



Politecnico
di Bari

Repository Istituzionale dei Prodotti della Ricerca del Politecnico di Bari

Development of a multi-GPU solver for atmospheric entry flows with gas-surface interactions

This is a PhD Thesis

Original Citation:

Availability:

This version is available at <http://hdl.handle.net/11589/245802> since: 2022-12-13

Published version

Politecnico di Bari
DOI: 10.6057/poliba/iris/ninni-davide_phd2022

Terms of use:

Altro tipo di accesso

(Article begins on next page)



Politecnico
di Bari

DEPARTMENT OF MECHANICS, MATHEMATICS AND
MANAGEMENT

MECHANICAL AND MANAGEMENT ENGINEERING
PH.D. PROGRAM

SSD: ING-IND/06 - FLUID DYNAMICS

Final dissertation

Development of a multi-GPU solver for atmospheric entry flows with gas-surface interactions

by
DAVIDE NINNI

Supervisors:
Prof. Giuseppe PASCAZIO
Dr. Ing. Francesco BONELLI

Coordinator of the Ph.D. Program:
Prof. Giuseppe P. Demelio

Course XXXV, 01/11/2019 - 31/10/2022

LIBERATORIA PER L'ARCHIVIAZIONE DELLA TESI DI DOTTORATO

Al Magnifico Rettore
del Politecnico di Bari

Il/la sottoscritto/a DAVIDE NINNI nato a MODUGNO (BA) il 16/09/1993 residente a BARI in via UMBERTO MINERVINI 4 e-mail personale d.ninni93@gmail.com e-mail istituzionale davide.ninni@poliba.it iscritto al 3° anno di Corso di Dottorato di Ricerca in INGEGNERIA MECCANICA E GESTIONALE ciclo 35 ed essendo stato ammesso a sostenere l'esame finale con la prevista discussione della tesi dal titolo:

Development of a multi-GPU solver for atmospheric entry flows with gas-surface interactions

DICHIARA

- 1) di essere consapevole che, ai sensi del D.P.R. n. 445 del 28.12.2000, le dichiarazioni mendaci, la falsità negli atti e l'uso di atti falsi sono puniti ai sensi del codice penale e delle Leggi speciali in materia, e che nel caso ricorressero dette ipotesi, decade fin dall'inizio e senza necessità di nessuna formalità dai benefici conseguenti al provvedimento emanato sulla base di tali dichiarazioni;
- 2) di essere iscritto al Corso di Dottorato di ricerca INGEGNERIA MECCANICA E GESTIONALE ciclo 35, corso attivato ai sensi del "Regolamento dei Corsi di Dottorato di ricerca del Politecnico di Bari", emanato con D.R. n.286 del 01.07.2013;
- 3) di essere pienamente a conoscenza delle disposizioni contenute nel predetto Regolamento in merito alla procedura di deposito, pubblicazione e autoarchiviazione della tesi di dottorato nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica;
- 4) di essere consapevole che attraverso l'autoarchiviazione delle tesi nell'Archivio Istituzionale ad accesso aperto alla letteratura scientifica del Politecnico di Bari (IRIS-POLIBA), l'Ateneo archiverà e renderà consultabile in rete (nel rispetto della Policy di Ateneo di cui al D.R. 642 del 13.11.2015) il testo completo della tesi di dottorato, fatta salva la possibilità di sottoscrizione di apposite licenze per le relative condizioni di utilizzo (di cui al sito <http://www.creativecommons.it/Licenze>), e fatte salve, altresì, le eventuali esigenze di "embargo", legate a strette considerazioni sulla tutelabilità e sfruttamento industriale/commerciale dei contenuti della tesi, da rappresentarsi mediante compilazione e sottoscrizione del modulo in calce (Richiesta di embargo);
- 5) che la tesi da depositare in IRIS-POLIBA, in formato digitale (PDF/A) sarà del tutto identica a quelle **consegnate**/inviate/da inviarsi ai componenti della commissione per l'esame finale e a qualsiasi altra copia depositata presso gli Uffici del Politecnico di Bari in forma cartacea o digitale, ovvero a quella da discutere in sede di esame finale, a quella da depositare, a cura dell'Ateneo, presso le Biblioteche Nazionali Centrali di Roma e Firenze e presso tutti gli Uffici competenti per legge al momento del deposito stesso, e che di conseguenza va esclusa qualsiasi responsabilità del Politecnico di Bari per quanto riguarda eventuali errori, imprecisioni o omissioni nei contenuti della tesi;
- 6) che il contenuto e l'organizzazione della tesi è opera originale realizzata dal sottoscritto e non compromette in alcun modo i diritti di terzi, ivi compresi quelli relativi alla sicurezza dei dati personali; che pertanto il Politecnico di Bari ed i suoi funzionari sono in ogni caso esenti da responsabilità di qualsivoglia natura: civile, amministrativa e penale e saranno dal sottoscritto tenuti indenni da qualsiasi richiesta o rivendicazione da parte di terzi;
- 7) che il contenuto della tesi non infrange in alcun modo il diritto d'Autore né gli obblighi connessi alla salvaguardia di diritti morali ed economici di altri autori o di altri aventi diritto, sia per testi, immagini, foto, tabelle, o altre parti di cui la tesi è composta.

Bari, 5/12/2022

Firma

Davide Ninni

Il/La sottoscritto, con l'autoarchiviazione della propria tesi di dottorato nell'Archivio Istituzionale ad accesso aperto del Politecnico di Bari (POLIBA-IRIS), pur mantenendo su di essa tutti i diritti d'autore, morali ed economici, ai sensi della normativa vigente (Legge 633/1941 e ss.mm.ii.),

CONCEDE

- al Politecnico di Bari il permesso di trasferire l'opera su qualsiasi supporto e di convertirla in qualsiasi formato al fine di una corretta conservazione nel tempo. Il Politecnico di Bari garantisce che non verrà effettuata alcuna modifica al contenuto e alla struttura dell'opera.
- al Politecnico di Bari la possibilità di riprodurre l'opera in più di una copia per fini di sicurezza, back-up e conservazione.

Bari, 5/12/2022

Firma

Davide Ninni



Politecnico
di Bari

DEPARTMENT OF MECHANICS, MATHEMATICS AND
MANAGEMENT

MECHANICAL AND MANAGEMENT ENGINEERING
PH.D. PROGRAM

SSD: ING-IND/06 - FLUID DYNAMICS

Final dissertation

Development of a multi-GPU solver for atmospheric entry flows with gas-surface interactions

by

DAVIDE NINNI

Daide Ninni

Referees:

Prof. Thierry E. MAGIN

Prof. Raffaele SAVINO

Supervisors:

Prof. Giuseppe PASCAZIO

Giuseppe Pascazio

Dr. Ing. Francesco BONELLI

Francesco Bonelli

Coordinator of the Ph.D. Program:

Prof. Giuseppe P. Demelio

Giuseppe P. Demelio

Course XXXV, 01/11/2019 - 31/10/2022

”Stay Hungry. Stay Foolish.
And I have always wished that for myself.
And now, as you graduate to begin anew, I wish that for you.”

Steve Jobs

Acknowledgments

First of all, I wish to thank my team, Prof. Pascazio and Francesco, source of productive discussions and inspiration. I also thank Gianpiero, the (unofficial) state-to-state guru.

Thanks to my Italian colleagues: the open space would not be the same without you.

Thanks to my Belgian friends: I know...Parvis Saint Gilles was a better place with me.

Thanks to my mother, for her endless love and support.

Lastly, I wish to express my gratitude to my family and all my friends: my goal, my infinite strength.

*Sincerely,
Davide*

Riassunto esteso

Contesto del progetto

Nel settore dell'ingegneria aerospaziale, uno dei principali obiettivi è la caratterizzazione dei sistemi di rientro in atmosfera. In questo contesto, gli investimenti delle più importanti industrie mondiali sono diventati sempre più ingenti, permettendo lo sviluppo di apparecchiature e macchinari avanzati in grado di riprodurre le condizioni di volo ipersonico. In particolare, visti i regimi di alta entalpia in cui un velivolo spaziale si imbatte, è importante riuscire a fornire una descrizione dei complessi fenomeni termici e fluidodinamici che caratterizzano il flusso insieme ad un'accurata predizione di alcune tra le più importanti informazioni di interesse aerodinamico e termodinamico, come il flusso termico o le distribuzioni della pressione e delle forze di attrito che il corpo deve sostenere durante la fase di rientro. Le alte velocità raggiunte durante la manovra di rientro promuovono la propagazione di onde d'urto in prossimità del corpo, attraverso le quali la maggior parte dell'energia cinetica del fluido viene convertita in energia interna per via di eccitazione dei diversi modi energetici (traslazionale, rotazionale, vibrazionale ed elettronico). Questi sono legati ai gradi di libertà delle particelle che, per via del loro moto caotico, possono interagire le une con le altre. A seconda dei regimi di temperatura, le collisioni possono essere distinte in elastiche ed inelastiche. Come è intuibile, le collisioni inelastiche provocano reazioni chimiche (dissociazione molecolare) che non possono essere trascurate per una corretta descrizione del fenomeno fisico. Ciò rappresenta un ulteriore fattore di accumulo o rilascio di energia, che definisce le ben note condizioni di non equilibrio. Inoltre, per valori estremi di temperatura (solitamente superiori ai 9000 K), gli effetti di ionizzazione diventano rilevanti. Essi provocano la formazione di elettroni liberi all'interno della miscela, rendendo praticamente impossibili le comunicazioni tra il velivolo e la stazione di terra (*blackout*).

