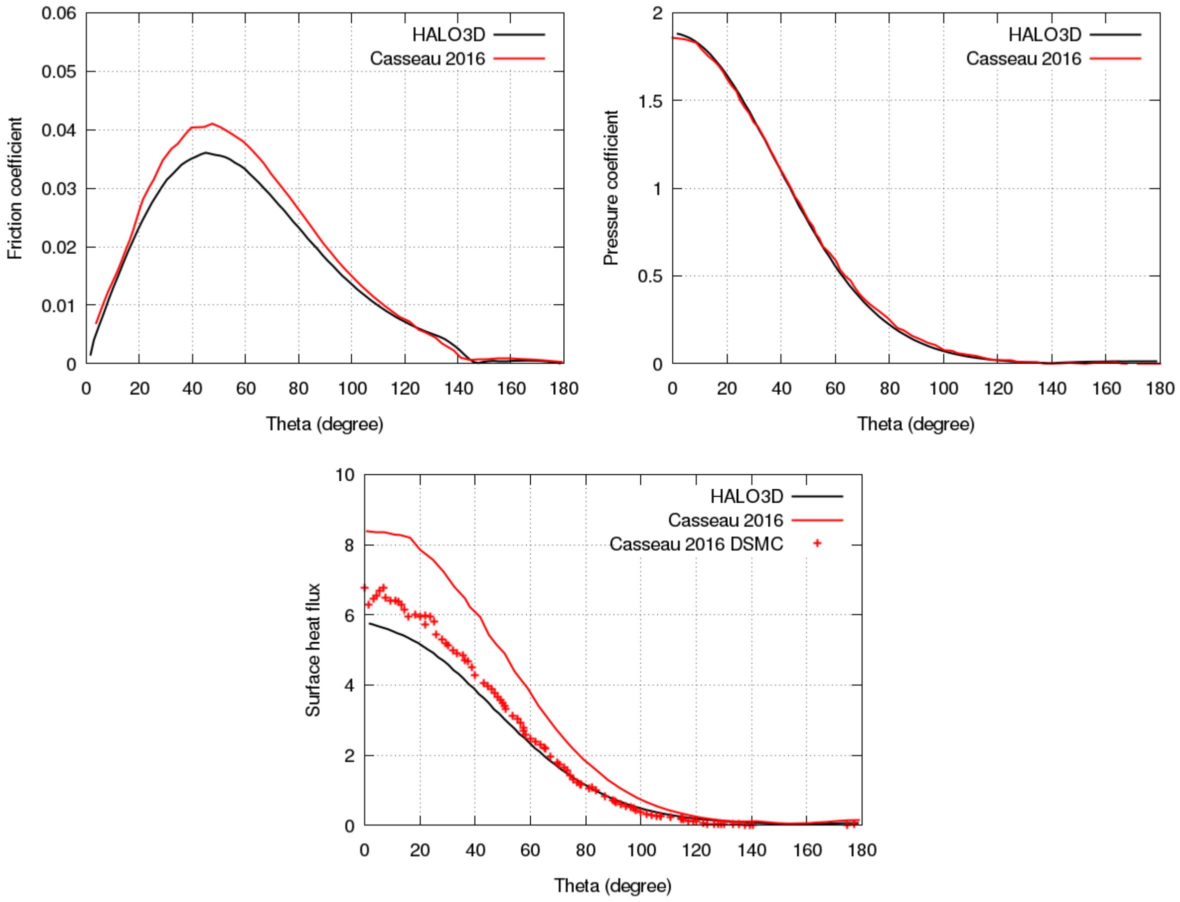

Figure 18. Mach 20 flow past a cylinder (reacting): skin friction coefficient, pressure coefficient, and surface heat flux on the wall.

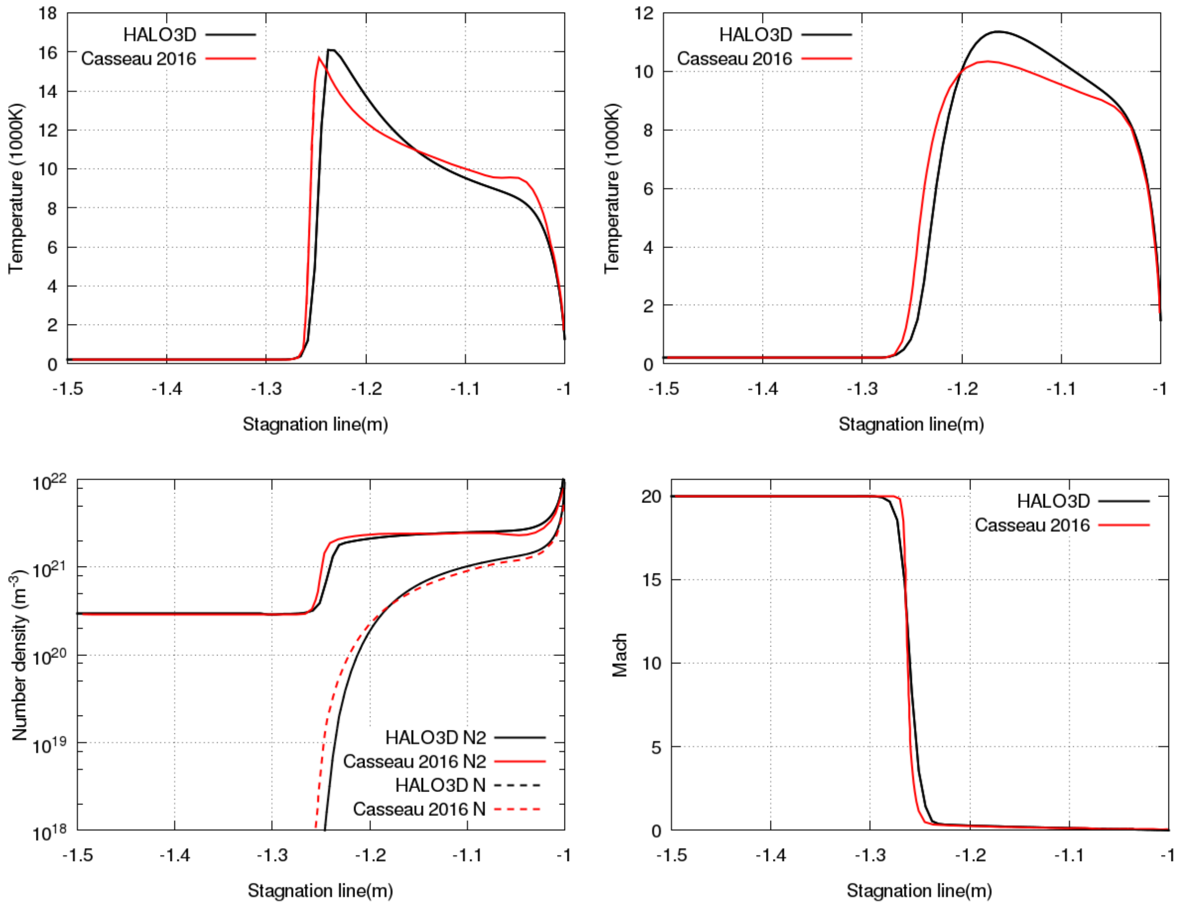

Figure 19. Mach 20 flow past a cylinder (reacting): translational-rotational temperature, vibrational-electronic temperature, N_2 and N density, and Mach number along the stagnation line.

4.4 Introducing the concept of anisotropic mesh optimization

The simulations presented in Sections 4.2 and 4.3, as well as the majority of other hypersonic simulations, are performed on structured grids. Candler [94] remarked that if the grid is poorly aligned with the shock, an artificial vorticity is generated in the post-shock region and transported downstream, eventually affecting the solutions on the wall. Aligning the grid with the shock or multiple shocks is clearly impossible for arbitrary problems for which the locations of the shocks are not known a priori. To overcome this limitation, we use an anisotropic mesh optimization algorithm (OptiGrid, a code developed by the authors and currently commercialized by ANSYS) [95] in conjunction with the flow solver. OptiGrid can be used for both structured [96] and unstructured meshes [97, 98]. It starts with the premise that mesh refinement based on gradients is totally impractical in 3D simply because of the very large grids that result. Hence, some directionality is needed in order to intelligently refine, and at the same time coarsen, when needed.

A better measure of directional refinement is the truncation error which is the true difference between the partial differential equation and its discretized form for linear elements. The truncation error, being proportional to the second derivatives on linear elements, has nine components forming a Hessian, $\boldsymbol{H}$. The eigenvalues of $\boldsymbol{H}$ give a measure of the relative length of an edge, and its eigenvectors specify the ideal orientation of the edge. A posteriori error measure can be expressed in an edge-based fashion as

$$error(\boldsymbol{x}_i - \boldsymbol{x}_j) = \int_0^1 \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{M}(l)(\boldsymbol{x}_i - \boldsymbol{x}_j)}\, dl \tag{4.3}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two endpoints of an edge and $\boldsymbol{M}$ is the absolute value of $\boldsymbol{H}$. In practice, $\boldsymbol{M}$ is approximately computed by a post-processing of the solution on a fixed background mesh. The error is then equally distributed among the edges by resorting to node movement, edge-face-swapping, mesh refinement, and coarsening. The above error estimator may be viewed as a transformation of the mesh from a Cartesian space to a Riemannian space. Uniformly distributing errors tends to yield an isotropic mesh in the Riemannian space, but a highly stretched anisotropic mesh in the Cartesian space that is automatically aligned with any number of shocks. This is because the tetrahedral elements normal to the shock are refined, and parallel to the shock are coarsened.

In OptiGrid, two options are available: optimizing to a specified error level in the solution domain or optimizing to a fixed number of mesh points. In the next test cases, the second option is chosen that keeps the number of mesh points the same as the initial mesh. It will be clearly demonstrated that mesh optimization not only enables a much better accuracy on unstructured meshes *at no increase in mesh size*, but also is perhaps the best and only way that makes possible the use of unstructured meshes to accurately predict aerodynamic and thermal loads of hypersonic vehicles.

4.5 Mach 15.3 Sphere Flow past a Sphere

This test case is a Mach 15.3 laminar flow of air past a sphere in three dimensions. The radius is 0.00635 m. The Reynolds number based on cylinder radius is 14,605. An initial unstructured mesh is shown in Figure 20, consisting of 679,293 nodes, 820,736 tetrahedra, and

1,022,540 prisms. The height of the first layer of elements near the wall is 5 μm. The solution is obtained with the Roe scheme and extended to second order by the van Albada slope limiter. The initial conditions are: Mach number 15.3, velocity 5263 m/s, pressure 664 Pa, both freestream T-R and V-E temperatures are 293 K. An isothermal boundary conditions are set, with both temperatures 1,000 K. The chemical non-equilibrium is modelled by the Park's five-species non-ionizing air model with 17 reactions. Electronic energy modes are not considered.

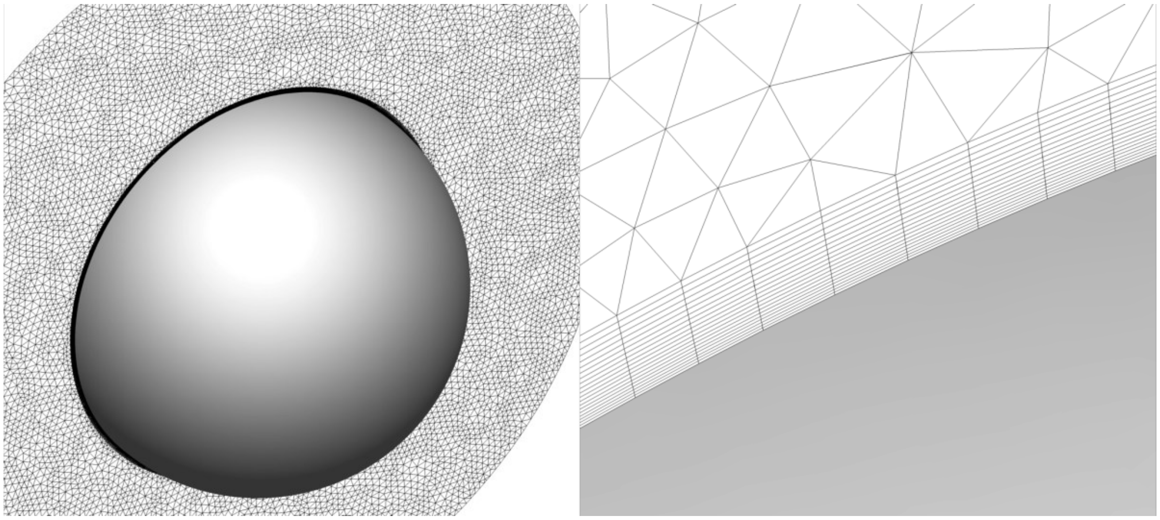

Figure 20. Mach 15.3 non-equilibrium flow over a sphere: hybrid grid (left) and prism layers (right)

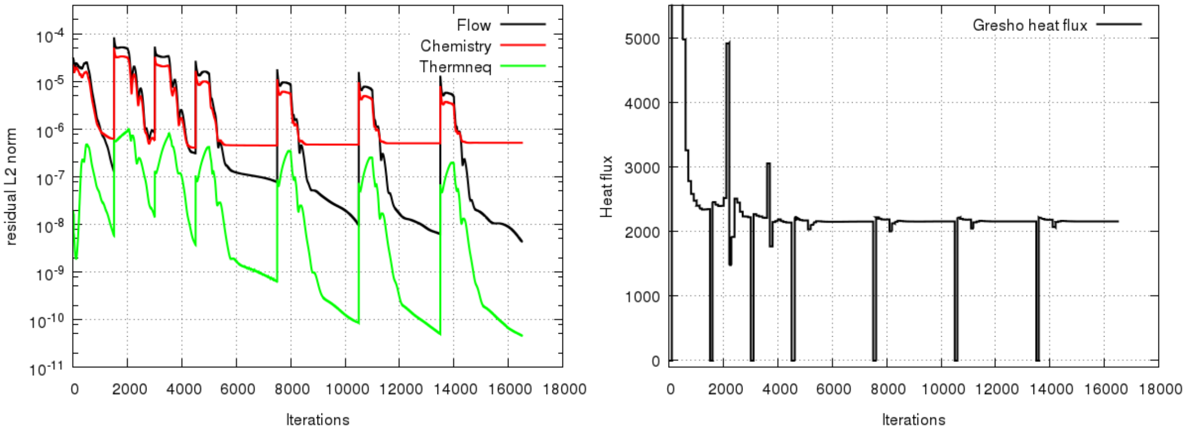

Figure 21. Mach 15.3 non-equilibrium flow over a sphere: convergence curves (left) and integrated heat flux (right)

Six mesh optimization cycles have been performed by OptiGrid using a combination of density, velocity, and T-R temperature to build the error estimator. The targeting number of nodes is 700,000. The final optimized mesh, shown in Figure 22, consists of 667,912 nodes, 1,448,876 tetrahedra, and 798,900 prisms. The initial CFL number is 0.1 and is exponentially increased to 3 within 500 iterations.

Figure 21 (left) shows the convergence history. At the end of each optimization cycle, the solution on the previous mesh is interpolated onto the new optimized mesh to provide a faster restart. This is reflected in the convergence plot as a jump in the residual. The chemistry solver achieves about one to two orders of reduction in the residual and then stalls on the last four meshes. The flow and thermal non-equilibrium solvers achieve about three orders of reduction in the residual on all meshes. Figure 21 (right) shows the convergence of the integrated heat flux. The large zero value spike at the beginning of each restart is non-physical. This is because heat flux is a post-processed value from temperatures, and at each restart, the codes simply read in heat fluxes as zero. It can also be seen that the integrated heat flux reaches convergence on the third optimized mesh.

Comparisons between the initial mesh and the final optimized mesh are shown in Figure 22, where the left halves are results from the initial mesh, and the right halves are those from the final optimized mesh. With mesh optimization, the mesh near the shock is improved by node movements and mesh refinement, and consequently, a much sharper shock is captured. The positive effects of mesh optimization can be also seen in the density and heat fluxes contours, wherein the initial mesh spurious oscillations can be found on the surface of the sphere, but this problem is fixed by mesh optimization. Figure 23 plots the T-R, V-E temperatures, and species mass fractions along the stagnation line. The results from 6th optimized mesh are compared with the ones from CERANS [99]. Good agreements are achieved for all the variables. The results from the initial mesh are very smeared and do not capture the shock in the T-R temperature.

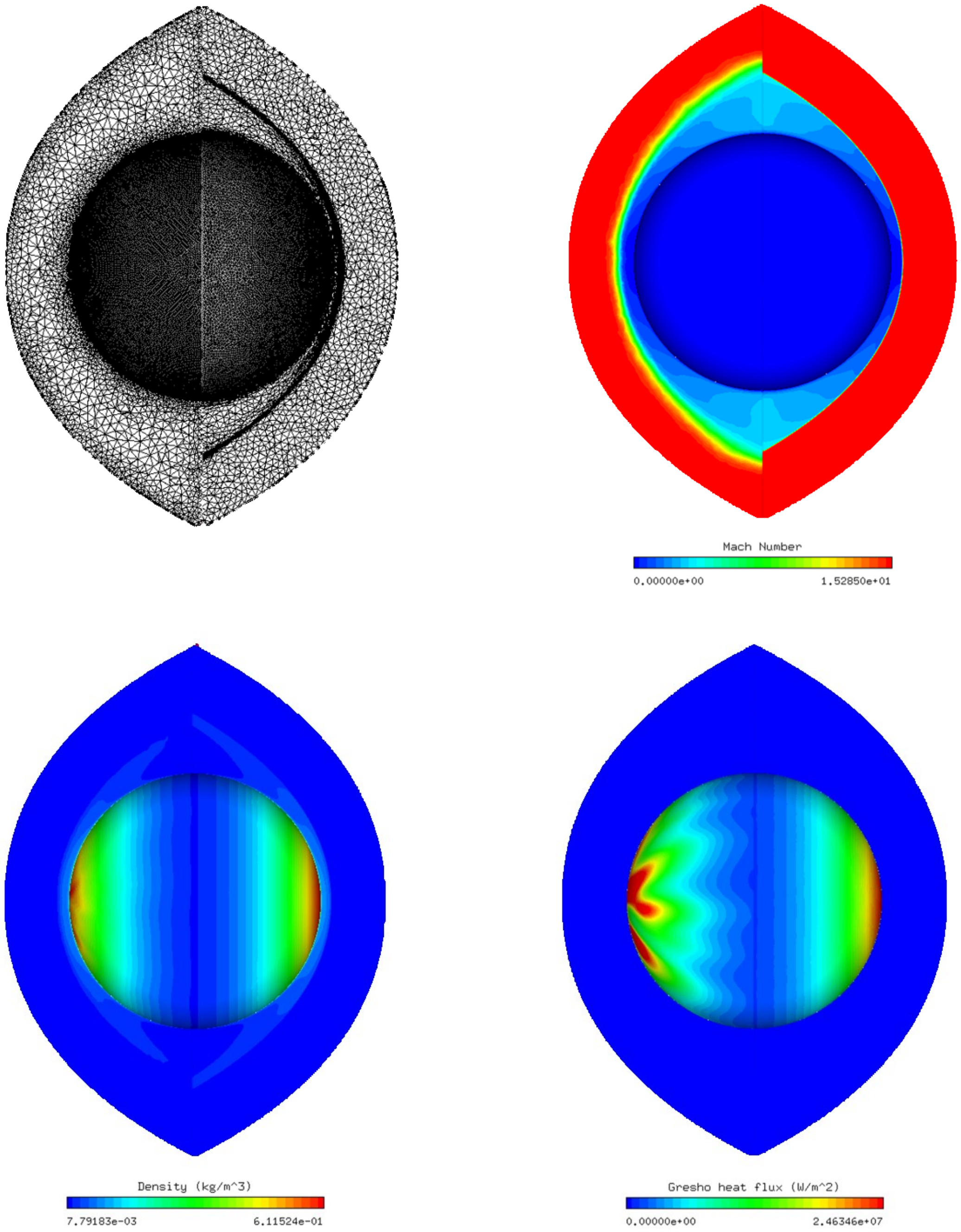

Figure 22. Mach 15.3 non-equilibrium flow over a sphere: mesh (top left), Mach (top right), density (bottom left), and heat fluxes (bottom right) contours, the left halves are results from initial mesh, while the right halves are results from final optimized mesh.

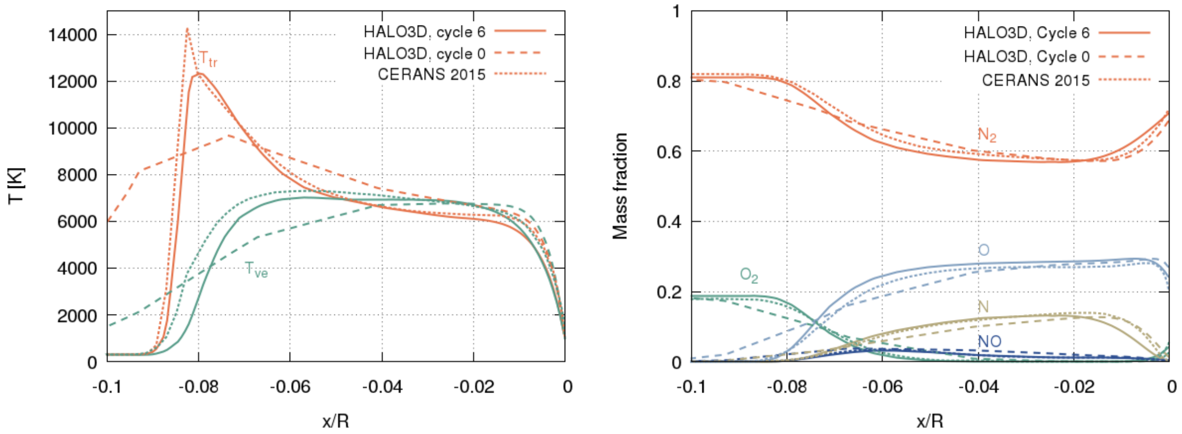

Figure 23. Mach 15.3 non-equilibrium flow over a sphere: translation-rotational temperature, vibrational-electronic temperature, N_2, O_2, N, O, NO, and N mass fraction profiles along the stagnation line.

4.5.1 Convergence Test of Mesh Optimization

To show that the solutions on final optimized meshes are independent of initial meshes, three meshes with an initial number of nodes ranging from 370,000 to 1,400,000, labelled coarse medium and dense are optimized. The target number of nodes is 700,000 for all three test cases. Statistics from Table 1 shows that the number of nodes and elements on the final optimized meshes is roughly the same.

Table 1 Statistics of Mesh Optimizations

	Initial mesh			Final optimized mesh		
	Coarse	Medium	Dense	Coarse	Medium	Dense
Nodes	370,033	679,293	1,390,609	668,082	667,912	666,137
Tetrahedra	429,758	820,736	1,740,956	1,455,510	1,448,876	1,444,920
Prism	562,320	1,022,540	2,082,080	795,340	798,900	795,880

The initial meshes, shown in Figure 24 (top row), are created by adjusting the global element scale factor in ICEM. The final optimized meshes and Mach contours are plotted in the middle and bottom rows of Figure 24, showing statistically insignificant differences between the three cases. The solutions on unstructured grids are compared to those on a structured grid, which has 529,350 nodes, 525,672 elements and the height of the first layer of elements near the wall is 3 μm. The profiles of heat fluxes on the wall and species mass fractions along the stagnation lines are shown in Figure 25. The heat fluxes on the optimized unstructured grids are almost identical. A small kink is found on the structured mesh near the stagnation point, but the same problem is not observed on the unstructured meshes. For this case, the heat fluxes on the structured grid are slightly lower than those on the unstructured grids. The heat fluxes on the initial dense unstructured mesh are neither axisymmetric nor correct. For the species mass fractions along the stagnation line, a good agreement can be found for results on unstructured and structured meshes, except for minor differences near the wall. Figure 26 plots the heat fluxes at the stagnation point versus the mesh optimization cycle. After roughly three mesh optimization cycles, the heat fluxes at the stagnation point on all unstructured meshes converge to a single value. It is also interesting to see that the stagnation point heat flux of the finest initial mesh yielded the most inaccurate value. This demonstrates that adding grid points does not necessarily improve solutions unless the points are added in the correct positions dictated by the governing equations.

The importance of the mesh optimization technique used in this thesis cannot be over-emphasized. In a practical problem, the flow structures, such as shocks, are not known a priori as it is greatly influenced by the flow conditions, gas compositions, and geometries, etc. Generating unstructured grids for such configurations requires expertise and it is considered more an art than a science. Using mesh optimization on unstructured grids keeps the subjectivity of users and the bias of the initial mesh low, as minimum constraint is put on the resolution of flow structures on the initial mesh. Alternatively, the flow structures are automatically resolved during the mesh optimization process by evenly distributing the error on all edges, thus refining the mesh across the shock, while keeping it coarser along the shock. Furthermore, the tremendous potential of unstructured grids in hypersonic flows is fully demonstrated with the help of mesh optimization. The optimized mesh yields essentially better resolved solutions along the stagnation line and more accurate heat flux at the stagnation point with a similar or a smaller number of nodes. In addition, tetrahedral elements are aligned with the shock, thereby producing a clean and sharp shock, as well as axis-symmetric surface heat fluxes for this symmetric problem. Finally, it is demonstrated in the convergence test that for given flow conditions and mesh parameters, mesh optimization statistically yields the same mesh with the same solution no matter what the initial mesh is. Those observations suggest a promising direction towards mesh-independent CFD.

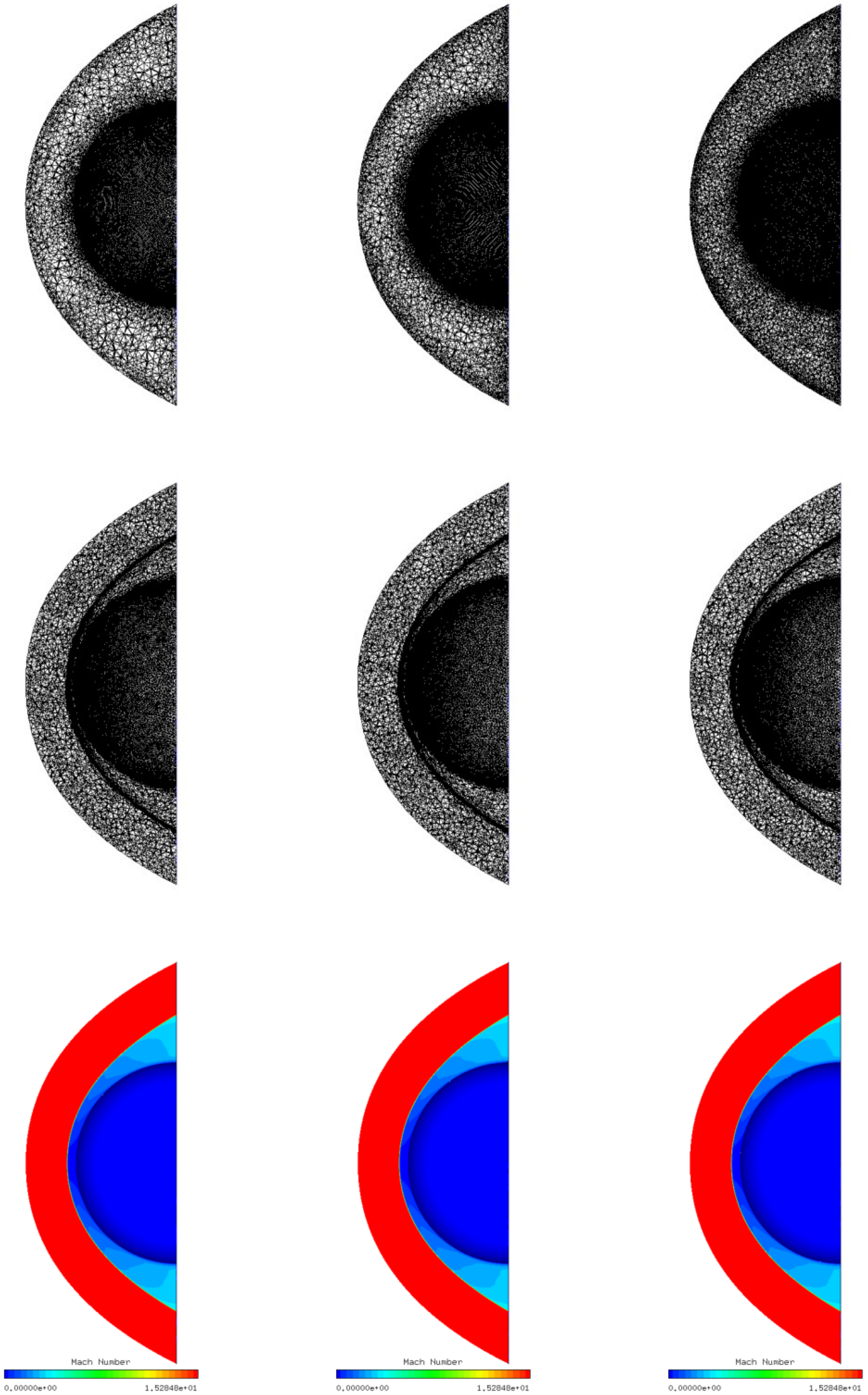

Figure 24. Mach 15.3 non-equilibrium flow over a sphere: initial mesh (top row), final optimized mesh (middle row), Mach contour on final mesh (bottom row). The columns from left to right are results from coarse, medium, and dense meshes.

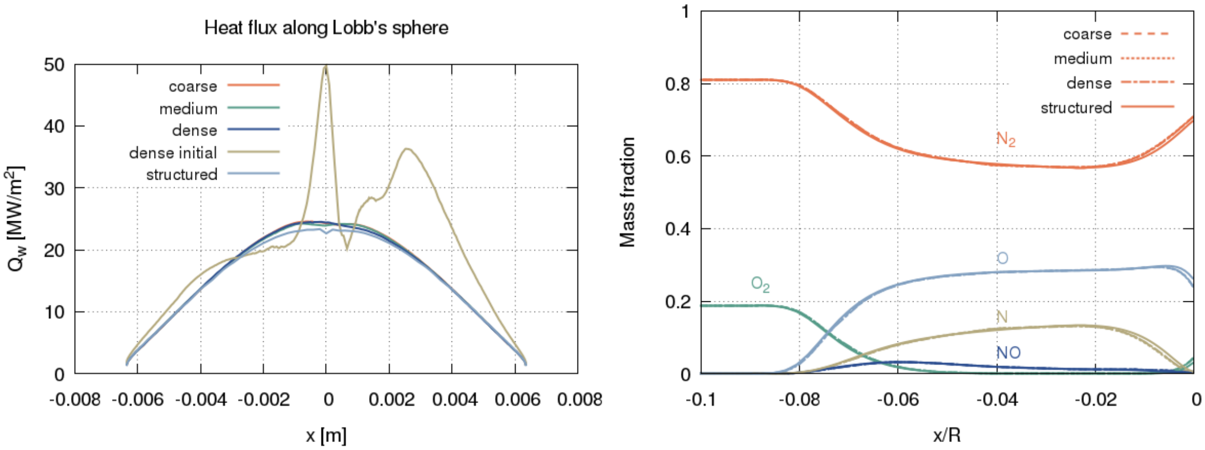

Figure 25. Mach 15.3 non-equilibrium flow over a sphere: heat fluxes on the wall (left) and mass species along the stagnation lines (right).

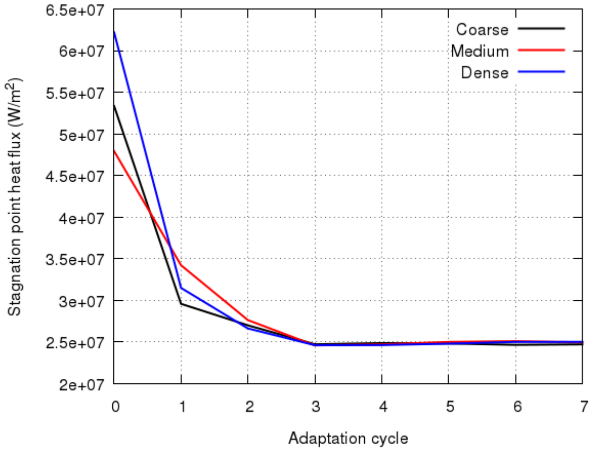

Figure 26. Mach 15.3 non-equilibrium flow over a sphere: heat fluxes at the stagnation point.

5 Jacobian-free Newton-Krylov Method

5.1 Motivation

Solving a large and sparse linear system at each Newton iteration may potentially cause difficulties for thermo-chemical non-equilibrium flows. First, the number of governing equations increases from 5 for thermal equilibrium frozen chemistry flows to $N_s + 6$. Since the dimension of the Jacobian grows quadratically with the number of governing equations, this may cause troubles of storing the Jacobian as the number of species increases. Second, it is difficult to compute the derivatives of MUSCL reconstructed state fluxes with respect to the conservative variables because a larger stencil is required. Third, due to the large size of the Jacobian, a considerable amount of CPU time and computer memory are required to compute and store the linear system. To circumvent some of these above-mentioned difficulties, the Jacobian-free Newton Krylov (JFNK) [58] method is introduced as it does not require the storage of the Jacobian/linear system. Directly applying JFNK to thermo-chemical non-equilibrium flows is nontrivial, therefore, as a starting point, a JFNK solver for **viscous compressible flows** is developed in this section. The main purpose of this section is to thoroughly assess the accuracy, efficiency, robustness, and development difficulty of JFNK, and seeking the potentiality of applying it to thermo-chemical non-equilibrium flows.

5.2 Governing Equations

The equations governing unsteady compressible viscous thermal equilibrium flows with frozen chemistry in conservative form is [100]

$$\frac{\partial Q}{\partial t} + \nabla \cdot (\mathbf{F}^A(Q) - \mathbf{F}^V(Q, \nabla Q)) = 0 \tag{5.1}$$

where Q is the vector of conservative variables, given as

$$Q = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho e \end{Bmatrix} \tag{5.2}$$

where e is the total energy per unit mass defined as the sum of the internal and kinetic energy. $\mathbf{F}^A$ and $\mathbf{F}^V$ are the inviscid and viscous fluxes, respectively. The inviscid fluxes in Cartesian coordinates are

$$F_x^A = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho u\left(e + \frac{p}{\rho}\right) \end{Bmatrix}, \quad F_y^A = \begin{Bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vw \\ \rho v\left(e + \frac{p}{\rho}\right) \end{Bmatrix}, \quad F_z^A = \begin{Bmatrix} \rho w \\ \rho wu \\ \rho wv \\ \rho w^2 + p \\ \rho w\left(e + \frac{p}{\rho}\right) \end{Bmatrix} \tag{5.3}$$

The viscous fluxes are given as

$$F_x^V = \begin{Bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ \boldsymbol{\tau}_x \cdot \boldsymbol{V} \end{Bmatrix}, \quad F_y^V = \begin{Bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ \boldsymbol{\tau}_y \cdot \boldsymbol{V} \end{Bmatrix}, \quad F_z^V = \begin{Bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ \boldsymbol{\tau}_z \cdot \boldsymbol{V} \end{Bmatrix} \tag{5.4}$$

where $\boldsymbol{V}$ is the velocity vector and $\boldsymbol{\tau}_i$ is the stress tensor in i direction. Newton's law for the stress tensor and the Stokes hypothesis are used, as in Eq. (2.5), and a calorically perfect ideal gas is assumed.