Un'accurata descrizione di tutti questi fenomeni è fondamentale nell'ottica di un'appropriata progettazione dei componenti strutturali. Uno tra i principali è lo scudo termico, responsabile della protezione del velivolo dal calore che si sviluppa a causa delle onde d'urto. Questo componente è spesso costituito da particolari materiali, detti ablativi, che permettono dispersione termica per degradazione. Ciò ovviamente comporta una notevole complessità del sistema da analizzare, in cui non è possibile scindere il trattamento della fase gassosa (fluido) da quella

solida (corpo). Infatti, i materiali ablativi favoriscono l'interazione del gas con la superficie, dando origine a fenomeni di catalisi e, appunto, ablazione.

Ad oggi, la sperimentazione di tali flussi risulta molto costosa. Inoltre, è ostico riuscire a replicare tutte le condizioni di volo ipersonico (parametri adimensionali, composizione della miscela) per un intervallo di tempo sufficientemente ampio, il che incentiva l'utilizzo di strumenti numerici per effettuare calcoli predittivi. La potenza computazionale odierna permette di sviluppare codici di calcolo molto efficienti che sfruttano i principali concetti della fluidodinamica computazionale, insieme a modellazione avanzata dei processi di diffusione e delle relazioni termochimiche. Ovviamente, a seconda della complessità del problema il tempo richiesto dalle simulazioni può aumentare da poche ore a diversi giorni.

Metodologia

Nel simulare flussi ipersonici, bisogna tenere conto di tutti i fenomeni di non equilibrio sopra citati. In tal senso, le classiche equazioni di Navier-Stokes (che governano il moto di un fluido in regime continuo) devono essere estese. Estenderle significa introdurre nuovi termini (ed equazioni aggiuntive), mediante l'assunzione di appropriate leggi costitutive, che richiedono un'appropriata modellazione per simulare fenomeni come l'eccitazione vibrazionale o le reazioni chimiche. Per fare questo, si possono impiegare modelli termochimici più o meno complessi, in grado di trattare un generico gas come una miscela multispecie. I due modelli implementati in questa tesi seguono due approcci differenti: il primo è il classico approccio multitemperatura, molto diffuso nella comunità della gasdinamica ipersonica vista la sua semplicità e il suo basso costo computazionale. Questo modello coinvolge un numero contenuto di reazioni chimiche, e aggiunge un'equazione di trasporto per l'eccitazione vibrazionale. Il secondo modello si basa su un approccio cinetico più avanzato, noto come stato-a-stato. Come viene descritto in questa tesi, questo modello può risultare più accurato, ma ad un costo estremamente più elevato rispetto al modello multitemperatura. Infatti, il numero di specie della miscela aumenta vertiginosamente per riuscire a descrivere correttamente la distribuzione delle particelle nei vari livelli di eccitazione energetica.

I classici metodi numerici adottati per la risoluzione di flussi comprimibili si basano su un approccio ai volumi finiti, che permette di trasformare equazioni differenziali in equazioni algebriche, la cui soluzione risulta molto più semplice. A tal fine, le equazioni di governo vengono discretizzate all'interno di un dominio di calcolo costituito da un numero finito di celle, nelle quali viene calcolata la soluzione. Nel simulare flussi ipersonici o, più in generale, supersonici, la soluzione numerica prevede la presenza di discontinuità causate dalla presenza delle onde d'urto nel campo di moto. Per una corretta descrizione numerica degli urti, è necessario implementare nel codice di calcolo formulazioni proprie, capaci di gestire al meglio la ricostruzione della soluzione a cavallo di dette discontinuità. A tal fine, è conveniente combinare schemi di basso e alto ordine, in modo da conferire

allo schema robustezza e di rispettare le condizioni di stabilità numerica. Una volta discretizzati i termini spaziali, si procede con l'integrazione temporale delle equazioni, per la quale si può procedere con due approcci principali. Nell'ottica di un flusso stazionario, è possibile far avanzare la soluzione in una cella indipendentemente dalle altre, aumentando così la velocità di convergenza. Al contrario, volendo risolvere un flusso non stazionario e che, in ogni caso, è caratterizzato da una fase transitoria non trascurabile, la soluzione deve avanzare con lo stesso passo temporale in ogni cella di calcolo.

Un altro aspetto importante del metodo di risoluzione è il tipo di dominio di calcolo impiegato. I più classici presentano una topologia strutturata (ordinata) e conforme al corpo, costituiti quindi da una forma che segue la geometria della parete. In questo modo, è possibile limitare al massimo il numero di punti impiegato per discretizzare il dominio fluido, preservando la libertà di infittire la griglia nelle regioni più critiche, come la parete o le zone degli urti, dove è necessaria elevata accuratezza. Nonostante ciò, questo approccio può risultare inefficiente se si trattano problemi di corpi a geometria complicata (per cui è necessario passare all'impiego di griglie non strutturate) oppure di corpi in movimento, terminologia con la quale si intende far riferimento sia a corpi liberi di muoversi all'interno del fluido, sia a corpi la cui parete recede per via di fenomeni ablativi. Indipendentemente da queste due configurazioni, è necessario riadattare la griglia di calcolo affinché il solutore sia a conoscenza del criterio di spostamento del corpo. Questa procedura può risultare complicata e, soprattutto, costosa, data la necessità di effettuarla in tempo reale durante la simulazione. Per questi motivi, si stanno sfruttando sempre di più approcci ai contorni immersi. L'idea di base di questi metodi è quella di sovrapporre due griglie separate, una che discretizza il dominio fluido e l'altra la superficie e, se necessario, il dominio solido. In particolare, il dominio fluido è discretizzato con una griglia cartesiana, che si traduce in una notevole semplicità dello schema numerico. L'aspetto cruciale riguarda l'imposizione delle condizioni di parete. In tal caso è necessario modificare le equazioni per far sì che il solutore tenga conto della presenza del corpo e, dunque, riesca a calcolare le forze scambiate con il fluido. In questa tesi, viene presentato nel dettaglio un approccio ai contorni immersi in tutti i suoi aspetti.

Infine, è doveroso specificare che risolvere le equazioni di detti flussi può richiedere tempi prolungati, che spesso rappresentano il collo di bottiglia delle simulazioni. Per ovviare a questo limite, si procede con lo sviluppo di un codice di calcolo in ambiente CUDA (Compute Unified Device Architecture). Questo permette di programmare un codice che sfrutta le unità di processo grafiche (GPU) che sono caratterizzate da componenti hardware molto più performanti rispetto ai classici processori centrali (CPU). In tal modo, i tempi di simulazione vengono di gran lunga ridotti, rendendo possibile simulazioni anche le configurazioni di complicate.

Applicazioni

Le più interessanti applicazioni dei flussi ipersonici riguardano le configurazioni geometriche tozze, che riproducono le caratteristiche macroscopiche dei sistemi di rientro classici (capsule). Altre geometrie di particolare interesse sono doppie rampe o doppi cono, che possono essere viste come riproduzioni semplificate di fusoliere di velivoli. Dette configurazioni inducono fenomeni di interazioni tra onde d'urto e strato limite, oggetto di recenti analisi da parte di diversi ricercatori.

In questa tesi, vengono presentati i risultati per corpi tozzi (cilindri) e per configurazioni doppia rampa e doppio cono. Diversi regimi di entalpia sono analizzati, in modo da fornire informazioni dettagliate riguardo l'influenza del non equilibrio sugli aspetti principali dei flussi ipersonici, nella speranza che queste analisi possano risultare un punto di partenza per lo sviluppo di modelli più sofisticati nello scenario delle future missioni spaziali.