5.3 Numerical Formulation

An edge-based Galerkin formulation is used for the spatial discretization. Following the same derivations in Section 2.1

$$\begin{aligned} L_i \frac{dQ_i}{\partial t} + \sum_{j \in K_i} \boldsymbol{\eta}_{ij} \cdot \frac{\mathbf{F}^{A}{}_{i} + \mathbf{F}^{A}{}_{j}}{2} - \sum_{j \in K_i} \boldsymbol{\chi}_{ij} \cdot \frac{\mathbf{F}^{A}{}_{j} - \mathbf{F}^{A}{}_{i}}{2} + \boldsymbol{\xi}_i \cdot \mathbf{F}^A(Q_i) + \sum_{e \in E_i} \int_V \nabla W_i \cdot \mathbf{F}^V dV \\ - \sum_{e \in F_i} \int_A W_i \mathbf{n} \cdot \mathbf{F}^V \, dA = 0 \end{aligned} \tag{5.5}$$

Note that the lump mass matrix and the edge coefficients have been substituted into Eq. (5.5) and that the source term is zero.

5.3.1 Jacobian-free Newton-Krylov Method

The JFNK strategy [58] is introduced by making use of a pseudo-transient continuation method where the original steady problem is transformed into a pseudo-unsteady one. Eq. (5.5) can be rewritten as

$$L \frac{Q^{n+1} - Q^n}{\Delta \tau^n} + R(Q^{n+1}) = 0 \tag{5.6}$$

where L is the lumped mass matrix, R is the residual term of Eq. (5.5), $\Delta\tau$ is the pseudo time step introduced in Section 3.5. At each iteration of the Newton procedure, the solution update is computed as

$$\left[\frac{L}{\Delta\tau^n}\mathbf{I}+\frac{\partial R}{\partial Q}\right]^n \Delta Q = -R(Q^n) \tag{5.7}$$

The above linear system is solved by means of the Flexible Generalized Minimal Residual method (FGMRES) [101]. FGMRES is an iterative solver for nonsymmetric linear systems where the preconditioner can vary at each iteration. The convergence of FGMRES depends on the condition number of the matrix; preconditioning techniques can be enforced to cluster the eigenvalues of the matrix and to improve the convergence of the linear system. Defining

$$A=\left[\frac{L}{\Delta\tau^n}\mathbf{I}+\frac{\partial R}{\partial Q}\right]^n \tag{5.8}$$

The linear system becomes

$$A\Delta Q = -R \tag{5.9}$$

The right preconditioning M is applied as

$$AM^{-1}M\Delta Q = -R \tag{5.10}$$

Defining

$$\Delta P = M\Delta Q \tag{5.11}$$

The original problem is transformed into two sub-problems

$$(AM^{-1})(\Delta P) = -R \qquad \text{and} \qquad \Delta P = M\Delta Q \tag{5.12}$$

The goal of preconditioning is to reduce the condition number of AM^{-1}, thus making the new sub-problems easier to solve. Since FGMRES only accesses the matrix through matrix-vector multiplications, in order to get the solution ΔQ, it is sufficient to provide an operator that computes the product between M^{-1} and an arbitrary vector. Therefore, neither the matrix A nor the preconditioner M is formed explicitly or stored. In the framework of Jacobian-free methods, the product of the preconditioned system matrix with the preconditioned solution is replaced by a Fréchet derivative, i.e.

$$(AM^{-1})(\Delta P) = \left[\frac{L}{\Delta\tau}\mathbf{I}+\frac{\partial R}{\partial Q}\right]^n M^{-1}\Delta P = L\frac{\Delta Q}{\Delta\tau}+\frac{R(Q^n+\varepsilon M^{-1}\Delta P)-R(Q^n)}{\varepsilon} \tag{5.13}$$

where ε is a suitably-chosen small number [25] defined as

$$\varepsilon = error\_rel \times \frac{\sqrt{1 + \|\Delta Q\|}}{\|\Delta Q\|} \tag{5.14}$$

with error_rel being the square root of the machine precision.

Note that the JFNK methods require one evaluation of the residual function R at each FGMRES [101] iteration, while traditional methods that explicitly form the Jacobian require one matrix-vector product per FGMRES iteration. For a problem of millions of unknowns, one evaluation of the residual function typically requires more time than one matrix-vector multiplication, suggesting that overall JFNK might be slower than traditional methods. It is also worth mentioning that the JFNK methods only require the evaluation of R, but not its derivatives, which enables the flexibilities of computing the inviscid contribution to the Jacobian matrix through either the MUSCL-reconstructed state with slope-limiter, or a simple piecewise constant reconstruction of the solutions.

5.3.2 Lower-upper symmetric Gauss-Seidel Preconditioner

To obtain a truly Jacobian-free method, the preconditioning step must also be Jacobian-free. LU-SGS is an efficient iterative solver specifically designed for advection-dominated flows that does not require storing the Jacobian. Following [60], LU-SGS is applied by approximately solving

$$A\Delta Q = -R(Q^n) \tag{5.15}$$

through a forward and backward sweep, namely

$$\begin{aligned} (\mathrm{D} + \mathrm{L})\Delta Q^* &= -R \\ (\mathrm{D} + \mathrm{U})\Delta Q &= \mathrm{D}\Delta Q^* \end{aligned} \tag{5.16}$$

where A is the sum of the pseudo-time contribution and the Jacobian of the approximate residual function

$$A = \left[\frac{L}{\Delta\tau}\mathbf{I} + \frac{\partial R}{\partial Q}\right] \tag{5.17}$$

And D, U, and L are the diagonal, upper, and lower parts of matrix A, respectively.

To compute A cost-effectively, the approximate residual function R is defined after discarding the boundary inviscid fluxes term (third term in Eq. (5.5)), neglecting the boundary contribution of the viscous fluxes (sixth term in Eq. (5.5)) and replacing the inviscid fluxes (second term in Eq. (5.5)) with a simpler approximate one, i.e.

$$R = \underset{\substack{\text{inviscid fluxes}\\ \text{domain edge}\\ R^{A}}}{\sum_{j\in K_i}\boldsymbol{\Phi}_{ij}} + \underset{\substack{\text{viscous fluxes}\\ \text{domain edge}\\ R^{V}}}{\sum_{j\in K_i}\mathbf{F}_{ij}^{V}} + \underset{\substack{\text{inviscid fluxes}\\ \text{boundary conditions}\\ R^{\partial}}}{\boldsymbol{\xi}_i\cdot\mathbf{F}^{A}(Q_i)} \qquad (5.18)$$

where R^A, R^V, and R^∂ are the contributions from inviscid and viscous fluxes for the domain edges and the natural boundary conditions, respectively. The approximate residual function R used in this work is different from Luo's [60]. In his work, R^∂ is ignored and R^V is replaced by a scalar appearing in R^A.

$\boldsymbol{\Phi}_{ij}$ is a simplified version of $\boldsymbol{\Phi}_{num}$, i.e.

$$\boldsymbol{\Phi}_{ij} = \boldsymbol{\Phi}(Q_i, Q_j, \boldsymbol{\eta}_{ij}) = \frac{\mathbf{F}^A(Q_i) + \mathbf{F}^A(Q_j)}{\mathbf{2}}\cdot\boldsymbol{\eta}_{ij} - \frac{|\Lambda_{ij}|}{2}(Q_j - Q_i) \qquad (5.19)$$

where the spectral radius

$$|\Lambda_{ij}| = |\boldsymbol{V}_{ij}\cdot\boldsymbol{\eta}_{ij}| + |\boldsymbol{\eta}_{ij}|a_{ij} \qquad (5.20)$$

is used to approximate the artificial diffusion operator. In Eq. (5.20) the velocity $\boldsymbol{V}_{ij}$ and the sound speed a_{ij} are evaluated at a Roe-averaged state between the two nodes.

Addressing the Jacobian of the inviscid fluxes R^A for the domain edges, Eq. (5.18) is differentiated with respect to Q_i and Q_j, and then the upper matrix U_{ij}^A, the lower matrix L_{ij}^A and the diagonal matrix D_{ij}^A are

$$\mathrm{U}_{ij}^A = \frac{1}{\mathbf{2}}\left[\frac{\partial\mathbf{F}_j^A\cdot\boldsymbol{\eta}_{ij}}{\partial Q_j} - |\Lambda_{ij}|\mathbf{I}\right], \mathrm{L}_{ij}^A = -\frac{1}{\mathbf{2}}\left[\frac{\partial\mathbf{F}_j^A\cdot\boldsymbol{\eta}_{ij}}{\partial Q_j} + |\Lambda_{ij}|\mathbf{I}\right], \mathrm{D}_{ii}^A = \frac{1}{\mathbf{2}}\left[\frac{L_{ii}}{\Delta\tau_i} + \sum_j\frac{|\Lambda_{ij}|}{2}\right]\mathbf{I} \qquad (5.21)$$

Note that the zero-sum property of the edge coefficients for domain nodes has been used and the pseudo-time contribution has been added to the diagonal matrix. As a result, the 5x5 matrix located on the diagonal block is replaced by a scalar, thus making its inversion a trivial operation.

With this strategy, U^A, L^A, and D^A must be computed and explicitly stored, but this can be avoided by introducing a Jacobian-free approximation of the matrix-vector product as

$$\frac{\partial\mathbf{F}_j^A\cdot\boldsymbol{\eta}_{ij}}{\partial Q_j}\Delta Q_j^* = \frac{\mathbf{F}^A(Q_j + \varepsilon\Delta Q_j^*)\cdot\boldsymbol{\eta}_{ij} - \mathbf{F}^A(Q_j)\cdot\boldsymbol{\eta}_{ij}}{\varepsilon} \qquad (5.22)$$

where ε has been defined in Eq. (5.14).

To address the Jacobian of R^V, $\mathbf{F}_{ij}^V$ are discretized with the standard continuous Galerkin approximation and assembled in an edge-based fashion, as in [76]. The upper matrix U_{ij}^V, the lower matrix L_{ij}^V and the diagonal matrix D_{ij}^V are

$$\begin{aligned}
\mathrm{D}_{ii}^V = {} & \mathbf{I}_{5,3} \cdot \left[-\sum_{i<j} \left(A_{ij}^{V,S} + A_{ij}^{V,A} \right) - \sum_{j<i} \left(A_{ji}^{V,S} - A_{ji}^{V,A} \right) \right] \frac{\partial \boldsymbol{V}_i}{\partial Q_i} + \\
& \mathbf{I}_{5,1} \cdot \left[-\sum_{i<j} \boldsymbol{V}_{ij} \cdot \left(A_{ij}^{V,S} + A_{ij}^{V,A} \right) - \sum_{j<i} \boldsymbol{V}_{ij} \cdot \left(A_{ji}^{V,S} - A_{ji}^{V,A} \right) \right] \frac{\partial \boldsymbol{V}_i}{\partial Q_i} \\
\mathrm{U}_{ij}^V = {} & \mathbf{I}_{5,3} \cdot \left(A_{ij}^{V,S} + A_{ij}^{V,A} \right) \frac{\partial \boldsymbol{V}_j}{\partial Q_j} + \mathbf{I}_{5,1} \cdot \boldsymbol{V}_{ij} \cdot \left(A_{ij}^{V,S} + A_{ij}^{V,A} \right) \frac{\partial \boldsymbol{V}_j}{\partial Q_j} \\
\mathrm{L}_{ij}^V = {} & \mathbf{I}_{5,3} \cdot \left(A_{ji}^{V,S} + A_{ji}^{V,A} \right) \frac{\partial \boldsymbol{V}_j}{\partial Q_j} + \mathbf{I}_{5,1} \cdot \boldsymbol{V}_{ij} \cdot \left(A_{ji}^{V,S} + A_{ji}^{V,A} \right) \frac{\partial \boldsymbol{V}_j}{\partial Q_j}
\end{aligned} \tag{5.23}$$

where "*S*" and "*A*" superscripts indicate the symmetric and anti-symmetric parts of Eq. (3.9), i.e.

$$A_{ij}^{V,S} = \mu_{ij} \mathrm{tr}\left(\mathbf{d}_{ij}\right) \mathbf{I} + \left(\mu_{ij} + \lambda_{ij}\right) \mathbf{d}_{ij}^s, \quad A_{ij}^{V,A} = \left(\lambda_{ij} - \mu_{ij}\right) \mathbf{d}_{ij}^A \tag{5.24}$$

The transformation matrices are

$$\mathbf{I}_{5,3} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T, \quad \mathbf{I}_{5,1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \tag{5.25}$$

The Jacobian of R^∂ only contributes to the diagonal matrix, i.e.

$$D_{ii}^\partial = \frac{\partial \mathbf{F}_i^{A,\partial} \cdot \boldsymbol{\xi}_i}{\partial Q_i} \tag{5.26}$$

where the expression of $\mathbf{F}_i^{A,\partial}$ depends on the type of boundary conditions, which has been illustrated in Section 3.6.1. D^∂ can be combined with D^A for further simplification

$$\begin{aligned}
\mathrm{D}_{ii}^{A+\partial} & = \frac{L_{ii}}{\Delta\tau_i} \mathbf{I} + \frac{\partial}{\partial Q_i} \left[\mathbf{F}_i^{A,\partial} \cdot \boldsymbol{\xi}_i + \frac{1}{2} \sum_{j \in K_i} \left(\mathbf{F}^A(Q_i) \cdot \boldsymbol{\eta}_{ij} + \left|\Lambda_{ij}\right| Q_i \right) \right] \\
& = \left[\frac{L_{ii}}{\Delta\tau_i} + \sum_{j \in K_i} \frac{\left|\Lambda_{ij}\right|}{2} \right] \mathbf{I} + \frac{\partial}{\partial Q_i} \left[\left(\mathbf{F}_i^{A,\partial} - \frac{\mathbf{F}^A(Q_i)}{2} \right) \cdot \boldsymbol{\xi}_i + \sum_{j \in K_i} \left(\mathbf{F}^A(Q_i) \cdot \left(\frac{\boldsymbol{\eta}_{ij}}{\mathbf{2}} + \frac{\boldsymbol{\xi}_i}{\mathbf{2}} \right) \right) \right] \\
& = \left[\frac{L_{ii}}{\Delta\tau_i} + \sum_{j \in K_i} \frac{\left|\Lambda_{ij}\right|}{2} \right] \mathbf{I} + \frac{\partial}{\partial Q_i} \left[\left(\mathbf{F}_i^{A,\partial} - \frac{\mathbf{F}^A(Q_i)}{2} \right) \cdot \boldsymbol{\xi}_i \right]
\end{aligned} \tag{5.27}$$

where the zero-sum property of edge coefficients $\sum_{j\in K_i}\left(\frac{\eta_{ij}}{2}+\frac{\xi_i}{2}\right)$ =0 has been applied. An immediate benefit is that the last term of $D_{ii}^{A+\partial}$ is independent of the node j.

Finally, the LU-SGS sweeps in Eq. (III.12) can be written as

$$\begin{gathered}\left(\mathrm{D}^{A+\partial}+\mathrm{D}^{V}+\mathrm{L}^{A}+\mathrm{L}^{V}\right)\Delta Q^{*}=-R\\ \left(\mathrm{D}^{A+\partial}+\mathrm{D}^{V}+\mathrm{U}^{A}+\mathrm{U}^{V}\right)\Delta Q=\left(\mathrm{D}^{A+\partial}+\mathrm{D}^{V}\right)\Delta Q^{*}\end{gathered}\tag{5.28}$$

The velocity components on the wall are zero for the non-slip wall boundary conditions, however, the momentum contributions on the walls must be reset to zero after each sweep since the LU-SGS procedure may not automatically satisfy the condition. A remarkable feature of LU-SGS is that the storage of the Jacobian is not required. In this chapter, the PETSc [56] implementation of FGMRES is used, while an in-house version of LU-SGS has been implemented within the PETSc framework.

5.4 Results

The proposed numerical method has been validated over a representative range of Mach numbers, for 3D test cases, from subsonic flows to supersonic flows. Since this thesis focuses on hypersonic flows, only results from supersonic flows are presented, results from subsonic and transonic flows can be found in [102]. The test cases in this section are the Mach 1.93 laminar flow past a sphere, the Mach 4 turbulent flow past a waverider and the Mach 10.01 laminar flow past a sphere.

With the present JFNK implementation, the inviscid contribution to the Jacobian matrix can be computed either using a simple piecewise-constant reconstruction of the solution, or a MUSCL reconstructed state with slope-limiter. The first one is subsequently referred to as 1st order Jacobian (JFNK-1), and the second as 2nd order Jacobian (JFNK-2). To assess the performance of the JFNK solvers, the traditional method of explicitly forming the Jacobian and using block-Jacobi with ILU preconditioning is also tested. This is subsequently referred to as the explicit Jacobian method. The explicit Jacobian method uses a simple piecewise-constant reconstruction of the solution to compute the Jacobian. Computing the explicit Jacobian using the MUSCL reconstructed states and the slope-limiter is difficult because a larger stencil is required to compute their derivatives. However, JFNK does not have such a problem because its Jacobian is numerically approximated, and the derivatives are not needed.

Unless otherwise specified, all the results shown in the following sections are obtained with JFNK-2. FGMRES convergence is achieved when the relative drop in the preconditioned residual norm is below a specified tolerance of 10^{-4}. The size of the Krylov space is 20. For the first test case, FGMRES stops when the number of FGMRES iterations exceeds 20 and the code proceeds to the next Newton step with no FGMRES restart. For the other two test cases, the maximum number of FGMRES iterations is 40 with FGMRES restarting every 20 iterations. Inviscid fluxes are computed by the AUSM+-up [103, 104] or Roe scheme with the van Albada slope limiter. The ILU preconditioner remains fixed at each FGMRES iteration, in which case the results of FGMRES are identical to those of GMRES.

5.4.1 Mach 1.93 Viscous Flow past a Sphere

This example presents the Mach 1.93 viscous flow around a sphere. The sphere radius R is 7.5 mm, the freestream temperature and pressure are 294 K and 540 Pa, respectively. The Reynolds number based on the sphere radius is 1750, which is small enough to assume laminar flow. The computational grid, shown in Figure 27, consists of 300,993 nodes, 281,484 tetrahedra, and 487,680 prisms. The surface of the sphere is represented by 8,128 triangles. The near-wall region is represented by 60 layers of prisms and is approximately 3.9R thick. The initial CFL value of 10^{-1} is exponentially increased to 100 in 500 iterations.

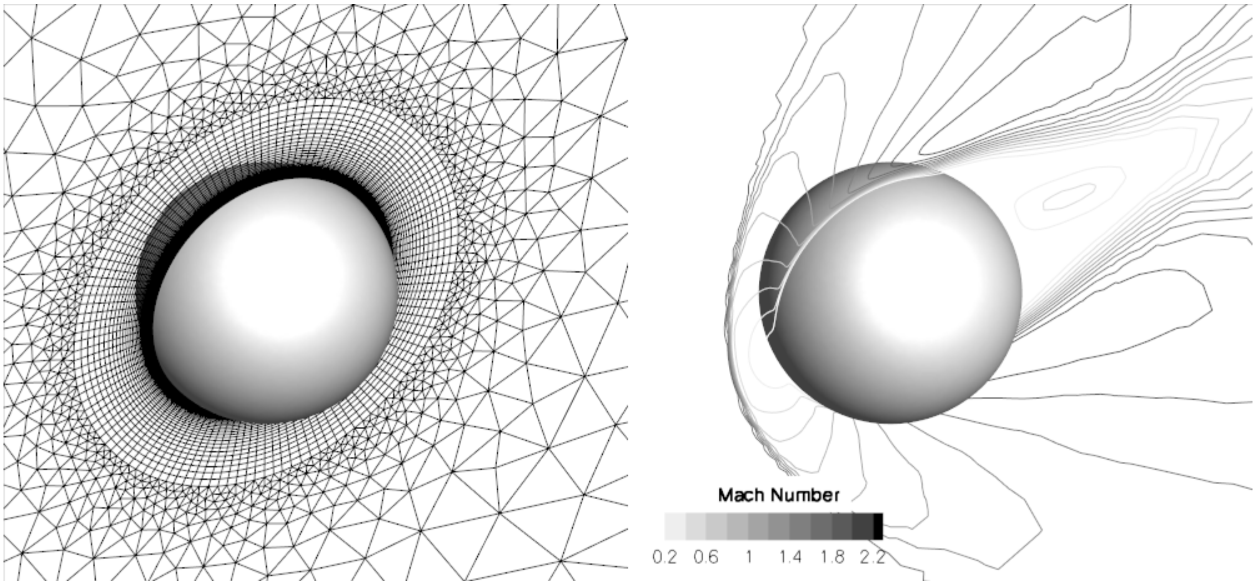

Figure 27. Mach 1.93 viscous flow past a sphere: hybrid grid (left) and Mach number contours (right)

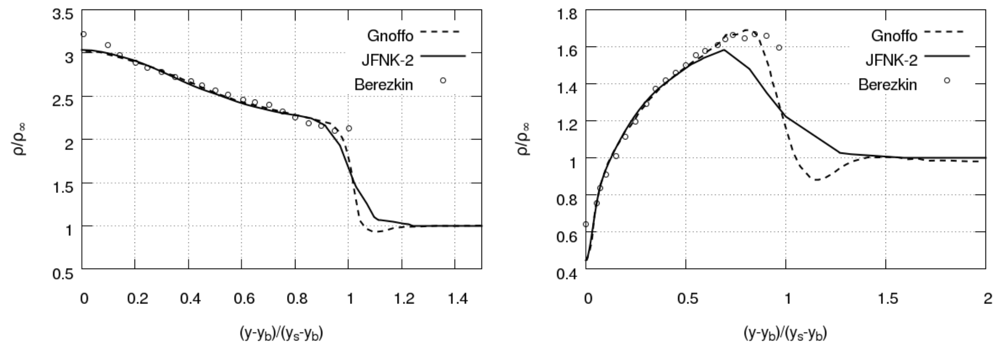

Figure 28. Mach 1.93 viscous flow past a sphere: non-dimensional density along the line normal to the axis in front of the sphere (left) and along the line normal to the sphere (right)

Mach contours are shown in Figure 27. A bow shock appears in front of the sphere. The non-dimensional density is plotted in Figure 28 along the crosswind direction at two different locations: at the nose, $x = R$, and at the mid-section, $x = 0$. The crosswind coordinate is scaled with respect to the distance between the shock location, y_s, and the boundary location, y_b. Since neither of the references provides a value for it, y_s is measured from the solution: $y_s(x = 0) = 2.743R$ and $y_s(x = R) = 1.32R$. The agreement between the JFNK solution, the results of Gnoffo and the experiments is fairly good [105]. Both the present computation and the result by Gnoffo under predict the value of the density at the stagnation point.

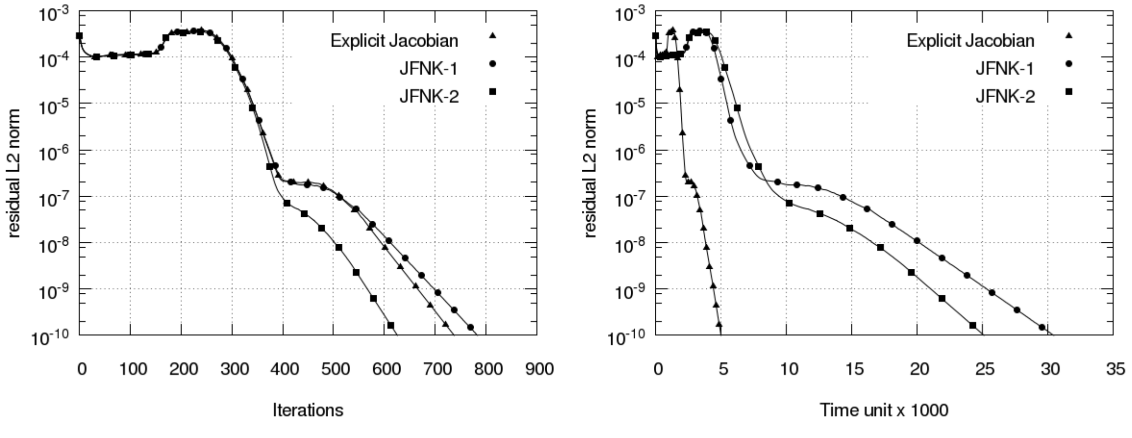

Figure 29. Mach 1.93 viscous flow past a sphere: convergence history as a function of the number of iterations (left) and time (right)

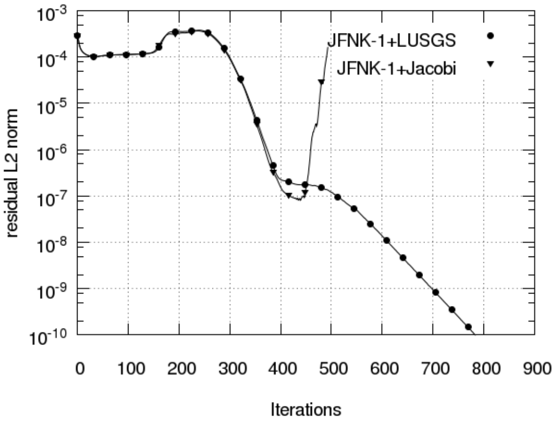

Figure 30. Mach 1.93 viscous flow past a sphere: convergence history of the LU-SGS and Jacobi preconditioners

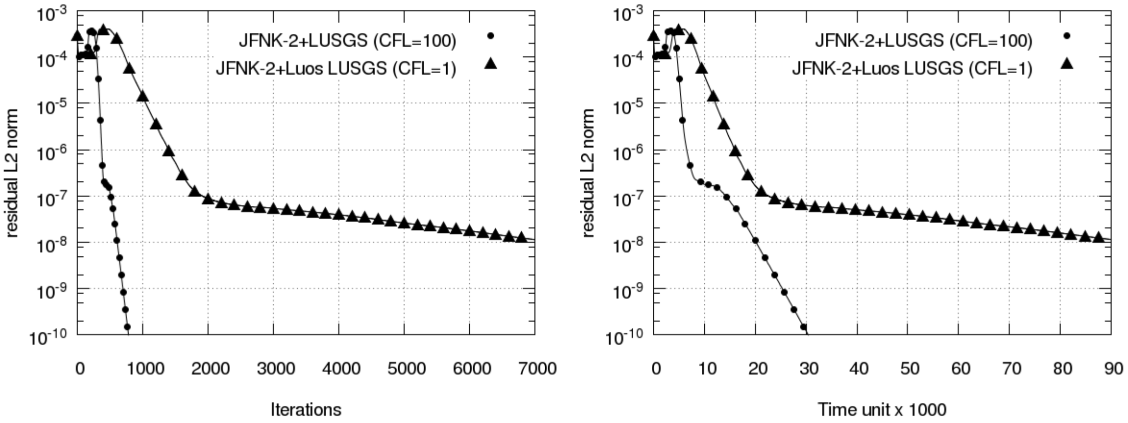

Figure 31. Mach 1.93 viscous flow past a sphere: convergence history of the LU-SGS and Luo's LU-SGS preconditioners as a function of the number of iterations (left) and time (right)

Figure 29 shows the convergence history for the explicit Jacobian, JFNK-1, and JFNK-2. Time is defined as the average time per FGMRES iteration for JFNK-2, which is 0.48s for this test case. All the residuals converge to 10^{-10}. The advantage of JFNK-2 is in the number of Newton iterations, only 626. This is expected because JFNK-2 includes the MUSCL reconstructed states and the slope-limiter in the approximate Jacobian. In terms of wall time, JFNK-1 and JFNK-2 take about the same time. Both are about 6 times slower than the explicit Jacobian method. In terms of the maximum memory storage, JFNK-2 (1.48 GB) achieves a 54% reduction over explicit Jacobian (3.18 GB). All simulations were run in parallel on 4 processors.

Figure 30 shows the comparison between JFNK-1 with LU-SGS and JFNK-1 with the Jacobi preconditioner. In conjunction with the LU-SGS preconditioner, FGMRES convergence is achieved when the relative drop in the preconditioned residual norm is below the specified tolerance of 10^{-2} or a maximum number of FGMRES iterations 20, whereas for the Jacobi preconditioner the tolerance was set at 10^{-6} and the maximum number of FGMRES iterations was set to 200. In general, Jacobi is a less effective preconditioner than LU-SGS and requires more linear iterations to converge. The size of the Krylov space is 20 for both methods. Although more FGMRES iterations are allowed in the case of the Jacobi preconditioner, it still fails at around 500 Newton iterations. This suggests that the Jacobi preconditioner is, for this case, less robust than LU-SGS. Note that the chosen maximum number of linear iterations is 10 times greater for Jacobi but is not sufficient to prevent numerical instabilities. Figure 31 shows the comparison between JFNK-2 with LU-SGS and original LU-SGS preconditioner [60]. The maximum CFL is 100 for LU-SGS and 1 for original LU-SGS. The cases of maximum CFL 100 and 10 have been tested for original LU-SGS, but both fail due to negative temperature, which suggests that robustness increases when the contributions from the boundaries are included and more accurate viscous fluxes are used.

An analysis of the effects of the relative tolerance level and the maximum number of FGMRES iterations (ksp_max_it) was carried out. Two relative tolerance levels were considered, 10^{-2} and 10^{-6}. The maximum number of FGMRES iterations was set to 5, 10, 40, 80, 120, 160 and 200. Figure 32 displays the results obtained with a relative tolerance of 10^{-2} and different values of the maximum number of FGMRES iterations for the explicit Jacobian method (top row), JFNK-1 (middle row) and JFNK-2 (bottom row). The left column shows the convergence history in terms of Newton iterations. The middle column shows the convergence history in terms of time units. The right column shows the number of FGMRES iterations in terms of Newton iterations. The parameters in Figure 33 are the same as those in Figure 32, except that the relative tolerance level is set to 10^{-6}. A clear trend can be seen where the wall time typically increases with the maximum number of FGMRES iterations. However, there are exceptions with the maximum number of FGMRES iterations of 5 and 10.

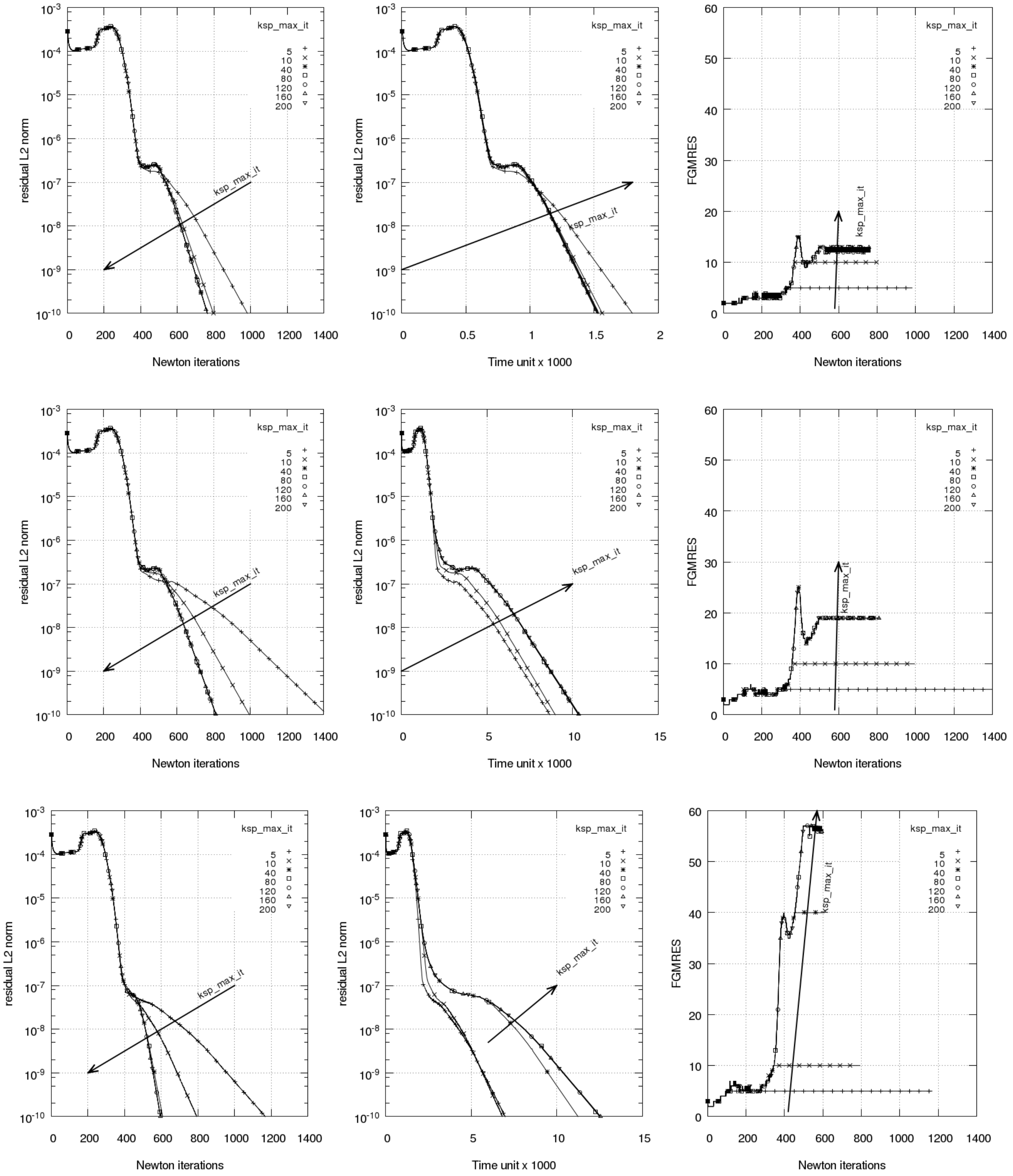

Figure 32. Mach 1.93 viscous flow past a sphere: behaviour of FGMRES for the explicit Jacobian method (top), 1st order JFNK (middle), and 2nd order JFNK (bottom) with FGMRES relative tolerance of 10^{-2}

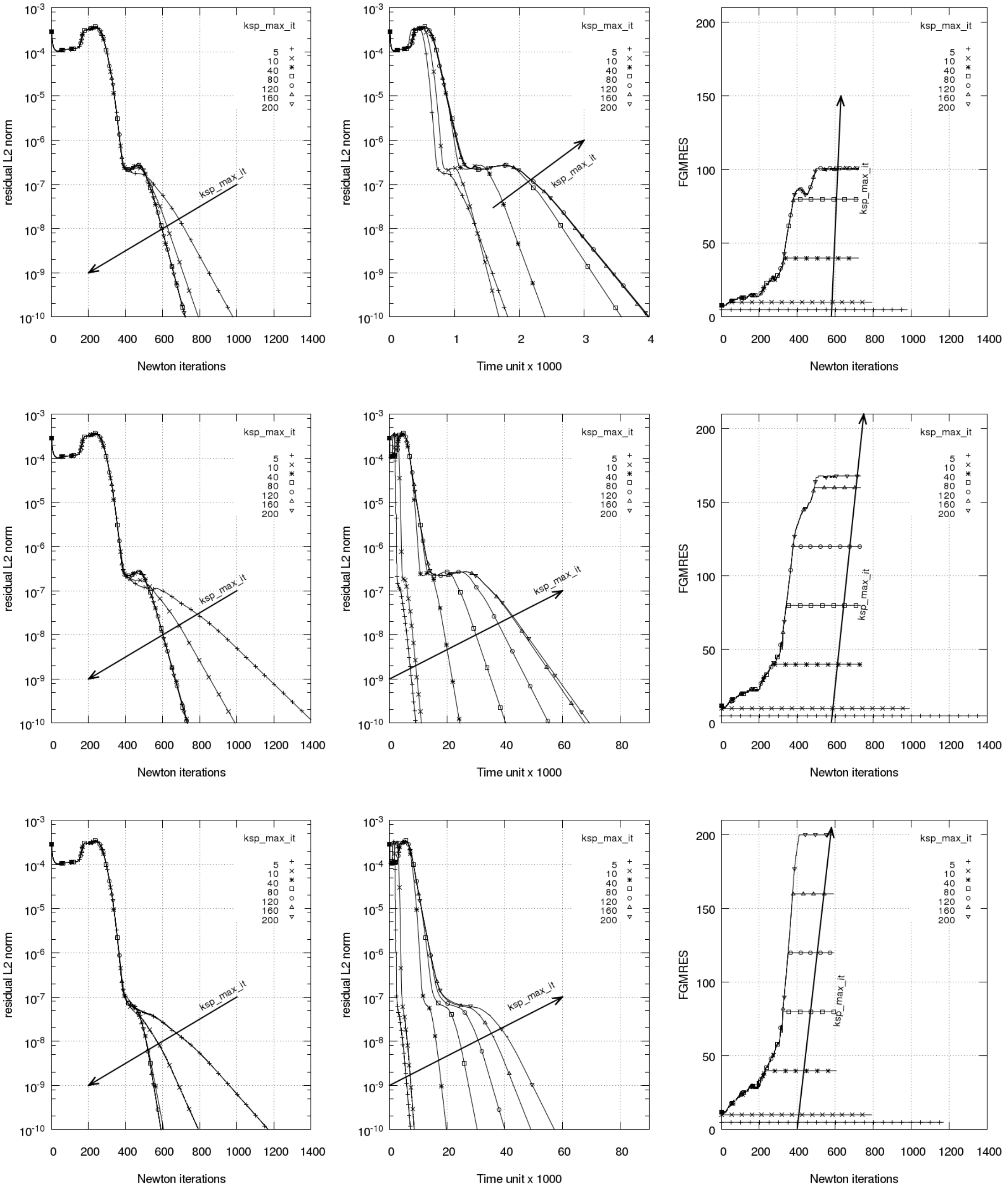

Figure 33. Mach 1.93 viscous flow past a sphere: behaviour of FGMRES for the explicit Jacobian method (top), 1st order JFNK (middle), and 2nd order JFNK (bottom) with FGMRES relative tolerance of 10^{-6}

The second graph in the first row of Figure 32 shows that the explicit Jacobian takes the longest time with FGMRES iterations of 5 and 10. The second graph in the third row of Figure 32 shows that JFNK-2 is slower with a maximum number of FGMRES iterations of 5 compared to 10. The second graph in the first row of Figure 33 shows that with lower relative tolerance levels the explicit Jacobian is faster with the maximum number of FGMRES iterations of 5 and 10, however, 10 is faster than 5. The above exceptions suggest that a maximum number of FGMRES iterations of 5 or 10 is too small to ensure an appropriate convergence of the linear system.