Abstract

The understanding of atmospheric re-entry is fundamental in the aerospace engineering field. The heat load experienced by a space vehicle while entering in the atmosphere is extreme and its correct prediction is necessary in the view of an appropriate design of the thermal shield. Technology progression allows to exploit sophisticated facilities able to reproduce the macroscopic features of entry flows. However, high fidelity experimental reproduction are still hard due to two main reasons, namely the cost of an experiment and the difficulty in reproducing each aspect of the flight conditions. This led many companies to invest more and more in numerical tool, representing a valid alternative to provide accurate prediction of interesting information, such as heat flux, pressure distribution or shock stand-off distance. Of course, the development of an efficient numerical tool is not trivial and requires particular attention. Indeed, dealing with hypersonic flows, one must account for 'real gas' effects, known as *non-equilibrium* phenomena.

By the years, many researchers have been devoting efforts to the development of physical models able to describe the correct evolution of the challenging conditions encountered during the re-entry. The high velocities of a space vehicle induce the formation of strong shock waves in front of it, across which the temperature reaches values of the order of 10000 K. It is immediate to understand that these extreme conditions implicate the conversion of the kinetic energy into internal energy, whose total content involves translational, rotational, vibrational and electronic modes. Also, molecular dissociation occurs due to the particle collisions in the shock layer and, if the temperature is large enough, ionization occurs. The latter is a relevant aspect of re-entry flows as the presence of electrons in the mixture is responsible for the well-known *blackout*. For the purpose of heat mitigation, several strategies are adopted. The employment of ablative material for the Thermal Protection System (TPS) has become very common. Thanks to material degradation, the heat flux on the surface of the vehicle is reduced, even if this introduces further complexity in the numerical modeling. The material directly interacts with the species in the mixture, leading the the occurrence of gas-surface interactions (GSI) such as catalysis and ablation.

Classical numerical approaches exploit finite-volume method applied in a body-fitted multi-block grids, very common in Computational Fluid Dynamics (CFD). Nevertheless, when dealing with complex and/or moving geometries, the employment of body-conformal domains can be complicated due to the need of run time

remeshing procedures. In this context, Immersed Boundary Methods (IBM) are suitable for a more versatile numerical solver. Such an approach allows for a unique Cartesian grid generation, that can be refined in the most critical region to increase the accuracy of the numerical solution.

Taking into account all the above mentioned phenomena is a complex task as the numerical model employed must be accurate and cheap at the same time. Indeed, given the huge computational cost required by these kind of numerical simulations, an affordable strategy must be thought in order to speed-up the calculations. Graphics Processing Units (GPUs) provide high performances for general purposes in the scientific field. NVIDIA Corporation is still actively working in the development of efficient interfaces between hardware and software. The most famous one is Compute Unified Device Architecture (CUDA) that allows a very easy interface with basic programming languages such as C/C++ or Fortran. Thanks to GPU programming, very fast simulations are possible even in the most demanding configurations.

All the aforementioned aspects are addressed in this manuscript, which aims at illustrating the main challenges in modeling hypersonic flows. A comparison of the current tools is presented for interesting aerospace applications, with the hope it can inspire further developments for technology progression.

Keywords: *hypersonics, non-equilibrium, multitemperature, Park, State-to-State, shock wave/boundary layer interaction, gas-surface interaction, catalysis, ablation, immersed boundary, CUDA, GPU*

Contents

Abstract	ix
1 Introduction	1
1.1 Atmospheric re-entry	1
1.2 Challenges in modeling hypersonic flows	3
1.3 State of the art	4
1.4 Present contribution	7
1.5 Outline	7
Part I : Characterization of atmospheric entry flows	9
2 Theoretical background and governing equations	11
2.1 Energy modes	11
2.2 Governing equations	13
2.2.1 Mass conservation	15
2.2.2 Momentum balance	16
2.2.3 Energy conservation	17
2.2.4 Equation of state and thermodynamics	18
2.3 Transport properties	20
2.3.1 Viscosity	20
2.3.2 Thermal conductivity	21
2.3.3 Diffusivity	23
2.4 Thermochemical models	24
2.4.1 Multitemperature Park model	24
2.4.2 State-to-State model	28
2.5 Gas-Surface Interaction (GSI)	33
2.5.1 Mass balance	34
2.5.2 Modeling of catalysis	35
2.5.3 Modeling of ablation	36
3 Methodology and numerical formulation	41
3.1 Finite volume approach	41
3.2 Advection fluxes discretization	44

3.3	Diffusive fluxes discretization	48
3.4	Time integration and source terms evaluation	49
3.4.1	First step: fluid dynamics	50
3.4.2	Second step: thermochemistry	51
3.5	Boundary conditions	52
3.5.1	General boundary conditions	52
3.5.2	Wall boundary conditions	53
4	GPU programming	55
4.1	CUDA programming	55
4.2	Example of a kernel and indexing	57
4.3	MPI-CUDA environment	60
4.4	CPU vs GPU	61
4.5	Scalability analysis	63
5	Results	67
5.1	Verification studies	67
5.1.1	0D reactor	67
5.1.2	1D diffusion problem with catalysis	68
5.1.3	1D shock tube	70
5.2	Hypersonic shock wave/boundary layer interaction	72
5.2.1	Low enthalpy double-wedge flow	74
5.2.2	Low enthalpy double-cone flow	76
5.2.3	High enthalpy double-wedge flow	77
5.2.4	High enthalpy double-cone flow	82
5.3	Reactive hypersonic flow past a cylinder	89
5.4	Effects of ablation	93
6	Conclusive remarks	97
A	Analytical solution of the 1D diffusion problem with catalysis	99
B	2D/3D codes verification	101
C	Grid independence studies	103
C.1	SWBLI (sec. 5.2)	103
C.2	Knight case (sec. 5.3)	103
C.3	Ablative test case (sec. 5.4)	106
Part II : Immersed boundary approach for hypersonic flows		109
7	Overview of the Immersed Boundary approach	111

8	Present technique	117
8.1	Generation of the mesh	117
8.2	Data structure	119
8.2.1	Connectivity	119
8.2.2	Ray-tracing procedure	120
8.2.3	Definition of ghost cells	121
8.2.4	Flux formulation	122
8.3	Boundary reconstruction	123
8.3.1	Wall quantity evaluation	126
8.4	Parallelization procedure	127
8.4.1	Partitioning of the domain	127
8.4.2	Data structure re-adaptation	128
9	Results	131
9.1	Perfect gas hypersonic flow over a compression ramp	131
9.2	Perfect gas hypersonic flow past a cylinder	132
9.2.1	Mach 3 flow (M3C)	133
9.2.2	Mach 8 flow (M8C)	134
9.3	Reactive hypersonic flow past a cylinder	139
9.4	Effects of ablation	141
10	Conclusions and perspectives	145
D	Connectivity and data structure information	147
E	Segment/triangle intersection	149
F	Influence of temperature gradients on the velocity field	151
G	Gantt chart of the project	155
	References	172

Chapter 1

Introduction

1.1 Atmospheric re-entry

The problem of the re-entry is one of the strongholds in aerospace engineering. During the re-entry phase, different regimes are distinguished depending on the flight altitude [1]. At very high altitudes, the gas is very rarefied and molecular collisions can be neglected. This is referred to as *molecular regime*. On the other hand, as the altitude decreases, the density of the gas increases so that the collisions are relevant. This is known as *continuum regime*, described by the well-known Navier-Stokes equations. The validity of *continuum* hypothesis can be verified by computing the Knudsen number. This gives an indication of the importance of the collisions in the flow and is defined as ratio between the mean free path and the characteristic length of the system

$$Kn = \frac{l}{L}$$

The assumption of *continuum* is valid for $Kn < 0.01$ and the flow is described by the Navier-Stokes equations. Problems arise in the transition regime (from *molecular* to *continuum*) and accurate methods must be employed. However, this goes beyond the scope of this thesis, focused on the description of the aerothermodynamics of hypersonics flow in the continuum regime.

The high velocity reached by a space vehicle ($\approx 10 \text{ km/s}$) or by a meteoroids ($\approx 70 \text{ km/s}$) leads to the formation of a strong bow shock in front of the body [2,3]. Across such a shock, the temperature reaches extreme values (10000 K) and most of the kinetic energy is converted into internal energy. The total content is composed of more energy mode contributions, namely *translational*, *rotational*, *vibrational* and *electronic*. Moreover, the system stores energy due to the chemical composition changes because of particle collisions (dissociation/recombination) and, for sufficiently large temperatures, ionization. The prediction of electrons' molar fraction is fundamental. The presence of electrons in the mixture provokes what is known as *blackout*: the plasma layer reflects or attenuates radio waves transmitted by the vehicle, making impossible telecommunications with the ground station.