It is also interesting to see that in the first columns of Figure 32 and Figure 33 that values of maximum number of FGMRES iterations greater than 40 do not diminish the number of Newton iterations required for convergence, but instead increase the number of FGMRES iterations (third column) and consequently dramatically increase the wall time (second column). This analysis shows that to achieve an optimal computational time, it is desirable to put a limit on the maximum number of FGMRES iterations and the relative tolerance in the convergence of the non-preconditioned residual norm.

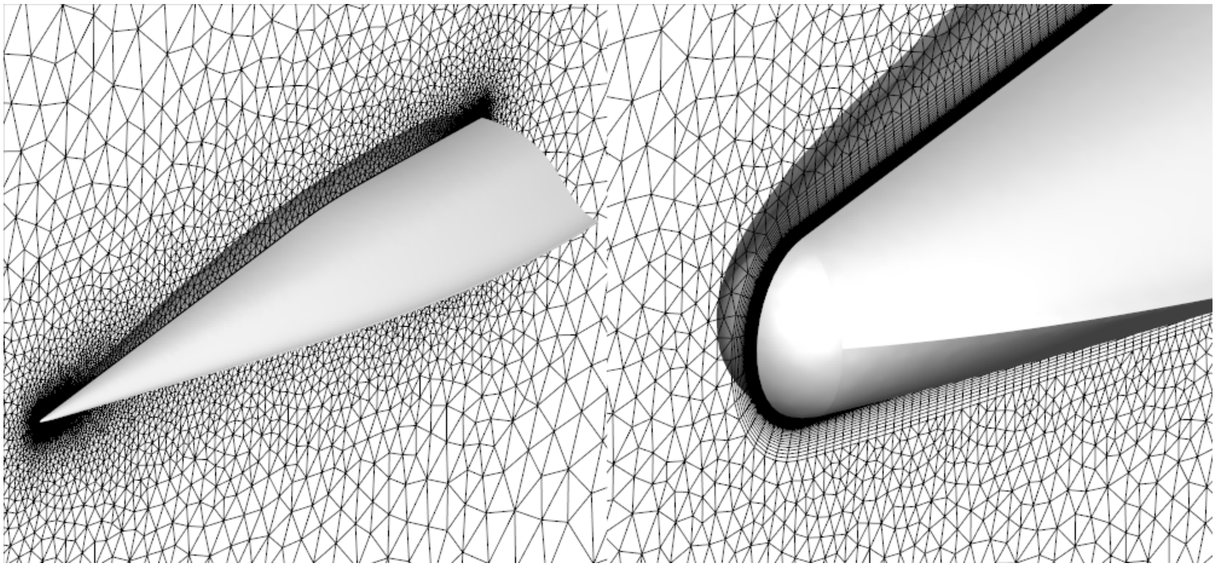

Figure 34. Mach 4.0 Viscous Flow past a Waverider: hybrid grid

5.4.2 Mach 4.0 Viscous Flow past a Waverider

This example presents a Mach 4 viscous flow around the waverider geometry shown in Figure 34. The waverider's length is 3 m, the nose radius 5 mm, and the free-stream temperature and pressure 279.75 K and 8.42×10^4 Pa, respectively. The Reynolds number based on waverider length is 2.41×10^8, with an angle of attack of 1 degree. Due to the high Reynolds number, the Spalart-Allmaras turbulence [106] model is used. An adiabatic wall boundary condition specified. The computational grid consists of 4,392,286 nodes, 1,441,235 tetrahedra, and 8,193,459 prisms, while the surface of the waverider is discretized with 151,766 triangles. The initial CFL of 10^{-2} is exponentially increased to 10 in 1000 iterations. All runs take 40 FGMRES iterations at every Newton step.

The Mach and density contours are shown in Figure 35. A detached bow shock is in front of the waverider and a separation zone forms behind it. The speed-up diagram for 16, 32, 64, and 128 processors shown in Figure 36 highlights an 88% parallel performance on 128 processors. The performance reflects the increase of communication cost as the number of processors increase.

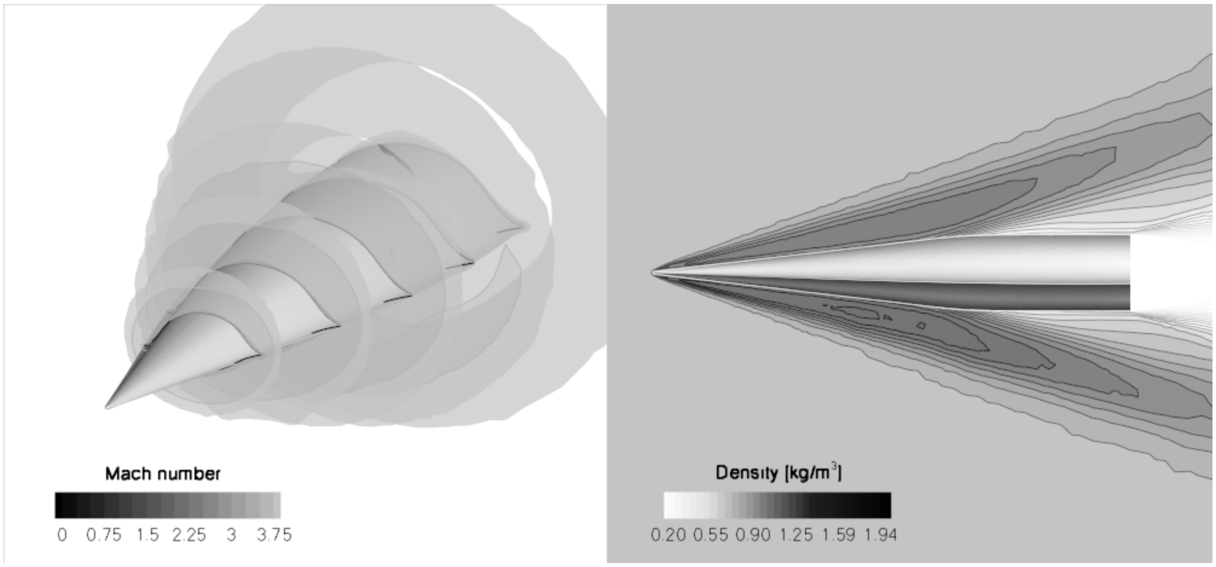

Figure 35. Mach 4 viscous flow past a Waverider: Mach number contours (left) and density contours (right)

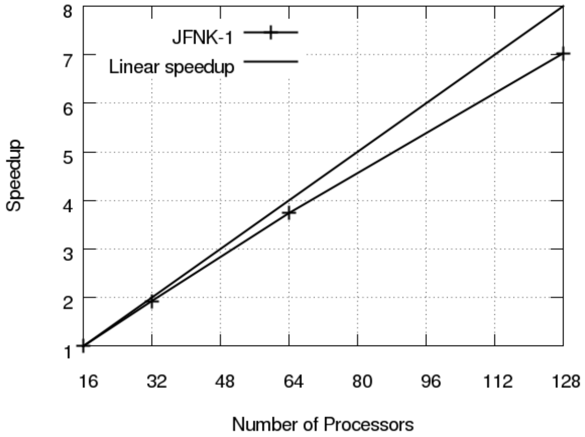

Figure 36. Mach 4 viscous flow past a Waverider: speed-up diagram

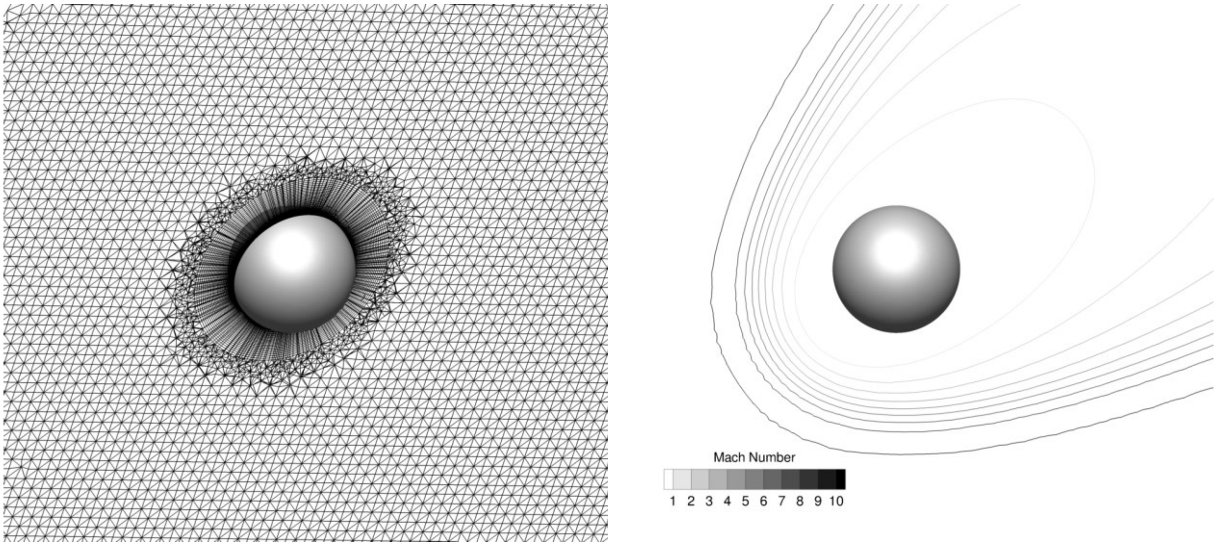

Figure 37. Mach 10.01 viscous flow past a sphere: hybrid grid (left) and Mach number (right)

5.4.3 Mach 10.01 Viscous Flow past a Sphere

This example is the Mach 10.01 viscous flow of Nitrogen around a sphere of radius 0.1524 m. The free stream temperature and pressure are 200 K and 0.0468 Pa, respectively. The Reynolds number based on the radius of the sphere is 26.99, which is small enough to assume a laminar flow. The computational grid, shown in Figure 37 (left), consists of 2,995,100 nodes, 16,477,103 tetrahedra and 380,800 prisms and the surface of the sphere is discretized with 234,952 triangles. The near wall region contains 40 layers of prisms and it is approximately 0.683 radii thick. The initial CFL is 10^{-5}, exponentially increasing to 10 in 3,000 iterations. The stabilization scheme is Roe's with the van Albada slope limiter. The contours of the Mach number are shown in Figure 37 (right). As expected, at such low Reynolds number the detached bow shock in front of the sphere is significantly smeared by the viscous terms.

Figure 38 shows the convergence curves for explicit Jacobian, JFNK-1, and JFNK-2. The time unit is defined as the average time per FGMRES iteration for JFNK-2 and it is 2.7 s for this test case. In terms of the number of Newton iterations, all solvers converge in approximately 3,200 Newton iterations. In terms of wall time, JFNK-1 is about 6 times slower than explicit Jacobian and JFNK-2 is about 4 times slower than explicit Jacobian.

In terms of maximum memory, JFNK-2 (17.18Gb) achieves a 45% reduction over the explicit Jacobian method (31.64Gb). The maximum memory consumptions of JFNK-1 (17.20Gb) and JFNK-2 (17.18Gb) are nearly the same. All simulations were carried out in parallel on 48 processors.

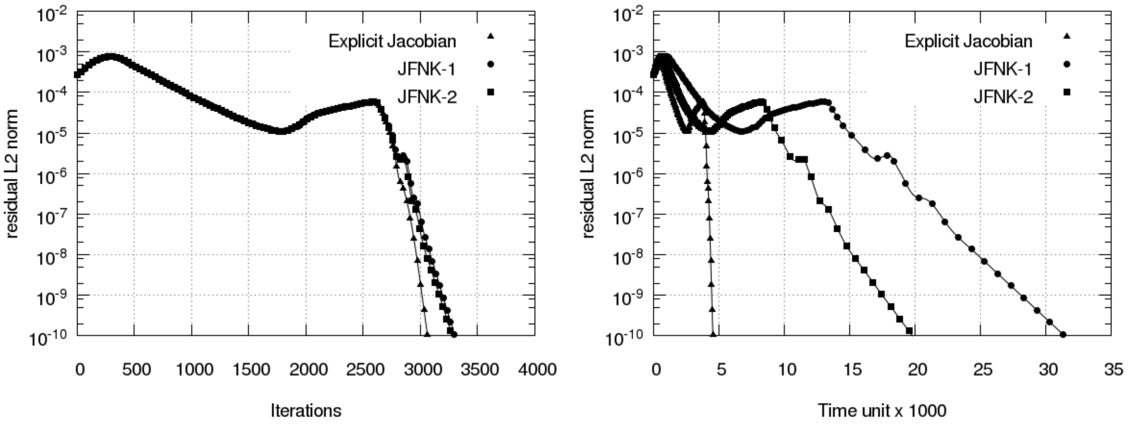

Figure 38. Mach 10.01 viscous flow past a sphere: convergence history as a function of the number of iterations (left) or time (right)

5.5 Remarks of JFNK

In this section, an accurate edge-based Jacobian-free FE solver has been developed to address high-Mach viscous flows. Three-dimensional test cases are presented, with good agreement with the references. The performance of the Jacobian-free solver is assessed. Improved LU-SGS is found to be more robust and efficient than the Jacobi preconditioner and the original LU-SGS.

The proposed Jacobian-free approach has many advantages: first, the MUSCL reconstructed and slope-limited states are automatically included in the linearization of the residual. This benefit is crucial for solving thermo-chemical non-equilibrium flows, where the transport coefficients and reaction rate coefficients are functions of the primitive variables. Second, the Jacobian-free method introduces a general framework that allows the introduction of an arbitrary number of reactions/chemical species and non-equilibrium effects in an efficient manner, since the method only needs the evaluation of the residual and does not require an analytical expression of its derivative. Third, Jacobian-free solvers are more memory-efficient, with savings of 50% observed for chemical-thermal equilibrium flow. When more complex physical phenomena are introduced, and the number of chemical species and governing equations is substantially increased, much higher memory savings are anticipated.

Despite its many advantages, the proposed Jacobian-free approach also has some disadvantages, perhaps the most remarkable of which is the high computational cost. The Jacobian-free approach is generally 4-6 times slower than the explicit Jacobian method. In practice, it is also found that JFNK is sometimes less robust than the explicit Jacobian method when the same simulation is carried out. Another disadvantage is that although this approach is nominally called Jacobian-free, it is really not. When calculating the LU-SGS preconditioner, a simpler version of Jacobian is still assembled on-the-fly and then discarded. This has three consequences. First, a simpler Jacobian means the computation of Jacobian in JFNK is much easier and faster than that in the explicit Jacobian method. Second, since the Jacobian cannot be stored, it is computed on-the-fly repeatedly, resulting in a waste of computation resources. Last, the difficulties of calculating the Jacobians of fluxes still exist in LU-SGS.

In the author's opinion, the Jacobian-free approach is an elegant solution for solving a system of equations. For the future development of a Jacobian-free hypersonic flow solver, it is recommended that developing an explicit Jacobian solver first and then converting that solver to a Jacobian-free solver. The conversion is straightforward since the explicit assembled Jacobian can be used as a preconditioner in the Jacobian-free solver. After that, a matrix-free preconditioner should be developed to replace the explicitly assembled Jacobian.

6 Conclusion and Future Development

This thesis is part of a large and sustained effort to develop an edge-based FEM hypersonic flow solver capable of simulating non-equilibrium effects, turbulence, magnetic interaction, and more. The present work addresses a robust and accurate loosely-coupled solver for high-Mach thermo-chemical non-equilibrium flows. The finite rate chemistry and two-temperature models are implemented to account for the non-equilibrium processes. The employed edge-based strategy yields a stable FE formulation with the numerical fluxes computed by the Roe scheme. The flow, chemistry, and thermal non-equilibrium solvers are loosely-coupled to reduce computational costs and simplify implementation. The linear system is solved using the GMRES method with an ILU preconditioner. Various zero, two, and three-dimensional numerical results on both structured and unstructured grids are presented. The numerical results are compared with references and good agreement is found in all test cases.

An anisotropic mesh optimization technique is studied on unstructured grids and it is found that mesh optimization is crucial to retrieve accurate solutions at no increase in mesh size. Some key features are (1) unstructured mesh generation on complex geometries is effortless, compared to structured meshes. (2) In non-trivial problems, the location of the salient flow features is not necessarily known a priori. Mesh adaptation effortlessly identifies these regions while locally optimizing the grid to resolve them. The sensitivity of the solution to the user input is thus minimized, as long as the initial mesh is sufficient. (3) The tetrahedral elements near the shock are automatically elongated and aligned to the shock. This results in a highly-stretched mesh featuring sharp shocks. (4) Wall quantities such as the heat-flux are indirectly, yet dramatically, improved by the resulting increased shock resolution.

To explore the potential of using Jacobian-free approaches for hypersonic flows, an accurate edge-based JFNK solver with an LU-SGS preconditioner is developed for thermal equilibrium flows with frozen chemistry. The traditional LU-SGS formulation is enriched by including the contributions from viscous fluxes and boundary conditions. Two and three-dimensional test cases are presented, demonstrating good agreement with the reference solutions. The performance of the JFNK solver is subsequently assessed. The improved LU-SGS is found to be more robust and efficient than the Jacobi preconditioner and the original LU-SGS. Comparisons between JFNK and the explicit Jacobian method are then carried out and the results show that the present method, although taking more computational time than the explicit Jacobian method, reduces the memory footprint by as much as 50%.

Future development can be carried out in the following aspects: firstly, a more efficient matrix-free preconditioner can be developed to accelerate the JFNK solver. The current LU-SGS preconditioner performs forward and backward sweeps on nodes. However, in the literature [24], it has been shown that line relaxation LU-SGS can improve the convergence of the JFNK method by performing these sweeps along a line in the direction normal to the wall. If the JFNK method combined with this new preconditioner can be as fast as the explicit method with the ILU preconditioner, the next step would be to extend the JFNK solver from equilibrium flows with frozen chemistry to thermo-chemical non-equilibrium flows. Secondly, the current hypersonic solver uses a loosely-coupled approach. This is an adequate approach at the beginning of the code development but has some disadvantages, such as stalls in the residuals and limitations on the CFL number, which suggest that fully coupling the solvers may improve convergence. Finally, to simulate more complex and realistic physics, it is necessary to develop additional physical models, such as weak ionization, ablation, and electron transpiration cooling. During re-entry, the temperature on the surface of the vehicle may be over 2,000 K, necessitating a thermal protection system. Ablation is able to reduce the heat on the surface by removing the surface material through chemical reactions. Electron transpiration cooling is another approach which decreases the heat by emitting electrons from the hot leading edge. Accurate modeling of the coupling between those physical phenomena and the flow is required to reduce design margins, thus improving efficiency. In this respect, the hypersonic flow solver developed in this work can be used as a solid basis for such future developments.

References

1. Sarma, G.S.R., *Physico-chemical Modelling in Hypersonic Flow Simulation.* Progress in Aerospace Sciences, 2000. **36**(3-4): p. 281-349.
2. Lynch, K., *Lockheed Martin, Aerion Reach MOU on Future of AS2.* Aviation International News, 2018; Available from: https://www.ainonline.com/aviation-news/business-aviation/2017-12-15/lockheed-martin-aerion-reach-mou-future-as2.
3. Nacheva, N. and Heldens, G., *The Next Generation of Commercial Supersonic Flight: Understanding the Industry and the Consumer Perspectives*. Master Thesis, Jönköping University, 2018.
4. Wuilbercq, R., Ahmad, A., Scanlon, T., and Brown, R., *Towards Robust Aero-Thermodynamic Predictions for Re-Usable Single-Stage to Orbit Vehicles*, *AIAA Paper 2012-5803*.
5. Mack, A. and Steelant, J., *FAST20XX: First Progress on European Future High-Altitude High-Speed Transport*, *AIAA Paper 2011-2337*.
6. Coltman, D., Turner, J., Curtis, R., Blake, D., Holland, B., Pearson, R., Arden, A., and Nuglisch, H., *Project Sabre: A Close-Spaced Direct Injection 3-Cylinder Engine with Synergistic Technologies to Achieve Low CO_2 Output.* SAE International Journal of Engines, 2009. **1**(1): p. 129-146.
7. Launius, R.D., *NASA: a History of the US Civil Space Program*. Krieger Pub. Co. 1994.
8. Kennicutt Jr, R.C., *Star Formation in Galaxies along the Hubble Sequence.* Annual Review of Astronomy and Astrophysics, 1998. **36**(1): p. 189-231.
9. Gibbs, G. and Sachdev, S., *Canada and the International Space Station Program: Overview and Status.* Acta Astronautica, 2002. **51**(1-9): p. 591-600.
10. Hiltz, M., Rice, C., Boyle, K., and Allison, R., *Canadarm: 20 Years of Mission Success through Adaptation.* International Symposium on Artificial Intelligence, Robotics and Automation in Space, Montreal, Quebec, 2001.
11. Misra, P. and Enge, P., *Global Positioning System: Signals, Measurements and Performance Second Edition.* Massachusetts: Ganga-Jamuna Press, 2006.
12. Seedhouse, E., *SpaceX: Making Commercial Spaceflight a Reality*. 2013: Springer-Parxis Books.
13. Liou, J.-C. and Johnson, N.L., *Risks in Space from Orbiting Debris*. Science, 2006. **311**(5759): p. 340-341.
14. Cheng, H.K. and Emanuel, G., *Perspective on Hypersonic Nonequilibrium Flow.* AIAA Journal, 1995. **33**(3): p. 385-400.
15. Law, C.K., *Combustion Physics*. 2006: Cambridge University Press.
16. Anderson, J.D., *Hypersonic and High-Temperature Gas Dynamics*. 2000: AIAA Textbooks.
17. Sakai, T. and Sawada, K., *Calculation of Nonequilibrium Radiation from a Blunt-Body Shock Layer.* Journal of Thermophysics and Heat Transfer, 2001. **15**(1): p. 99-105.
18. Bityurin, V.A., Vatazhin, A.B., Gus'kov, O.V., and Kopchenov, V.I., *Hypersonic Flow past the Spherical Nose of a Body in the Presence of a Magnetic Field.* Fluid Dynamics, 2004. **39**(4): p. 657-666.
19. Gosse, R. and Candler, G.V., *Ablation Modeling of Electro-Magnetic Launched Projectile for Access to Space*, *AIAA Paper 2007-1210.*

20. Bugel, M., Reynier, P., and Smith, A., *Survey of European and Major ISC Facilities for Supporting Mars and Sample Return Mission Aerothermodynamics and Tests Required for Thermal Protection System and Dynamic Stability.* International Journal of Aerospace Engineering, 2011. **2011**: p. 18.
21. Candler, G.V., *Rate-Dependent Energetic Processes in Hypersonic Flows.* Progress in Aerospace Sciences, 2013(72): p. 37-48.
22. Farbar, E., Boyd, I.D., and Martin, A., *Numerical Prediction of Hypersonic Flowfields including Effects of Electron Translational Nonequilibrium.* Journal of Thermophysics and Heat Transfer, 2013. **27**(4): p. 593-606.
23. Benjamin, K., Roy, S., Todd, A.O., and Paul, T.B., *Recent Advancements in Fully Implicit Numerical Methods for Hypersonic Reacting Flows*, *AIAA Paper 2013-2559.*
24. Scalabrin, L. and Boyd, I.D., *Numerical Simulation of Weakly Ionized Hypersonic Flow for Reentry Configurations*, *AIAA Paper 2006-3773.*
25. Gnoffo, P.A., *Computational Fluid Dynamics Technology for Hypersonic Applications*, *AIAA Paper 2003-2829.*
26. Gyftopoulos, E. and Beretta, G., *Thermodynamics: Foundations and Applications*. 2012: Courier Dover Publications.
27. Candler, G.V. and MacCormack, R.W., *Computation of Weakly Ionized Hypersonic Flows in Thermochemical Nonequilibrium.* Journal of Thermophysics and Heat Transfer, 1991. **5**(3): p. 266-273.
28. Lee, J.H., *Basic Governing Equations for the Flight Regimes of Aeroassisted Orbital Transfer Vehicles, AIAA Paper 1984-1729.*
29. Park, C., *Radiation Enhancement by Nonequilibrium in Earth's Atmosphere.* Journal of Spacecraft and Rockets, 1985. **22**(1): p. 27-36.
30. Scalabrin, L., *Numerical Simulation of Weakly Ionized Hypersonic Flow over Reentry Capsules*. Ph.D. Dissertation, University of Michigan, 2007.
31. Min Kwan, K., Ali, G., and Iain D., B., *Modeling of Electron Temperature in Hypersonic Flows*, *AIAA Paper 2011-1028.*
32. Park, C., *Assessment of Two-Temperature Kinetic Model for Ionizing Air.* Journal of Thermophysics and Heat Transfer, 1989. **3**(3): p. 233-244.
33. Park, C., *Problems of Rate Chemistry in the Flight Regimes of Aeroassisted Orbital Transfer Vehicles*, *AIAA Paper 1984-1730.*
34. 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 TM-2867, 1989.
35. Candler, G.V., Subbareddy, P.K., and Nompelis, I., *Decoupled Implicit Method for Aerothermodynamics and Reacting Flows.* AIAA Journal, 2013. **51**(5): p. 1245-1254.
36. Schwartz, R.N., Slawsky, Z.I., and Herzfeld, K.F., *Calculation of Vibrational Relaxation Times in Gases.* The Journal of Chemical Physics, 1952. **20**(10): p. 1591-1599.
37. Park, C., *Review of Chemical-Kinetic Problems of Future NASA Missions .I. Earth Entries.* Journal of Thermophysics and Heat Transfer, 1993. **7**(3): p. 385-398.
38. Park, C., Howe, J.T., Jaffe, R.L., and Candler, G.V., *Review of Chemical-Kinetic Problems of Future NASA Missions. II - Mars Entries.* Journal of Thermophysics and Heat Transfer, 1994. **8**(1): p. 9-23.

39. Blottner, F.G., Johnson, M., and Ellis, M., *Chemically Reaction Viscous Flow Program for Multi-Component Gas Mixtures*. 1971, Sandia Laboratories Report NO. SC-RR-70-754, Albuquerque, New Mexico.
40. Vincenti, W.G. and Kruger, C.H., *Introduction to Physical Gas Dynamics*. 1967: Krieger.
41. Wilke, C.R., *A Viscosity Equation for Gas Mixtures.* The Journal of Chemical Physics, 1950. **18**(4): p. 517-519.
42. Yos, J.M., *Transport Properties of Nitrogen, Hydrogen, Oxygen, and Air to 30,000 K.* 1963, Tech. Memo., RAD-TM-63-7: Wilmington, Delaware.
43. 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 30000 K.* NASA STI/Recon Technical Report N, 1990. **90**: p. 27064.
44. Wright, M.J., Bose, D., Palmer, G.E., and Levin, E., *Recommended Collision Integrals for Transport Property Computations Part 1: Air Species.* AIAA Journal, 2005. **43**(12): p. 2558-2564.
45. Callen, H.B., *Thermodynamics and an Introduction to Thermostatistics*. 2006, New York: Wiley.
46. Chase, M.W., *NIST-JANAF Thermochemical Tables for the Bromine Oxides.* Journal of Physical and Chemical Reference Data, 1996. **25**(4): p. 1069-1111.
47. Hosoya, A., Sakagami, M.-A., and Takao, M., *Nonequilibrium Thermodynamics in Field Theory: Transport Coefficients.* Annals of Physics, 1984. **154**(1): p. 229-252.
48. Masel, R.I., *Chemical Kinetics and Catalysis*. 2001: Wiley-Interscience New York.
49. Levine, R.D., *Molecular Reaction Dynamics*. 2005: Cambridge University Press.
50. Hash, D., Olejniczak, J., Wright, M., Prabhu, D., Pulsonetti, M., Hollis, B., Gnoffo, P.A., Barnhardt, M., Nompelis, I., and Candler, G.A., *FIRE II Calculations for Hypersonic Nonequilibrium Aerothermodynamics Code Verification: DPLR, LAURA, and US3D, AIAA Paper 2007-605.*
51. Casseau, V., Espinoza, D.E., Scanlon, T.J., and Brown, R.E., *A Two-Temperature Open-Source CFD Model for Hypersonic Reacting Flows, Part Two: Multi-Dimensional Analysis.* Aerospace, 2016. **3**(4): p. 45.
52. Lani, A., *An Object Oriented and High-performance Platform for Aerothermodynamics Simulations*. Ph.D. Thesis, Von Karman Institute for Fluid Dynamics, 2009.
53. Ait-Ali-Yahia, D. and Habashi, W.G., *Finite Element Adaptive Method for Hypersonic Thermochemical Nonequilibrium Flows.* AIAA Journal, 1997. **35**(8): p. 1294-1302.
54. Kirk, B.S., Stogner, R.H., Bauman, P.T., and Oliver, T.A., *Modeling Hypersonic Entry with the Fully-Implicit Navier–Stokes (FIN-S) Stabilized Finite Element Flow Solver.* Computers & Fluids, 2014. **92**(0): p. 281-292.
55. Brooks, A.N. and Hughes, T.J.R., *Streamline Upwind Petrov-Galerkin Formulations for Convection Dominated Flows with Particular Emphasis on the Incompressible Navier-Stokes Equations.* Computer Methods in Applied Mechanics and Engineering, 1982. **32**(1-3): p. 199-259.
56. Balay, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Rupp, K., Smith, B.F., and Zhang, H. *Petsc Web Page*. 2014; Available from: http://www.mcs.anl.gov/petsc.
57. Isola, D., *Equivalence Conditions between Continuous Linear Finite Element and Finite Volume Discretizations of Non-Simplicial Elements*, *AIAA Paper 2017-4276.*

58. Knoll, D.A. and Keyes, D.E., *Jacobian-Free Newton-Krylov Methods: a Survey of Approaches and Applications.* Journal of Computational Physics, 2004. **193**(2): p. 357-397.
59. Liu, Z., Ahmed, M., Sabir, A., Humphries, S., and Goldberg, S.N., *Computer modeling of the effect of perfusion on heating patterns in radiofrequency tumor ablation.* International Journal of Hyperthermia, 2007. **23**(1): p. 49-58.
60. Luo, H., Baum, J.D., and Löhner, R., *A Fast, Matrix-free Implicit Method for Compressible Flows on Unstructured Grids.* Journal of Computational Physics, 1998. **146**(2): p. 664-690.
61. Jameson, A. and Yoon, S., *Lower-upper Implicit Schemes with Multiple Grids for the Euler Equations.* AIAA Journal, 1987. **25**(7): p. 929-935.
62. Sharov, D. and Nakahashi, K., *Reordering of Hybrid Unstructured Grids for Lower-upper Symmetric Gauss-Seidel Computations.* AIAA Journal, 1998. **36**(3): p. 484-486.
63. Schwamborn, D., Gerhold, T., and Heinrich, R., *The DLR TAU-code: Recent Applications in Research and Industry.* 2006.
64. Barbante, P.F. and Magin, T., *Fundamentals of Hypersonic Flight–Properties of High Temperature Gases.* Critical Technologies for Hypersonic Vehicle Development, 2004.
65. McBride, B.J., Zehe, M.J., and Gordon, S., *NASA Glenn Coefficients for Calculating Thermodynamic Properties of Individual Species.* 2002.
66. Chapman, S., Cowling, T.G., and Burnett, D., *The Mathematical Theory of Non-uniform Gases: an Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases*. 1970: Cambridge University Press.
67. McBride, B.J., Gordon, S., and Reno, M.A., *Coefficients for Calculating Thermodynamic and Transport Properties of Individual Species*. 1993, NASA TM–4513.
68. Wilke, C., *A Viscosity Equation for Gas Mixtures.* Journal of Chemical Physics, 1950. **18**(4): p. 517-519.
69. Mason, E. and Saxena, S., *Approximate Formula for the Thermal Conductivity of Gas Mixtures.* The Physics of Fluids, 1958. **1**(5): p. 361-369.
70. Gosse, R. and Candler, G.V., *Diffusion Flux Modeling: Application to Direct Entry Problems*, *AIAA Paper 2005-0389*. 2005. p. 389.
71. Millikan, R.C. and White, D.R., *Systematics of Vibrational Relaxation.* The Journal of chemical physics, 1963. **39**(12): p. 3209-3213.
72. Park, C., *Nonequilibrium Hypersonic Aerothermodynamics.* 1989.
73. David, H., *Fundamentals of Finite Element Analysis.* McGraw−Hill, USA, 2004.
74. Fletcher, C., *A Comparison of Finite Element and Finite Difference Solutions of the One- and Two-dimensional Burgers' Equations.* Journal of Computational Physics, 1983. **51**(1): p. 159-188.
75. Löhner, R. and Galle, M., *Minimization of Indirect Addressing for Edge‐based Field Solvers.* Communications in Numerical Methods in Engineering, 2002. **18**(5): p. 335-343.
76. Selmin, V., *The Node-centred Finite Volume Approach: Bridge between Finite Differences and Finite Elements.* Computer Methods in Applied Mechanics and Engineering, 1993. **102**(1): p. 107-138.
77. Selmin, V. and Formaggia, L., *Simulation of Hypersonic Flows on Unstructured Grids.* International journal for numerical methods in engineering, 1992. **34**(2): p. 569-606.
78. Roe, P.L., *Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes.* Journal of computational physics, 1981. **43**(2): p. 357-372.