Inside the shock, the system relaxes towards equilibrium according to the mode relaxation time:

$$\tau_{tr} < \tau_{rot} < \tau_{vib} < \tau_{chem} < \tau_{el} < \tau_{ion}$$

where the subscripts stand for 'translational' (*tr*), 'rotational' (*rot*), 'vibrational' (*vib*), 'chemical' (*chem*), 'electronic' (*el*) and 'ionization' (*ion*). Depending on the characteristic time of the flow (τ_{flow}), three main conditions can be defined:

- the flow is considered *frozen* if the mode excitation is too slow when compared to the characteristic time scale of the flow ($\tau \gg \tau_{flow}$);
- the flow is considered at *equilibrium* if the mode excitation is too fast when compared to the characteristic time scale of the flow ($\tau \ll \tau_{flow}$);
- the flow is considered at *non-equilibrium* if the mode excitation occurs at a time scale comparable to that of the flow ($\tau \approx \tau_{flow}$).

Given the extreme temperature values, the heat mitigation is also one of the main issues. Especially in the case of space vehicles, the wall temperature must be kept at a low value to avoid damage. However, the heat flux prediction is fundamental also in the framework of meteoroids and space debris flow analysis. This induces the formation of a boundary layer, where atomic particles tend to recombine due to the lower temperatures. Moreover, many Thermal Protection Systems (TPS) are characterized by cooling materials that promote gas-surface interactions (GSI) in order to reduce the heat load by ablation [4].

The experimental reproduction of such flows is nowadays very complicated [5]. The current technology evolution allows to employ powerful facilities to simulate challenging flight conditions. One of the most famous is the Plasmatron facility at the von Karman Institute for Fluid Dynamics (VKI) shown in Fig. 1.1 [6], able to reach temperatures up to 10000 K for different gas mixtures (nitrogen, oxygen, argon, carbon dioxide, and others). Other well known facilities are the SCIROCCO Plasma Wind Tunnel at CIRA (Italy) [7] or the Arc Jet at NASA-AMES Research Center, designed to simulate very high Mach numbers (≈ 12) and sustain extreme heat fluxes. Unfortunately, experiments are always very costly and require an advanced user experience. Moreover, experiments are always affected by errors of different nature that hinder the reproduction of the real physical conditions and limit the quantity of obtained information (wall distributions, shock stand-off distance, etc.). For these reasons, space agencies and research centers are reducing the investment on experimental campaigns in favor of theoretical and numerical studies of atmospheric entry flows by means of the Computational Fluid Dynamics (CFD).

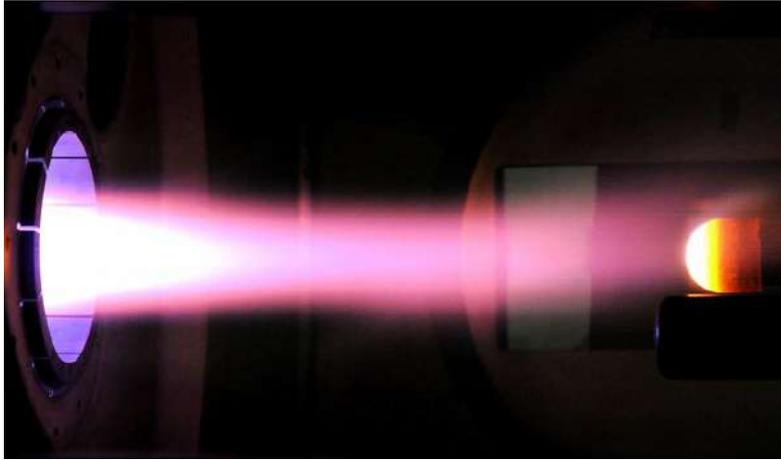


Figure 1.1: Plasmatron facility at VKI.

1.2 Challenges in modeling hypersonic flows

A tool capable to manage all the fundamental aspects of hypersonic flows is difficult to develop as the classical system of governing equations must be extended to account for non-equilibrium. Lot of current numerical tools rely on the thermochemical models developed during the last decades. One of the most commonly used in the hypersonics community is the multitemperature model (mT) proposed by Park [8]. In this model, the characteristic translational and rotational temperatures are very low (< 3 K) and quite similar with each other. Hence, the translational and rotational modes of freedom are considered at equilibrium and a single temperature T can be defined for both of them. On the other hand, the excitation of the other internal modes requires a larger amount of energy transfer. To give an indication, the characteristic vibrational temperature of the molecular nitrogen is of the order of 3000 K, whereas the electronic one is about 70000 K. Hence, these modes are excited following different temperatures, namely the vibrational temperature (T_{vib}) and the electronic temperature (T_{el}), and the population of the internal states follow a Boltzmann distribution [8,9]. Actually, for the applications presented in this thesis, the temperature reached in the shock layer is never sufficiently large to induce electronic excitation, nor ionization. Namely, in the model employed in this work the electronic energy mode is neglected. Hence, the thermal non-equilibrium condition is identified by the condition $T \neq T_{vib}$. Moreover, particle interactions give rise to variations in the chemical compositions, leading to thermochemical non-equilibrium. As a consequence of such condition, equations for the mass conservation and vibrational energy transport are needed to describe the non-equilibrium. The mechanisms considered in this model involve molecular dissociation, atomic recombination and exchange reactions for hetero-nuclear molecules such as NO or CO. The reaction rates follows Arrhenius relations [10–12]. Specifically, they are computed at a mean temperature in or-

der to link the vibrational excitation to the chemical activity. These assumptions make this model very cheap and easy to implement, justifying its wide application in the hypersonics community.

Nevertheless, the assumption of a Boltzmann distribution is not always suitable. Indeed, large deviations have been observed in the actual reaction rates with respect the Arrhenius ones [13–15], convincing that the description of the population of the internal vibrational states need a proper treatment. The State-to-State approach (StS) represents an alternative to the classical mT model [15] to accomplish a detailed description of non-equilibrium. The basis of the StS model is the the assumption for which a molecule is considered as an anharmonic oscillator. This idea allows to immediately know the vibrational energy of each level, with no need of additional equations. On the other hand, each level is treated as a pseudo-species, increasing considerably the number of mass conservation laws needed. A generic 5 species air mixture (N_2 , O_2 , NO, N, O) accounts for 68 levels for N_2 and 47 for O_2 , for a total of 118 species and about 10000 chemical processes. It is immediate to understand that the computational cost required by StS simulations is huge, limiting its application to mostly 1D configurations [16–19], being 2D and 3D configurations very onerous [20–24].

1.3 State of the art

By the years, many CFD studies have been carried out to simulate atmospheric entry flows and validate numerical tools. Several institutions collaborated for the analysis of the flow over a double-wedge [25], supplemented by experimental measurements of wall heat flux and pressure [26]. Double-wedge or double-cone geometries are very attractive for the aerospace community due to the complex shock wave/boundary layer interaction (SWBLI) that characterized the flows over these configurations. The efforts devoted by the involved researchers resulted in important findings. In particular, relevant differences in the boundary layer separation were observed between 2D and 3D simulations, convincing that these flows are affected by three-dimensional effects, further studied in next articles [27, 28]. Moreover, it was also found that non-equilibrium becomes relevant only for a high free stream enthalpy (≈ 8 MJ/kg). On the other hand, a lower free stream enthalpy (≈ 2 MJ/kg) seems unaffected by non-equilibrium, given the low temperature reached. Despite this, an important aspect of the low enthalpy regime is the periodicity of the flow shown by Durna and Celik [29]. Such an aspect was never fully analyzed for the high enthalpy regime. The experimental measurements were performed in the transient state and no information about the steadiness of the flow was known.

Similar investigations were carried out for the double-cone geometry [30]. In such a case, the flow is expected to be stationary whether the free stream enthalpy is high or low. Also in this case, numerical simulations aimed at reproducing

experimental measurements [31]. The main issue of these flows is the difficulty in capturing the correct separation extent for high enthalpy flows. Indeed, for low free stream enthalpy values (≈ 4 MJ/kg), numerical results are in a good agreement with the expectations, as found by Nompelis and Candler [32]. As the free stream enthalpy increases, the error in the prediction of the separation extent becomes more and more relevant, reaching very large deviations for a stagnation enthalpy of 10 MJ/kg. Possible reasons were addressed to the incorrect evaluation of the free stream conditions, but a complete explanation remained still unknown.

Another attractive configurations are cylinders or spheres as they represent simplified models of re-entry vehicles: their bluff shape promotes strong bow shocks that accelerate the occurrence of non-equilibrium phenomena. Considerable efforts were devoted by many researchers for a hypersonic flow past a cylinder [30] in order to provide accurate predictions of the wall quantity distributions (pressure and heat flux). In this study, it was found that the evaluation of the wall pressure is very accurate, whereas the temperature gradients at wall are very sensitive to several issues (spatial resolutions, numerical scheme, wall model). Indeed, a catalytic wall was also analyzed and it was found that wall kinetics can be relevant in the prediction of the heat flux. Numerical simulations for the flow past a sphere have been conducted to reproduce the experimental findings of Nonaka et al. [33]. In this work, the authors compared the stand-off distance of the bow shock forming in front of a spherical sample, arguing that the position of the shock strongly depends on the non-equilibrium impact. Frozen flows often present a larger shock stand-off distance with respect to non-equilibrium flows. This result was confirmed by Colonna et al. [21], who in turn found that the StS model provides an even more accurate prediction of the shock stand-off distance with respect to the mT model. Relevant deviations between StS and mT approaches were also found for the flow through a nozzle [34, 35], convincing the community that the multitemperature model presents intrinsic limitations in the treatment of the non-equilibrium. Nevertheless, the huge computational cost required by StS simulations does not permit to perform complex configuration analysis such as 3D geometries or unsteady flows. A further analysis to simplify this kinetic model is needed to assess whether the number of levels can be reduced or the population can be modeled by means of reduced models.

Bluff body shapes are often object of investigations also in the framework of ablative systems. Olynick et al. [36] and Park [37] simulated the atmospheric re-entry of the Stardust vehicle, reporting similar results, thus assessing that numerical calculations can supplement the design process of the vehicle. More in detail, they found that an ablating material considerably reduces the heat flux and the shear stresses at wall, while the pressure remains unaffected. Nevertheless, the wall distributions are strongly sensitive to the gas-surface interaction model employed. As shown by Chen and Milos [38], the massblowing rate profile depends on the treatment of the wall and on the mechanisms occurring between the gas and the material. In particular, if the temperature is large enough to

induce oxygen dissociation, oxidation mechanism is negligible with respect to the nitridation. However, there is still lot of uncertainty about the numerical results available in the literature and they need to be supplemented by more experimental campaigns in the view of a full understanding of the phenomena involved.

Applications to turbulent configurations have been also investigated by the years. Reynolds-Averaged-Navier-Stokes (RANS) and other classical models have been employed to many configurations involving spheres [39], compression ramps [40], flat plates [41] and propulsion systems [42]. Despite the overall good agreement with high-fidelity simulations such as Large Eddy Simulations (LES) or Direct Numerical Simulations (DNS), these turbulent models are under-resolved. For the purpose of fully understanding turbulence phenomena, more complex approaches (LES or DNS) must be adopted. Of course, this leads to a very onerous computational cost, and the geometries are strongly simplified. Common configurations are shear flows [43] or flat plates [44–46]. In this context, GPU programming can be a solution to reduce the simulation time in the framework of DNS simulations. However, in this thesis only laminar configurations are analyzed, turbulent flow simulations representing a perspective for future works.

Lastly, it is noteworthy that, especially in the case of ablation and recessing surfaces, the need of remeshing remains a critical drawback. As the wall moves in space, the computational grid needs to be adapted to the motion of the boundary. Also, even more relevant issues arise when dealing with non-uniform recession (for instance, to simulate the destruction of an asteroid entering in the atmosphere). In this scenario, an Immersed Boundary approach comes to rescue. This allows to employ non-conformal Cartesian grids to simulate generic flows [47] by applying a forcing term in the vicinity of the wall to ensure the proper boundary conditions. The first applications came out at the end of the XX century thanks to the works by Peskin [48–50] who mainly focused on incompressible flows. This formulation exploited a forcing in the first fluid cell next to the body to simulate the flow past deformable bodies. The results were satisfactory but the method employed presented some limitations due to the fact that it was not suitable for rigid bodies. Basically, a rigid boundary can be obtained by decreasing the deformability function, but this would make to problem stiff and numerically hard. Next authors such as Basdevant and Sadourny [51], Briscolini and Santangelo [52] and Goldstein et al. [53] tried to optimize this methodology for rigid bodies applications. In the first two works, the main idea was to distribute the forcing term in a narrow band next to the wall (typically 3 or 4 fluid points) but this approach generates spurious oscillations in the numerical solution. For the purpose of reducing the oscillations, Goldestein et al. [53] proposed a central finite difference method to apply the forcing term. Nevertheless, it required a spreading of the function along the wall, making the formulation problem-tuning and reducing its computational efficiency. Important improvements were proposed by Modh-Yusof [54], Fadlun [55] and Balaras [56], that opened the doors to a wider range of applications also in the compressible regime. The philosophy behind their formulation was the possibility

of applying the forcing term in the discrete domain, namely not directly introduced in the equations. The desired boundary condition was satisfied imposing an interpolation along a certain direction (Cartesian or wall normal) with respect to the body. One of the first applications of the IB approach to compressible flows was presented by De Tullio et al. [57] who generalized the formulation proposed by Modh-Yusof [54] to ensure the desired boundary conditions for weakly supersonic flows. In recent years, several immersed boundary methods have been developed to facilitate the simulations of hypersonic flows around complex bodies [58]. Different kind of applications are available in the literature, but still presenting relevant limitations. These are explained in detail in the second part of this thesis.

1.4 Present contribution

The aim of this project is the development of a high efficiency computational tool to perform 2D/3D numerical simulations of atmospheric entry flows. Both the mT and the StS model are implemented to model thermochemical non-equilibrium. In order to reduce the computational cost of the simulations, the algorithm is extended to CUDA environment to allow GPU executions. This is an affordable strategy to speed-up the calculation. Gas-surface interactions models are also included to accurately treat ablative materials and catalytic walls. As explained in detail in chapter 2, the code is coupled with the well known MUTATION⁺⁺ library, also to take advantage of advanced transport models.

In this view, given the complexity in the treatment of moving bodies as in the case of recessing walls, an Immersed Boundary approach (IB) is exploited, such a way complex geometries can be simulated with minimal mesh generation efforts. Results from IB simulations are compared to those obtained from the classical body-fitted approach (BF).

1.5 Outline

This manuscript is divided into two macro parts, organized as follows.

1. The first part focuses on the fundamental characterization of hypersonic flows, covering all the main aspects. Chapter 2 is dedicated to the description of the theoretical background at the basis of hypersonics, involving the governing equations, the models employed to handle non-equilibrium, and the gas-surface interactions. In chapter 3, the description of the methodology is illustrated, focusing on the body-fitted numerical approach to discretize the equations. Chapter 4 illustrates some detail about GPU programming, useful to speed-up the calculation. Basic examples of CUDA kernels are given and the performance of the solver are analyzed. The discussion of the results is presented in chapter 5, where different cases are investigated. The

shock wave/boundary layer interaction is simulated for double-wedge and double-cone configurations, being very attractive for aerospace engineering. Moreover, bluff body simulations (cylindrical samples) are performed to test the robustness of the code when dealing with strong bow shocks. Gas-surface interaction mechanisms are also considered. Chapter 6 is dedicated to the conclusions.

2. The second part is mainly focused on the extension of the solver to an alternative numerical methodology, namely the Immersed Boundary approach. A background and a general description of this method is given in chapter 7, whereas chapter 8 presents the details of the present solver, explaining the most practical procedures (reconstruction, parallelization, data structure). Results are discussed in chapter 9, starting from simple cases and finally performing simulations on two tests already presented in the first part of the thesis, to provide a comparison with the body-fitted approach of the code. Conclusions are drawn in chapter 10.

Part I :
**Characterization of atmospheric
entry flows**

Chapter 2

Theoretical background and governing equations

This chapter is devoted to the description of the fundamental phenomena occurring in high enthalpy flows, involving energy modes excitation, chemical activity and gas-surface interactions. A typical scenario of such challenging conditions is reported in Fig. 2.1. As soon as a molecule travels across the bow shock forming in front of the body, molecular dissociation and vibrational excitation are induced. Moreover, if the temperature is large enough (> 9000 K), the concentration of free electrons starts increasing, promoting ionization phenomena. Due to the lower temperature regime in the boundary layer, atomic recombination can occur at wall. In some cases, the material can interact with the gas in the ablative layer. This leads to the formation of carbon species due to the material decomposition.

All these aspects are illustrated in the next sections.

2.1 Energy modes

Dealing with high enthalpy flows, the total energy content is composed of more internal contributions [9], each one linked to the different energy forms (modes). The most general case is the one of a polyatomic molecule. It is composed of two masses (atoms) and four different energy modes can be excited. They are listed below and sketched in Fig. 2.2.

- *Translational mode*: the molecule is free to translate in all the three main directions (3 degrees of freedom).
- *Rotational mode*: the molecule can rotate around its center of mass, being the rotational mode around the inter-nuclear axis negligible (2 degrees of freedom for diatomic and polyatomic linear molecules, 3 for non linear molecules).
- *Vibrational mode*: the molecule can vibrate along the direction of elastic force (2 degrees of freedom).