79. Fluent, A., *12.0 Theory Guide.* Ansys Inc, 2009. **5**(5).
80. LeVeque, R.J., *Conservative Methods for Nonlinear Problems*, in *Numerical Methods for Conservation Laws*. 1990, Springer. p. 122-135.
81. Larrouturou, B., *How to Preserve the Mass Fractions Positivity When Computing Compressible Multi-component Flows.* Journal of Computational Physics, 1991. **95**(1): p. 59-84.
82. Van Leer, B., *Towards the Ultimate Conservative Difference Scheme. V. A Second-order Sequel to Godunov's Method.* Journal of Computational Physics, 1979. **32**(1): p. 101-136.
83. Dervieux, A. and Desideri, J.A., *Compressible Flow Solvers Using Unstructured Grids*. Institut National de Recherche en Informatique et en Automatique, Rapport INRIA 1732, 1992.
84. Venkatakrishnan, V., *Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters.* Journal of Computational Physics, 1995. **118**(1): p. 120-130.
85. Jameson, A., *Time Dependent Calculations using Multigrid, with Applications to Unsteady Flows past Airfoils and Wings*, *AIAA Paper 1991-1596*.
86. Gropp, W.D., Gropp, W., Lusk, E., and Skjellum, A., *Using MPI: Portable Parallel Programming with the Message-passing Interface*. Vol. 1. 1999: MIT Press.
87. Schloegel, K. and Kumar, V., *Parmetis: Parallel Graph Partitioning and Sparse Matrix Ordering Library*. 1997, Tech. Report 97-060, Department of Computer Science, University of Minnesota, Minneapolis.
88. Saad, Y. and Schultz, M.H., *GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems.* SIAM Journal on Scientific and Statistical Computing, 1986. **7**(3): p. 856-869.
89. Barrett, R., Berry, M.W., Chan, T.F., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., and Van der Vorst, H., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Vol. 43. 1994: SIAM Press.
90. Saad, Y., *Iterative Methods for Sparse Linear Systems*. Vol. 82. 2003: SIAM Press.
91. Dumas, M.-E., Habashi, W.G., Baruzzi, G.S., Isola, D., and Fossati, M., *Finite Element Modeling of Nonequilibrium Fluid–Wall Interaction at High-Mach Regime.* AIAA Journal of Aircraft, 2017. **54**(6): p. 2330-2339.
92. Seguin, J., *Non-equilibrium Chemistry for Hypersonic Flows using an Edge-based Finite Element Method*. Master Thesis, McGill University, 2018.
93. Casseau, V., Scanlon, T.J., and Brown, R.E., *Development of a Two-temperature Open-source CFD Model for Hypersonic Reacting Flows*, *AIAA Paper 2015-3637*.
94. Candler, G.V., Mavriplis, D., and Trevino, L., *Current Status and Future Prospects for the Numerical Simulation of Hypersonic Flows*, *AIAA Paper 2009-153*.
95. Habashi, W.G., Dompierre, J., Bourgault, Y., Ait-Ali-Yahia, D., Fortin, M. and Vallet, M-G., *Anisotropic Mesh Adaptation: Towards User-Independent, Mesh-Independent and Solver-Independent CFD Solutions: Part I: General Principles*. International Journal for Numerical Methods in Fluids, 2000. **32**(6): p. 725-744.
96. Ait-Ali-Yahia, D., Baruzzi, G.S., Habashi, W.G., Fortin, M., Dompierre, J. and Vallet, M-G., *Anisotropic Mesh Adaptation: Towards User-Independent, Mesh-Independent and Solver-Independent CFD Solutions: Part II: Structured Grids*. International Journal for Numerical Methods in Fluids, 2002. **39**(8): p. 657-673.

97. Dompierre, J., Vallet, M-G., Bourgault, Y., Fortin, M. and Habashi, W.G., *Anisotropic Mesh Adaptation: Towards User-Independent, Mesh-Independent and Solver-Independent CFD Solutions: Part III: Unstructured Grids*. International Journal for Numerical Methods in Fluids, 2002. **39**(8): p. 675-702.
98. Habashi, W.G., Dompierre, J., Bourgault, Y., Fortin, M. and Vallet, M-G., *Certifiable Computational Fluid Dynamics Through Mesh Optimization, Special Issue on Credible Computational Fluid Dynamics Simulation.* AIAA Journal, 1998. **36**(5): p. 703-711.
99. Balasubramanian, R. and Anandhanarayanan, K., *Development of a Thermo-chemical Non-equilibrium Solver for Hypervelocity Flows.* Journal of the Institution of Engineers (India): Series C, 2015. **96**(2): p. 119-134.
100. Landau, L. and Lifshitz, E., *Fluid mechanics, 1959.* Course of Theoretical Physics, 1959.
101. Saad, Y., *A Flexible Inner-outer Preconditioned GMRES Algorithm.* SIAM Journal on Scientific Computing, 1993. **14**(2): p. 461-469.
102. Gao, S., Habashi, W.G., Isola, D., Baruzzi, G.S., and Fossati, M., *A Jacobian-free Edge-based Galerkin Formulation for Compressible Flows.* Computers & Fluids, 2017. **143**: p. 141-156.
103. Liou, M.-S., *A Sequel to AUSM: AUSM+.* Journal of Computational Physics, 1996. **129**(2): p. 364-382.
104. Liou, M.-S., *A Sequel to AUSM, Part II: AUSM+-up for All Speeds.* Journal of computational physics, 2006. **214**(1): p. 137-170.
105. Gnoffo, P.A., *Forebody and Afterbody Solutions of the Navier-Stokes Equations for Supersonic Flow over Blunt Bodies in a Generalized Orthogonal Coordinate System.* NASA TP-107, 1978.
106. Spalart, P. and Allmaras, S., *A One-equation Turbulence Model for Aerodynamic Flows, AIAA Paper 1992-439.*
107. Rohde, A., *Eigenvalues and Eigenvectors of the Euler Equations in General Geometries, AIAA Paper 2001-2609.*

Appendix A: Vector and Tensor Identities

The inner product between two vectors is defined as

$$\mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{3} x_i y_i \tag{0.1}$$

where **x** and **y** are two vectors, x_i and y_i are components of **x** and **y**, respectively.

The inner product between a vector and a second-order tensor is defined as

$$(\mathbf{x} \cdot \mathbf{T})_j = \sum_{i=1}^{3} x_i T_{ij} \tag{0.2}$$

where **x** is a vector, **T** is a second-order tensor, x_i and T_{ij} are components of **x** and **T**, respectively.

The column product between two second-order tensors **T** and **S** is defined as

$$\mathbf{T} : \mathbf{S} = \sum_{i=1}^{3} \sum_{j=1}^{3} T_{ij} S_{ij} \tag{0.3}$$

where T_{ij} and S_{ij} are components of **T** and **S**, respectively.

The L_2 norm of a vector R is defined as

$$|R|_{L_2} = \sqrt{\sum_i R_i^2} \tag{0.4}$$

Appendix B: Species Data

This appendix lists data used in the present work. Table 2 shows the species data, including molecular weight, heat formation at 0K, characteristic vibrational temperature and dissociation potential for the preferential model. Table 3 lists the coefficients for Blottner's curve fits. Table 4 lists the coefficients for NIST polynomial curve fits.

Table 2 Species data

Species	M_s[kg/m^3]	$h_{0,s}$[J/Kg]	$\theta_{v,s}[K]$	D_s[J/Kg]
N_2	28	0	3395	3.36E+07
O_2	32	0	2239	1.54E+07
N	14	3.36E+07	0	0
O	16	1.54E+07	0	0
NO	30	3.00E+06	2817	2.09E+07

Table 3 Blottner's curve fits

Species	Blottner's curve fit		
	A_s	B_s	C_s
N_2	2.68E-02	3.18E-01	-1.13E+01
O_2	4.49E-02	-8.26E-02	-9.20E+00
N	1.16E-02	6.03E-01	-1.24E+01
O	2.03E-02	4.29E-01	-1.16E+01
NO	4.36E-02	-3.36E-02	-9.58E+00

Table 4 NASA NIST polynomial curve fits

Species	NIST polynomials				
	Temperature range	A_s	B_s	C_s	D_s
N_2	200-1000 K	0.62526577E0	-0.31779652E2	-0.16407983E4	0.17464992E1
	1000-5000 K	0.87395208E0	0.56152222E3	-0.17394809E6	-0.39335958E0
	5000-15000 K	0.88503551E0	0.90902171E3	-0.73129061E6	-0.53503838E00

O_2	200-1000 K	0.60916180E0	-0.52244847E2	-0.59974009E3	0.20410801E1
	1000-5000 K	0.72216486E0	0.17550839E3	-0.57974816E5	0.10901044E1
	5000-15000 K	0.73981127E0	0.39194906E3	-0.37833168E6	0.90931780E0
N	1000-5000 K	0.83724737E0	0.43997150E3	-0.17450753E6	0.10365689E0
	5000-15000 K	0.89986588E0	0.14112801E4	-0.18200478E7	-0.55811716E0
O	1000-5000 K	0.77269241E0	0.83842977E2	-0.58502098E5	0.85100827E0
	5000-15000 K	0.87669586E0	0.10158420E4	-0.10884566E7	-0.18001077E0
NO	200-1000 K	0.60262029E0	-0.62017783E2	-0.13954524E3	0.20268332E1
	1000-5000 K	0.78009050E0	0.30486891E3	-0.94847722E5	0.52873381E0
	5000-15000 K	0.8058058xE0	0.62427878E3	-0.57879210R6	0.26516450E0

Appendix C: Chemical Models

Table 5 lists the Park's 1993 reaction data used in the present work.

Table 5 Park's 1993 reaction model

Reaction	M	A_r^f	η_r^f	$\frac{E_a}{R_u}$	$K_r^{eq}(\bar{T})$				
					$B_{r,2}$	$B_{r,3}$	$B_{r,8}$	$B_{r,4}$	$B_{r,5}$
$N_2 + M \leftrightarrow 2N + M$	N_2	7.00E21	-1.60	113200	1.4766	1.6291	1.2153	-11.457	-0.009444
	O_2	7.00E21	-1.60	113200					
	NO	7.00E21	-1.60	113200					
	N	3.00E21	-1.60	113200					
	O	3.00E21	-1.60	113200					
$O_2 + M \leftrightarrow 2O + M$	N_2	2.00E21	-1.50	59500	0.50989	2.4773	1.7132	-6.5441	0.02959
	O_2	2.00E21	-1.50	59500					
	NO	2.00E21	-1.50	59500					
	N	1.00E22	-1.50	59500					
	O	1.00E22	-1.50	59500					
$NO + M \leftrightarrow N + O + M$	N_2	5.00E15	0.00	75500	0.50765	0.73575	0.48042	-7.4979	-0.016247
	O_2	5.00E15	0.00	75500					
	NO	5.00E15	0.00	75500					
	N	1.10E17	0.00	75500					
	O	1.10E17	0.00	75500					
$NO + O \leftrightarrow N + O_2$	-	8.40E12	0.00	19450	-0.00242	-1.7415	-1.2331	-0.95365	-0.04585
$O + N_2 \leftrightarrow N + NO$	-	6.40E17	-1.00	38400	0.96921	0.89329	0.73531	-3.9596	0.006818

Appendix D: Derivation of Jacobian Matrices

Jacobian of Inviscid Fluxes

Following [107], <u>the Jacobian of inviscid fluxes $\boldsymbol{\Phi}_{ij}^{\text{num,flow}}$ for the flow solver</u> in Eq. (3.8) can be expressed with the help of $\hat{A}$

$$\hat{A} = \frac{\partial \mathbf{F}^{A} \cdot \mathbf{n}}{\partial Q}$$

$$= \begin{bmatrix} 0 & n_x & n_y & n_z & 0 \\ \Pi_\rho n_x - uV_n & V_n - (\gamma - 2)un_x & un_y - (\gamma - 1)vn_x & un_z - (\gamma - 1)wn_x & (\gamma - 1)n_x \\ \Pi_\rho n_y - vV_n & vn_x - (\gamma - 1)un_y & V_n - (\gamma - 2)vn_y & vn_z - (\gamma - 1)wn_y & (\gamma - 1)n_y \\ \Pi_\rho n_z - wV_n & wn_x - (\gamma - 1)un_z & wn_y - (\gamma - 1)vn_z & V_n - (\gamma - 2)wn_z & (\gamma - 1)n_z \\ [\Pi_\rho - h]V_n & hn_x - (\gamma - 1)uV_n & hn_y - (\gamma - 1)vV_n & hn_z - (\gamma - 1)wV_n & \gamma V_n \end{bmatrix} \tag{0.5}$$

where

$$\Pi_\rho = (\gamma - 2)[\frac{\boldsymbol{V} \cdot \boldsymbol{V}}{2} - e] + RT == (\gamma - 1)(\boldsymbol{V} \cdot \boldsymbol{V} - h) + \gamma RT \tag{0.6}$$

To derive the Roe scheme, an eigenvalue decomposition of $\hat{A}$ is necessary. The eigenvalues are

$$\hat{\Lambda} = [V_n - a, V_n, V_n + a, V_n, V_n\,]\mathrm{I}_{5\times 5} \qquad (0.7)(0.8)$$

where

$$V_n = \boldsymbol{V} \cdot \mathbf{n}, \qquad \mathbf{n} = \frac{\boldsymbol{\eta}_{ij}}{|\boldsymbol{\eta}_{ij}|}, \qquad \mathbf{n} = \left[n_x, n_y, n_z\right]^{\boldsymbol{T}} \qquad (0.9)(0.10)$$

Since only three distinct eigenvalues exist and the eigenvalue V_n is degenerate, there is no uniquely defined eigenvector. Following [107], the optimal numerical expressions of eigenvectors are given by three conditions, depending on the values of components of vector **n**.

In the first case, $|n_x| > |n_y|$ and $|n_x| > |n_z|$. The left eigenvectors and right eigenvectors are

$$\hat{L} = \begin{bmatrix} \dfrac{\Pi_\rho + aV_n}{2a^2} & \dfrac{(1-\gamma)u - an_x}{2a^2} & \dfrac{(1-\gamma)v - an_y}{2a^2} & \dfrac{(1-\gamma)w - an_z}{2a^2} & \dfrac{\gamma - 1}{2a^2} \\ 1 - \dfrac{\Pi_\rho}{a^2} & \dfrac{\gamma - 1}{a^2}u & \dfrac{\gamma - 1}{a^2}v & \dfrac{\gamma - 1}{a^2}w & \dfrac{1-\gamma}{a^2} \\ \dfrac{\Pi_\rho - aV_n}{2a^2} & \dfrac{(1-\gamma)u + an_x}{2a^2} & \dfrac{(1-\gamma)v + an_y}{2a^2} & \dfrac{(1-\gamma)w + an_z}{2a^2} & \dfrac{\gamma - 1}{2a^2} \\ \dfrac{v - V_n n_y}{n_x} & n_y & \dfrac{n_y^2 - 1}{n_x} & \dfrac{n_z n_y}{n_x} & 0 \\ \dfrac{V_n n_z - w}{n_x} & -n_z & -\dfrac{n_z n_y}{n_x} & \dfrac{1 - n_z^2}{n_x} & 0 \end{bmatrix} \qquad (0.11)(0.12)$$

$$\hat{R} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ u - an_x & u & u + an_x & n_y & -n_z \\ v - an_y & v & v + an_y & -n_x & 0 \\ w - an_z & w & w + an_z & 0 & n_x \\ h - aV_n & h - \dfrac{a^2}{\gamma - 1} & h + aV_n & un_y - vn_x & wn_x - un_z \end{bmatrix}$$

In the second case, $|n_y| > |n_x|$ and $|n_y| > |n_z|$. The left eigenvectors and right eigenvectors are

$$\hat{L} = \begin{bmatrix} \dfrac{\Pi_\rho + aV_n}{2a^2} & \dfrac{(1-\gamma)u - an_x}{2a^2} & \dfrac{(1-\gamma)v - an_y}{2a^2} & \dfrac{(1-\gamma)w - an_z}{2a^2} & \dfrac{\gamma - 1}{2a^2} \\ 1 - \dfrac{\Pi_\rho}{a^2} & \dfrac{\gamma - 1}{a^2}u & \dfrac{\gamma - 1}{a^2}v & \dfrac{\gamma - 1}{a^2}w & \dfrac{1 - \gamma}{a^2} \\ \dfrac{\Pi_\rho - aV_n}{2a^2} & \dfrac{(1-\gamma)u + an_x}{2a^2} & \dfrac{(1-\gamma)v + an_y}{2a^2} & \dfrac{(1-\gamma)w + an_z}{2a^2} & \dfrac{\gamma - 1}{2a^2} \\ \dfrac{V_n n_x - u}{n_y} & \dfrac{1 - n_x^2}{n_y} & -n_x & -\dfrac{n_x n_z}{n_y} & 0 \\ \dfrac{w - V_n n_z}{n_y} & \dfrac{n_x n_z}{n_y} & n_z & \dfrac{n_z^2 - 1}{n_y} & 0 \end{bmatrix} \qquad (0.13)(0.14)$$

$$\hat{R} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ u - an_x & u & u + an_x & n_y & 0 \\ v - an_y & v & v + an_y & -n_x & n_z \\ w - an_z & w & w + an_z & 0 & -n_y \\ h - aV_n & h - \dfrac{a^2}{\gamma - 1} & h + aV_n & un_y - vn_x & vn_z - wn_y \end{bmatrix}$$

In the third case, $|n_z| > |n_x|$ and $|n_z| > |n_x|$. The left eigenvectors and right eigenvectors are

$$\hat{L} = \begin{bmatrix} \dfrac{\Pi_\rho + aV_n}{2a^2} & \dfrac{(1-\gamma)u - an_x}{2a^2} & \dfrac{(1-\gamma)v - an_y}{2a^2} & \dfrac{(1-\gamma)w - an_z}{2a^2} & \dfrac{\gamma - 1}{2a^2} \\ 1 - \dfrac{\Pi_\rho}{a^2} & \dfrac{\gamma - 1}{a^2}u & \dfrac{\gamma - 1}{a^2}v & \dfrac{\gamma - 1}{a^2}w & \dfrac{1 - \gamma}{a^2} \\ \dfrac{\Pi_\rho - aV_n}{2a^2} & \dfrac{(1-\gamma)u + an_x}{2a^2} & \dfrac{(1-\gamma)v + an_y}{2a^2} & \dfrac{(1-\gamma)w + an_z}{2a^2} & \dfrac{\gamma - 1}{2a^2} \\ \dfrac{u - V_n n_x}{n_z} & \dfrac{n_x^2 - 1}{n_z} & \dfrac{n_x n_y}{n_z} & n_x & 0 \\ \dfrac{V_n n_y - v}{n_z} & -\dfrac{n_x n_y}{n_z} & \dfrac{1 - n_y^2}{n_z} & -n_y & 0 \end{bmatrix} \qquad (0.15)(0.16)$$

$$\hat{R} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ u - an_x & u & u + an_x & -n_z & 0 \\ v - an_y & v & v + an_y & 0 & n_z \\ w - an_z & w & w + an_z & n_x & -n_y \\ h - aV_n & h - \dfrac{a^2}{\gamma - 1} & h + aV_n & wn_x - un_z & vn_z - wn_y \end{bmatrix}$$