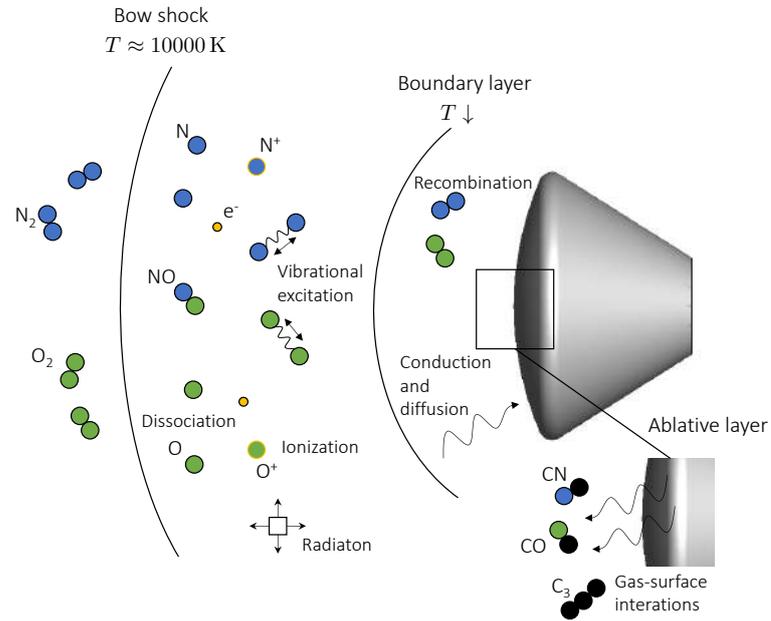


Figure 2.1: Typical scenario of hypersonic flows interacting with an ablative thermal protection system (not in scale).

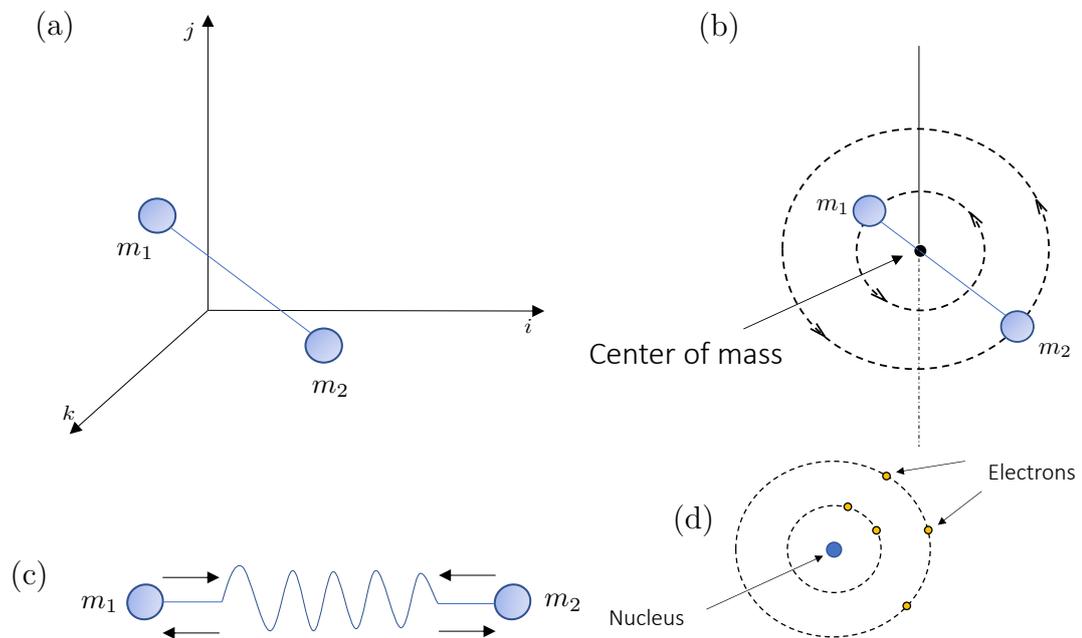


Figure 2.2: Contributions of energy modes, adapted from [9]: (a) translational, (b) rotational, (c) vibrational, (d) electronic.

- *Electronic mode*: the molecule is electronically excited due to the kinetic and potential energy of electrons in orbit.

A fifth energy contribution is given by the formation enthalpy, evaluated at a certain reference temperature. In the case of a molecule, all these contributions are taken into account. Nevertheless, atoms are characterized by just one point mass, and the rotational and vibrational contributions can not be defined. For the same reason, only translational and formation contributions are considered for electrons, for which also the electronic mode is of course not defined. When these modes are excited at a unique temperature T , thermal equilibrium condition holds. Furthermore, if chemical equilibrium is considered, a *local thermodynamic equilibrium* (LTE) condition can be defined. It means that for given values of pressure, temperature and elemental molar fractions, the mixture composition, along with thermodynamic and transport properties, are uniquely defined.

One of the most representative mixture of atmospheric entry flows is the 11 species air mixture (N_2 , O_2 , NO , N , O , N_2^+ , O_2^+ , NO^+ , N^+ , O^+ , e^-). It involves charged species arising from the interaction between heavy species and electrons. The chemical composition (molar fractions) under the assumption of LTE is reported in Fig. 2.3 for a pressure of 1 atm (dashed lines represent charged species). As expected, electrons' molar fraction starts becoming relevant for temperatures above 9000 K, even if their molar fraction is already 0.1% at 6000 K. Of course, molecular charged species have no relevance in the composition as they dissociate before ionization. On the other hand, atomic nitrogen and oxygen strongly interact with electrons giving rise to N^+ and O^+ . Preserving the assumption of LTE, it is worth analyzing the different contributions of the specific enthalpy. They are illustrated in Fig. 2.4. It is evident that the formation and translational contributions are predominant, whereas electronic enthalpy starts increasing after 9000 K. Of course, rotational and vibrational contributions increase with the temperature, becoming null after molecular dissociation is complete. Such graphs are useful to have an indication of the validity of each assumption behind the thermochemical models commonly used.

In this thesis, ionization phenomena are neglected as the temperature is never sufficiently high to induce such phenomenon.

2.2 Governing equations

Fluid dynamics problems are governed by the well-known Navier-Stokes equations, thus mass conservation, momentum balance and energy conservation. When dealing with high-temperature flows, the classical form of such equations can not describe all the phenomena involved and they need to be extended to account for chemical mechanisms and vibrational excitation. The number of equations depends on the thermochemical model employed, namely the classical multitemperature (mT) model or the detailed State-to-State (StS) model. As it is explained

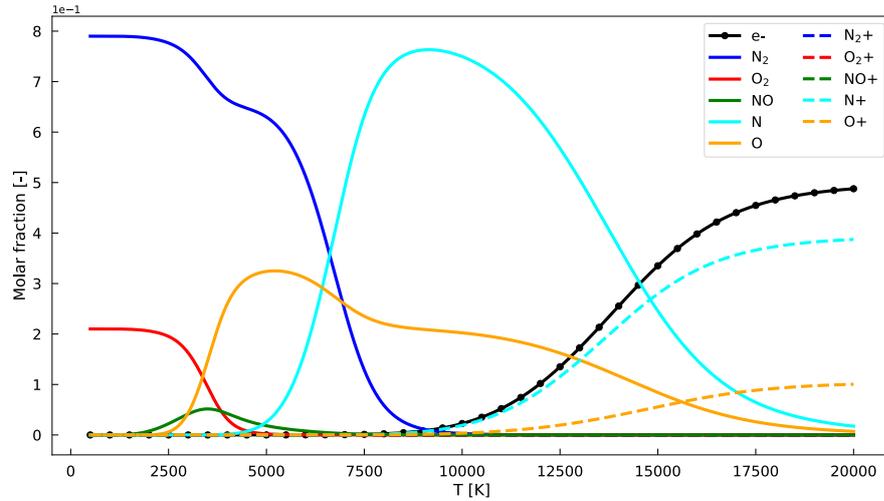


Figure 2.3: Chemical composition in terms of molar fractions for an 11 species air mixture ($p = 1$ atm).