The Jacobians of inviscid fluxes $\boldsymbol{\Phi}_{s,ij}^{\text{num,chemistry}}$ for the chemistry solver is derived following [92],

$$\begin{aligned} \frac{\partial \boldsymbol{\Phi}_{s,ij}^{\text{num,chemistry}}}{\partial \rho_i Y_{s,i}} &= \frac{\boldsymbol{\Phi}_{ij}^{\text{num,flow,density}}}{\rho_i} \times \begin{cases} 1 & \text{if } \boldsymbol{\Phi}_{ij}^{\text{num,flow,density}} > 0 \\ 0 & \text{if } \boldsymbol{\Phi}_{ij}^{\text{num,flow,density}} \leq 0 \end{cases} \\ \frac{\partial \boldsymbol{\Phi}_{s,ij}^{\text{num,chemistry}}}{\partial \rho_j Y_{s,j}} &= \frac{\boldsymbol{\Phi}_{ij}^{\text{num,flow,density}}}{\rho_j} \times \begin{cases} 0 & \text{if } \boldsymbol{\Phi}_{ij}^{\text{num,flow,density}} > 0 \\ 1 & \text{if } \boldsymbol{\Phi}_{ij}^{\text{num,flow,density}} \leq 0 \end{cases} \end{aligned} \tag{0.18}$$

The Jacobians of inviscid fluxes $\boldsymbol{\Phi}_{ij}^{\text{num,thermal}}$ for the thermal non-equilibrium solver is

$$\begin{aligned} \frac{\partial \boldsymbol{\Phi}_{ij}^{\text{num,thermal}}}{\partial \rho_i e_{ve,i}} &= \begin{cases} \boldsymbol{V}_i & \text{if } \widetilde{\boldsymbol{V}} \cdot \boldsymbol{\eta}_{ij} > 0 \\ 0 & \text{if } \widetilde{\boldsymbol{V}} \cdot \boldsymbol{\eta}_{ij} \leq 0 \end{cases} \\ \frac{\partial \boldsymbol{\Phi}_{ij}^{\text{num,thermal}}}{\partial \rho_j e_{ve,j}} &= \begin{cases} \boldsymbol{0} & \text{if } \widetilde{\boldsymbol{V}} \cdot \boldsymbol{\eta}_{ij} > 0 \\ \boldsymbol{V}_j & \text{if } \widetilde{\boldsymbol{V}} \cdot \boldsymbol{\eta}_{ij} \leq 0 \end{cases} \end{aligned} \tag{0.20}$$

Jacobian of Viscous Fluxes

The Jacobians of viscous fluxes are given individually. For the momentum equation, defining $\mathbf{F}_{ij}^{F,M}$ as

$$\mathbf{F}_{ij}^{F,M} = \mathbf{I}^{(m)}\left(\mu_{ij}\operatorname{tr}(\mathbf{d}_{ij})\mathbf{I} + (\mu_{ij} + \lambda_{ij})\mathbf{d}_{ij}^{s} + (\lambda_{ij} - \mu_{ij})\mathbf{d}_{ij}^{A}\right)(\boldsymbol{V}_j - \boldsymbol{V}_i) \tag{0.21}$$

the stress tensor in Eq. (3.9) can be reformulated as

$$\sum_{e \in E_i} \int_V \nabla W_i \cdot \boldsymbol{\tau} dV == \sum_{j \in K_i} \mathbf{F}_{ij}^{F,M} \tag{0.22}$$

The derivatives of $\mathbf{F}_{ij}^{F,M}$ can be written as

$$\begin{aligned} \frac{\partial \mathbf{F}_{ij}^{F,M}}{\partial \boldsymbol{V}_i} &= -A_{ij}^{V,S} - A_{ij}^{V,A}, \quad \frac{\partial \mathbf{F}_{ij}^{F,M}}{\partial \boldsymbol{V}_j} = A_{ij}^{V,S} + A_{ij}^{V,A} \\ \frac{\partial \mathbf{F}_{ji}^{F,M}}{\partial \boldsymbol{V}_i} &= A_{ij}^{V,S} - A_{ij}^{V,A}, \quad \frac{\partial \mathbf{F}_{ji}^{F,M}}{\partial \boldsymbol{V}_i} = -A_{ij}^{V,S} + A_{ij}^{V,A} \end{aligned} \tag{0.23}$$

For the total energy equation, defining $\mathbf{F}_{ij}^{F,E}$ as

$$\mathbf{F}_{ij}^{F,E} = \mathbf{I}^{(e)}\big(\mu_{ij}\mathrm{tr}(\mathbf{d}_{ij})\mathbf{I} + (\mu_{ij}+\lambda_{ij})d_{ij}^{S} + (\lambda_{ij}-\mu_{ij})\mathbf{d}_{ij}^{A}\big):\boldsymbol{V}_{ij}(\boldsymbol{V}_j - \boldsymbol{V}_i) \tag{0.24}$$

the stress tensor in Eq. (3.12) can be reformulated as

$$\sum_{e\in E_i}\int_V \nabla W_i \cdot (\boldsymbol{\tau}\cdot\boldsymbol{V})dV = \sum_{j\in K_i}\mathbf{F}_{ij}^{F,E} \tag{0.25}$$

The derivatives of $\mathbf{F}_{ij}^{F,E}$ can be written as

$$\begin{aligned} \frac{\partial\mathbf{F}_{ij}^{F,E}}{\partial\boldsymbol{V}_i} &= -\boldsymbol{V}_{ij}\cdot A_{ij}^{V,S} - \boldsymbol{V}_{ij}\cdot A_{ij}^{V,A}, \quad \frac{\partial\mathbf{F}_{ij}^{F,E}}{\partial\boldsymbol{V}_j} = \boldsymbol{V}_{ij}\cdot A_{ij}^{V,S} + \boldsymbol{V}_{ij}\cdot A_{ij}^{V,A} \\ \frac{\partial\mathbf{F}_{ji}^{F,E}}{\partial\boldsymbol{V}_i} &= \boldsymbol{V}_{ij}\cdot A_{ij}^{V,S} - \boldsymbol{V}_{ij}\cdot A_{ij}^{V,A}, \quad \frac{\partial\mathbf{F}_{ji}^{F,E}}{\partial\boldsymbol{V}_i} = -\boldsymbol{V}_{ij}\cdot A_{ij}^{V,S} + \boldsymbol{V}_{ij}\cdot A_{ij}^{V,A} \end{aligned} \tag{0.26}$$

To complete the Jacobian with respect to conservative variables, the derivatives of velocity are

$$\frac{\partial\boldsymbol{V}}{\partial\rho} = -\frac{\boldsymbol{V}}{\rho}, \quad \frac{\partial\boldsymbol{V}}{\partial\rho\boldsymbol{V}} = \frac{\mathbf{I}_{3\times3}}{\rho}, \quad \frac{\partial\boldsymbol{V}}{\partial\rho e} = 0 \tag{0.27}$$

For the total energy equation, the heat conduction of translation-rotational energy can be reformulated by defining $\mathbf{F}_{ij}^{F,H}$

$$\mathbf{F}_{ij}^{F,H} = \mathbf{I}^{(e)}\left(k_{tr,ij}\mathrm{tr}(\mathbf{d}_{ij})\right)(T_{tr,j} - T_{tr,i}) \tag{0.28}$$

the heat fluxes in Eq. (3.13) can be reformulated as

$$\sum_{e\in E_i}\int_V \nabla W_i \cdot \boldsymbol{q}_{tr}dV = \sum_{j\in K_i}\mathbf{F}_{ij}^{F,H} \tag{0.29}$$

The derivatives of $\mathbf{F}_{ij}^{F,H}$ can be written as

$$\begin{aligned} \frac{\partial\mathbf{F}_{ij}^{F,H}}{\partial T_{tr,i}} &= -k_{tr,ij}\mathrm{tr}(\mathbf{d}_{ij}), \quad \frac{\partial\mathbf{F}_{ij}^{F,H}}{\partial T_{tr,j}} = k_{tr,ij}\mathrm{tr}(\mathbf{d}_{ij}) \\ \frac{\partial\mathbf{F}_{ji}^{F,H}}{\partial T_{tr,i}} &= k_{tr|ve,ij}\mathrm{tr}(\mathbf{d}_{ij}), \quad \frac{\partial\mathbf{F}_{ji}^{F,H}}{\partial T_{tr,j}} = -k_{tr,ij}\mathrm{tr}(\mathbf{d}_{ij}) \end{aligned} \tag{0.30}$$

The derivatives of heat conduction of vibrational-electronic energy can be derived in the same way. The derivatives of temperatures with respect to conservative variables are

$$\begin{aligned} \frac{\partial T_{tr}}{\partial\rho} &= \frac{1}{\rho C_v}\left(\frac{\boldsymbol{V}\cdot\boldsymbol{V}}{2} - e\right), \quad \frac{\partial T_{tr}}{\partial\rho\boldsymbol{V}} = -\frac{\boldsymbol{V}}{\rho C_v}, \quad \frac{\partial T_{tr}}{\partial\rho e} = \frac{1}{\rho C_v} \\ \frac{\partial T_{ve}}{\partial\rho} &= 0, \quad \frac{\partial T_{ve}}{\partial\rho\boldsymbol{V}} = 0, \quad \frac{\partial T_{ve}}{\partial\rho e} = 0 \end{aligned} \tag{0.31}$$

The heat conduction term in vibrational-electronic energy equation can be treated similarly, with the derivative given as

$$\frac{\partial T_{ve}}{\partial \rho e_{ve}} = \frac{1}{\rho C_v^{ve}} \tag{0.32}$$

The inter-diffusional mass flux term (3.14) in species mass equation can be written as

$$\sum_{e \in E_i} \int_V \nabla W_i \cdot \boldsymbol{J}_s dV = \sum_{j \in K_i} \mathbf{F}_{s,ij}^{C,D} \tag{0.33}$$

where

$$\mathbf{F}_{s,ij}^{C,D} = -\mathbf{I}^{(c)} \left(\rho_{ij} \widetilde{D}_{ij} \mathrm{tr}(\mathbf{d}_{ij}) \right) (Y_{s,j} - Y_{s,i}) \tag{0.34}$$

Following [92], the derivatives of $\mathbf{F}_{s,ij}^{C,D}$ can be written as

$$\begin{aligned} \frac{\partial \mathbf{F}_{s,ij}^{C,D}}{\partial \rho_i Y_{s,i}} &= \frac{\rho_{ij} \widetilde{D}_{ij} \mathrm{tr}(\mathbf{d}_{ij})}{\rho_i}, \\ \frac{\partial \mathbf{F}_{s,ij}^{C,D}}{\partial \rho_j Y_{s,j}} &= -\frac{\rho_{ij} \widetilde{D}_{ij} \mathrm{tr}(\mathbf{d}_{ij})}{\rho_j}, \end{aligned} \tag{0.35}$$

The inter-diffusional energy term (3.15) in total energy equation can be written as

$$\sum_{e \in E_i} \int_V \nabla W_i \cdot \sum_{s=1}^{N_s} (\boldsymbol{J}_s h_s) \, dV = \sum_{j \in K_i} \mathbf{F}_{ij}^{F,D} \tag{0.36}$$

where

$$\mathbf{F}_{ij}^{F,D} = -\mathbf{I}^{(e)} \sum_{s=1}^{N_s} \left(\rho_{ij} h_{s,ij} \widetilde{D}_{ij} \mathrm{tr}(\mathbf{d}_{ij}) \right) (Y_{s,j} - Y_{s,i}) \tag{0.37}$$

Following [92], the derivatives of $\mathbf{F}_{ij}^{F,D}$ can be written as

$$\begin{aligned} &\frac{\partial \mathbf{F}_{ij}^{F,D}}{\partial p_i} = \frac{\mathbf{F}_{ij}^{F,D}}{2 p_{ij}}, \quad \frac{\partial \mathbf{F}_{ij}^{F,D}}{\partial \boldsymbol{V}_i} = 0, \quad \frac{\partial \mathbf{F}_{ij}^{F,D}}{\partial T_i} = -\frac{\mathbf{F}_{ij}^{F,D}}{2 T_{ij}} - \frac{\mathrm{tr}(\mathbf{d}_{ij}) \rho_{ij} \widetilde{D}_{ij}}{2} \sum_{s=1}^{N_s} \left(C_{p,s,ij} \right) (Y_{s,j} - Y_{s,i}) \\ &\frac{\partial \mathbf{F}_{ij}^{F,D}}{\partial p_j} = \frac{\partial \mathbf{F}_{ij}^{F,D}}{\partial p_i}, \quad \frac{\partial \mathbf{F}_{ij}^{F,D}}{\partial \boldsymbol{V}_j} = \frac{\partial \mathbf{F}_{ij}^{F,D}}{\partial \boldsymbol{V}_i}, \quad \frac{\partial \mathbf{F}_{ij}^{F,D}}{\partial T_j} = \frac{\partial \mathbf{F}_{ij}^{F,D}}{\partial T_i} \end{aligned} \tag{0.38}$$

the transformation matrix from primitive variables to conservative variables are

$$\begin{bmatrix} \frac{\partial p}{\partial \rho} = (\gamma - 1)[\boldsymbol{V} \cdot \boldsymbol{V}/2 - e] + RT, & \frac{\partial p}{\partial \rho \boldsymbol{V}} = -(\gamma - 1)\boldsymbol{V}, & \frac{\partial p}{\partial \rho e} = \gamma - 1 \\ \frac{\partial \boldsymbol{V}}{\partial \rho} = -\frac{\boldsymbol{V}}{\rho}, & \frac{\partial \boldsymbol{V}}{\partial \rho \boldsymbol{V}} = -\frac{\mathbf{I}_{3\times3}}{\rho}, & \frac{\partial \boldsymbol{V}}{\partial \rho e} = 0 \\ \frac{\partial T}{\partial \rho} = \frac{1}{\rho C_v}[\boldsymbol{V} \cdot \boldsymbol{V}/2 - e], & \frac{\partial T}{\partial \rho \boldsymbol{V}} = -\frac{\boldsymbol{V}}{\rho C_v}, & \frac{\partial T}{\partial \rho e} = \frac{\mathbf{1}}{\rho C_v} \end{bmatrix} \tag{0.39}$$

The inter-diffusional energy term (3.16) in vibrational-electronic energy equation can be written as

$$\sum_{e\in E_i}\int_V \nabla W_i\cdot\sum_{s=1}^{N_s}(\boldsymbol{J}_s e_{ve,s})\,dV=\sum_{j\in K_i}\mathbf{F}_{ij}^{VE,D} \tag{0.40}$$

where

$$\mathbf{F}_{ij}^{VE,D}=-\sum_{j\in K_i}\left[\mathbf{I}^{(eve)}\sum_{s=1}^{N_s}\left(\rho_{ij}e_{ve,s}\widetilde{D}_{ij}\mathrm{tr}(\mathbf{d}_{ij})\right)(Y_{s,j}-Y_{s,i})\right] \tag{0.41}$$

Following [92], the derivatives of $\mathbf{F}_{ij}^{VE,D}$ can be written as

$$\begin{gathered}\frac{\partial\mathbf{F}_{ij}^{VE,D}}{\partial\rho_i e_{ve,i}}=-\frac{\rho_{ij}\widetilde{D}_{ij}\mathrm{tr}(\mathbf{d}_{ij})}{2\rho_i C_{v,i}^{ve}}\sum_{s=1}^{N_s}(C_{v,s,ij}^{ve})(Y_{s,j}-Y_{s,i}),\\ \frac{\partial\mathbf{F}_{ij}^{VE,D}}{\partial\rho_j e_{ve,j}}=\frac{\partial\mathbf{F}_{ij}^{VE,D}}{\partial\rho_i e_{ve,i}}\end{gathered} \tag{0.42}$$

Jacobian of Source Terms

Following [92], the derivatives of chemical reaction source term $\dot{\omega}_s$ in Eq. (2.31) with respect to species t is

$$\frac{\partial\dot{\omega}_s}{\partial\rho_t Y_t}=\frac{M_s}{\rho Y_t}\left[\sum_{r=1}^{N_r}(\nu_{s,r}''-\nu_{s,r}')(\nu_{i,r}'R_r^f-\nu_{i,r}''R_r^b)\right] \tag{0.43}$$

Following [92], the derivatives of S_{c-v} for the non-preferential model in Eq. (2.38) with respect to conservative variable ρe_{ve} is

$$\frac{\partial S_{c-v}}{\partial\rho e_{ve}}=\frac{\overline{T}_c(a-1)}{T_{ve}\rho C_v^{ve}}\sum_{s=1}^{N_s}\left[e_{ve,s}M_s\sum_{r=1}^{N_r}(\nu_{s,r}''-\nu_{s,r}')\left[\left(\frac{k_r^f}{\overline{\overline{T}}_c}\left(1+\frac{E_{a,r}}{R_u\overline{\overline{T}}_c}\right)\right)\prod_{t=1}^{N_s}\left(\frac{\rho Y_t}{M_t}\right)\right]+\dot{\omega}_s C_{v,s}^{ve}\right] \tag{0.44}$$

The derivatives of translational vibrational energy relaxation S_{t-v} in Eq. (2.40) with respect to conservative variable ρe_{ve} is

$$\frac{\partial S_{t-v}}{\partial\rho e_{ve}}=-\frac{1}{\sum_{mol}\tau_s} \tag{0.45}$$