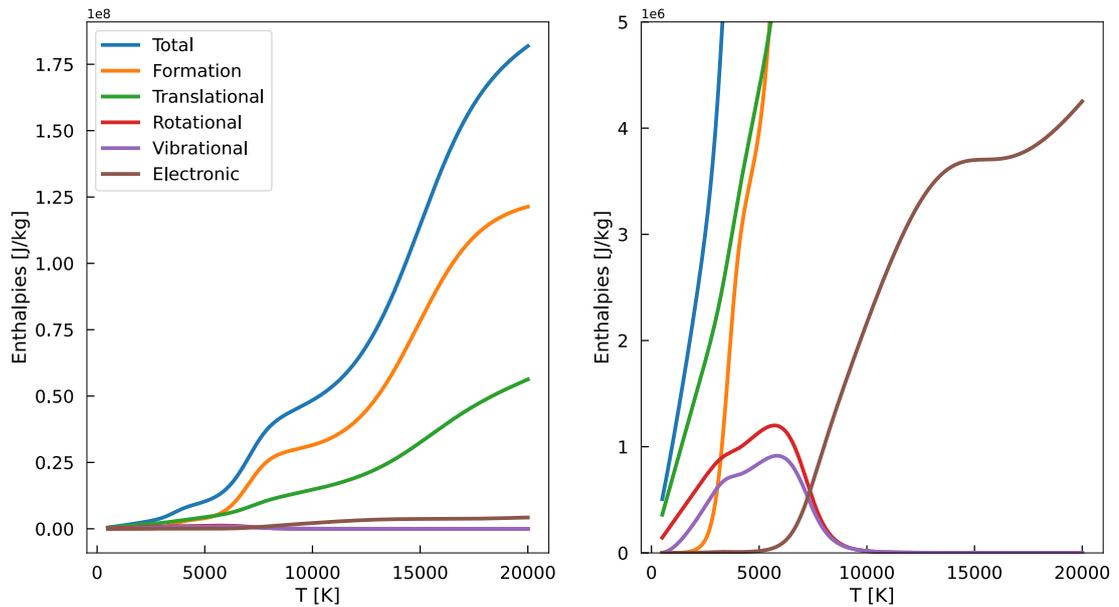


Figure 2.4: Specific enthalpy contributions for an 11 species air mixture ($p = 1$ atm). Zoom in on the right to emphasize minor contributions.

in section 2.4, only the first one needs the solution of a transport equation for the vibrational energy. On the other hand, the StS model requires a continuity equation for each vibrational level. Next subsections are devoted to the description of all the governing equations [59], considering a 5 species neutral air mixture.

2.2.1 Mass conservation

Dealing with multicomponent gases, a continuity equation is written for each species in the mixture. Anticipating a concept better described in section 2.4.2, each vibrational level of the molecules is treated as a separate species in the StS approach, so that continuity must hold for each of them. For simplicity, in the following subscripts s and l will refer to species and vibrational level respectively. So, the continuity equation reads:

$$\boxed{\frac{\partial \rho_{s,l}}{\partial t} + \nabla \cdot (\rho_{s,l} \mathbf{u} + \mathbf{j}_{s,l}) = \dot{\omega}_{s,l}} \quad (2.1)$$

where $\rho_{s,l}$ is the density of vibrational level l of the species s and $\mathbf{u} = (u, v, w)$ is the velocity vector of the fluid. The distinction between the vibrational level and the species is mandatory since the number of vibrational levels depends on the thermochemical model employed. In the mT model the vibrational energy modes are described by a Boltzmann distribution: hence, the number of species is $N_s = 5$ and the number of vibrational levels is $V_s = 1$ for all of them. On the contrary, the StS model is based on a detailed description of the vibrational levels distribution. More in detail, molecular nitrogen is composed of 68 levels ($V_{N_2} = 68$), whereas 47 levels are taken into account for molecular oxygen ($V_{O_2} = 47$); for all the other species, only the ground state is considered, thus $V_N = V_O = V_{NO} = 1$. This gives an indication of the computational cost encountered if the StS model is employed; since each vibrational level is treated as a separate species, 118 species are needed to describe the non-equilibrium for a StS kinetics model. This obviously limits its application mostly to 0D/1D configurations [16–19], with few 2D/3D exceptions [20–24]. In the following, we will refer to each vibrational level simply as species.

The additional terms in Eq. (2.1) are the diffusive flux ($\mathbf{j}_{s,l}$) and the source term ($\dot{\omega}_{s,l}$). The latter, is responsible for the mass fraction variation of the species in the mixture: its evaluation depends on the thermochemical model employed and is presented in section 2.4. Instead, the diffusive flux is linked to the phenomenon of *mass diffusion*. In multicomponent gases, each species is characterized by a certain mass fraction ($Y_{s,l}$) or molar fraction ($\mathcal{X}_{s,l}$), defined as:

$$Y_{s,l} = \frac{\rho_{s,l}}{\rho} \quad (2.2a)$$

$$\mathcal{X}_{s,l} = \frac{\mathcal{M} Y_{s,l}}{\mathcal{M}_s} \quad (2.2b)$$

where \mathcal{M}_s is the molar mass of the species s (it is the same for all the internal levels in case of the StS model) and \mathcal{M} and ρ are the molar mass and the density of the mixture, respectively. In case of the StS model, one can define the species quantities as the sum over all the levels. For instance, species mass fractions are defined as $Y_s = \sum_l Y_{s,l}$. Obviously, to ensure the global mass conservation, it has

to result $\sum_s \sum_l Y_{s,l} = \sum_s \sum_l \mathcal{X}_{s,l} = 1$ and $\rho = \sum_s \sum_l \rho_{s,l}$, and the molar mass of the mixture can be computed as follows:

$$\mathcal{M} = \frac{1}{\sum_s \sum_l Y_{s,l} / \mathcal{M}_s} \quad (2.3)$$

Because of concentration gradients, each species tends to diffuse in space to minimize the difference in terms of mass fraction in the surrounding environment [9]. This occurs at a certain speed known as *diffusion velocity*, indicated as $\mathbf{u}_{s,l}^D$, so that $\mathbf{j}_{s,l} = \rho_{s,l} \mathbf{u}_{s,l}^D$. The diffusion velocities can be evaluated by solving the following system of N_s^2 equations [60]:

$$\nabla \mathcal{X}_s = \sum_{r=1}^{N_s} \frac{\mathcal{X}_s \mathcal{X}_r}{\mathcal{D}_{sr}} (\mathbf{u}_s^D - \mathbf{u}_r^D) + (Y_r - \mathcal{X}_r) \frac{\nabla p}{p} + \frac{\rho}{s} \sum_{r=1}^{N_s} Y_s Y_r (\mathbf{f}_s - \mathbf{f}_r) \quad (2.4)$$

where subscripts s and r indicate the s -th and r -th species in the mixture. The inversion of such system is too demanding considering that it should be generally performed in each spatial point of the grid at each time step. In this scenario, the approximation proposed by Hirschfelder and Curtiss [61] comes to rescue. It is based on the diffusion of each species into the mixture. Hence, the diffusion velocity can be evaluated by using Fick's law:

$$\rho_{s,l} \mathbf{u}_{s,l}^D = -\rho \mathcal{D}_s \nabla Y_{s,l} + \mathbf{V}_c \quad (2.5)$$

where $\mathcal{D}_s = \frac{1 - \mathcal{X}_s}{\sum_{r \neq s} \mathcal{X}_r / \mathcal{D}_{sr}}$ is an equivalent diffusion coefficient, whose evaluation is discussed in section 2.3.3; $\mathbf{V}_c = \sum_s \sum_l Y_{s,l} \mathcal{D}_s \nabla Y_{s,l}$ is a correction velocity to ensure mass conservation [62]. In this way, only N_s diffusion velocities must be computed.

Finally, note that to recover the global mass conservation one can sum up all the species continuity equations and it must result that $\sum_s \sum_l Y_{s,l} \mathbf{u}_{s,l}^D = \sum_s \sum_l \dot{\omega}_{s,l} = 0$.

2.2.2 Momentum balance

Formally, momentum is not influenced by the thermochemical non-equilibrium: temporal, advective and diffusive terms in the equations are the same of a non-reacting mixture. The vectorial equation reads:

$$\boxed{\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \cdot \mathbf{u} + p \bar{\mathbf{I}} - \bar{\boldsymbol{\sigma}}) = 0} \quad (2.6)$$

where p is the pressure, $\bar{\mathbf{I}}$ is the identity tensor and $\bar{\boldsymbol{\sigma}}$ is the viscous stress tensor, that reads:

$$\bar{\boldsymbol{\sigma}} = \mu \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right] - \frac{2}{3} \mu (\nabla \cdot \mathbf{u}) \bar{\mathbf{I}} \quad (2.7)$$

being μ the dynamic viscosity of the mixture. This is evaluated basing on the single species transport properties, as it is explained in section 2.3.

2.2.3 Energy conservation

The energy conservation equation is formally similar to the classical version of a non-reacting mixture. Nevertheless, additional terms appear in the definition of the heat flux. The equation reads:

$$\boxed{\frac{\partial \rho E}{\partial t} + \nabla \cdot [(\rho E + p) \cdot \mathbf{u} - \mathbf{u} \cdot \bar{\boldsymbol{\sigma}} + \mathbf{q}] = 0} \quad (2.8)$$

where E is the total energy per unit mass and \mathbf{q} is the total heat flux. For thermochemical non-equilibrium flows, this term is composed of different contribution [63]:

$$\mathbf{q} = -\kappa_{tr} \nabla T - \underbrace{\sum_{m=1}^{N_m} \kappa_{vib,m} \nabla T_{vib,m}}_{\text{Only mT model}} + \sum_{s=1}^{N_s} \sum_{l=1}^{V_s} \rho_{s,l} h_{s,l} \mathbf{u}_{s,l}^D \quad (2.9)$$

In Eq. (2.9), the first contribution represents the heat flux occurring due to conduction, expressed by the classical Fourier's law. κ_{tr} and T are the roto-translational thermal conductivity and temperature of the mixture, respectively. The same law is applied for the second term, expressing the conductive heat flux due to vibrational excitation. In this case $\kappa_{vib,m}$ and $T_{vib,m}$ are the vibrational thermal conductivity and temperature of molecule m , respectively (N_m is the number of molecules in the mixture). As for the viscosity, the evaluation of the thermal conductivities depends on the single species transport properties and is discussed in section 2.3.2. It is noteworthy that this term is considered only in case of the mT model, since the vibrational temperature loses its physical meaning in the StS approach. More details are given in section 2.4.2. The third contribution represents the heat flux due to the mass diffusion and depends on the enthalpy per unit mass of species s in the level l , $h_{s,l}$. This includes roto-translational, vibrational and formation contributions. Finally, it must be highlighted that in Eq. (2.9), the contribution due to thermal diffusion is neglected. This is a reasonable assumption for most of the applications in hypersonics [9, 64], with the few exceptions [65].

As mentioned at the beginning of section 2.2, a transport equation is needed to evaluate the vibrational energy in case of the mT model. In this case, subscript l can be neglected and the equation reads:

$$\boxed{\frac{\partial \rho_m e_{vib,m}}{\partial t} + \nabla \cdot [(\rho_m e_{vib,m}) \cdot \mathbf{u} + \mathbf{q}_{vib,m}] = \dot{\omega}_{vib,m}} \quad (2.10)$$

where $e_{vib,m}$ is the vibrational energy per unit mass of the molecule m and $\dot{\omega}_{vib,m}$ is the vibrational source term that accounts for the effect of chemistry and vibrational

relaxation. Its evaluation is explained in section 2.4.1. The vibrational heat flux reads:

$$\mathbf{q}_{vib,m} = -\kappa_{vib,m} \nabla T_{vib,m} + \rho_m e_{vib,m} \mathbf{u}_m^D \quad (2.11)$$

Note that Eq. (2.10) is solved for each molecule in the mixture.

2.2.4 Equation of state and thermodynamics

In order to close the system, the equation of state is solved, assuming that each species behaves as a perfect gas:

$$p = \sum_{s=1}^{N_s} p_s = \sum_{s=1}^{N_s} \rho_s R_s T \quad (2.12)$$

being $R_s = \mathfrak{R}/\mathcal{M}_s$ the gas constant, with $\mathfrak{R} = 8.31451$ J/mol K the universal gas constant and \mathcal{M}_s the molar mass of the species s . The temperature is computed by inverting Eq. (2.12), whereas the pressure arises from the relation with the energy. If the temperature is below 9000 K, ionization phenomena do not take place and the electronic energy can be neglected. So, the internal energy per unit mass can be decomposed in three contributions [8,9]:

$$e_s = \begin{cases} e_s^{tr} & \text{for atoms} \\ e_s^{tr} + e_s^{rot} + e_s^{vib} & \text{for molecules} \end{cases} \quad (2.13)$$

where the superscripts stand for *translational*, *rotational* and *vibrational* respectively. While the vibrational contribution depends on the thermochemical model, the evaluation of the translational and rotational terms in Eq. (2.13) is based on the theorem of equipartition of energy from the kinetic theory [9]. They contribute with $1/2 R_s T$ for each degree of freedom, so that:

$$e_s = \begin{cases} \frac{3}{2} R_s T & \text{for atoms} \\ \frac{3}{2} R_s T + R_s T + e_s^{vib} & \text{for molecules} \end{cases} \quad (2.14)$$

The following formulation refers to molecular species, being the atomic species case straightforward. Denoting with h the specific enthalpy (in turn composed of three contributions) it must hold that $h_s = e_s + R_s T$:

$$\overbrace{h_s^{tr} + h_s^{rot} + h_s^{vib}}^{h_s} = \overbrace{\frac{3}{2} R_s T + \underbrace{R_s T}_{e_s^{rot}} + e_s^{vib}}^{e_s} + R_s T \quad (2.15)$$

As mentioned above, each species is considered as a thermally perfect gas; then the specific heats are $c_{v,s} = \partial e_s / \partial T$ and $c_{p,s} = \partial h_s / \partial T$. Deriving Eq. (2.15) with respect to the temperature one obtains:

$$c_{p,s}^{tr} + c_{p,s}^{rot} + c_{p,s}^{vib} = \underbrace{\frac{3}{2}R_s}_{c_{v,s}^{tr}} + \underbrace{R_s}_{c_{v,s}^{rot}} + c_{v,s}^{vib} + R_s = \frac{7}{2}R_s + c_{v,s}^{vib} \quad (2.16)$$

Using Mayer's relation, $c_p = c_v + R$ and Eq. (2.16) becomes:

$$\frac{7}{2}R_s + c_{p,s}^{vib} = \frac{7}{2}R_s + c_{v,s}^{vib} \quad (2.17)$$

that means $c_{p,s}^{vib} = c_{v,s}^{vib}$, namely $h_s^{vib} = e_s^{vib}$. This is an important result because, focusing only on the translational and rotational contributions, it allows a generalization of the expression for the enthalpy, that is:

$$h_s^{tr} + h_s^{rot} = \bar{c}_{p,s}T = \alpha_s R_s T \quad (2.18)$$

where $\alpha_s = 7/2$ for molecular species and $\bar{c}_{p,s}$ indicates the constant pressure specific heat that is linked only to the translational and rotational modes of energy. Repeating the same procedure, one obtains $\alpha_s = 5/2$ for atomic species.

Finally, the internal energy per unit mass of the mixture can be evaluated by weighting the single species energies on the species mass fractions, thus $e = \sum_s Y_s e_s$. Now, recalling that $\rho h = \rho e + \rho RT = \rho e + p$ and considering that $\rho e = \rho E - \rho e_{kin}$ (e_{kin} being the kinetic energy), it holds:

$$\rho (h_{tr} + h_{rot} + e_{vib} + e_{chem}) = \rho E - \rho e_{kin} + p \quad (2.19)$$

After few algebraical operations, one obtains:

$$p = (\bar{\gamma} - 1) \left[\rho E - \rho (e_{vib} + e_{chem}) - \rho \frac{u^2 + v^2 + w^2}{2} \right] \quad (2.20)$$

that allows to compute the pressure [66,67]. Note that:

$$\bar{\gamma} = \frac{\bar{c}_p}{\bar{c}_v} = \frac{\alpha}{\alpha - 1} \quad (2.21)$$

is not the isentropic coefficient, but it is just the portion of such coefficient that accounts for translational and rotational energy modes. In such expression, α is weighted on all the species:

$$\alpha = \frac{\sum_s c_s \alpha_s}{\sum_s c_s} \quad (2.22)$$

where c_s is the molar concentration of the species s .

In Eq. (2.20), the chemical contribution is defined through the following expression:

$$e_{chem} = \frac{1}{\rho} \sum_{s=1}^{N_s} \rho_s h_s^f \quad (2.23)$$

where h_s^f is the specific formation enthalpy of species s (summarized in table 2.10). The evaluation of the vibrational contribution, instead, depends on the thermochemical model and is presented in section 2.4.

2.3 Transport properties

The evaluation of the transport properties is one of the most crucial issues when dealing with multicomponent gases, since it needs a strategy able to account for non-equilibrium. The discussion of such procedures is based on classical mixing rules [63, 68–70] and is here presented. Note that the following formulations are directly implemented in the source code, with no external library usage.

2.3.1 Viscosity

Each transport property is function of the single species properties. Hence, in order to evaluate the properties of the mixture, one has to start focusing on the single species. This subsection is dedicated to the viscosity. The calculation of the single species viscosity depends on the value of the temperature:

$$\begin{cases} \ln \mu_s = \sum_{n=1}^N A_{n,s} (\ln T)^{n-1} & \text{if } T \leq 1000 \text{ K} \\ \mu_s = [\exp(C_{\mu_s})] T^{(A_{\mu_s} \ln T + B_{\mu_s})} & \text{if } T > 1000 \text{ K} \end{cases} \quad (2.24)$$

The first expression is taken from Chemkin library [71] and $N = 4$ to ensure a third order polynomial expression. On the other hand, Gupta fitting [72] is employed for higher values of the temperature. The coefficients of both expressions are summarized in tables 2.1 and 2.2.

Table 2.1: Coefficients to compute the viscosity for $T \leq 1000$ K (obtained in kg/m s).

Species	$A_{1,s}$	$A_{2,s}$	$A_{3,s}$	$A_{4,s}$
N ₂	-16.26173	2.25174	-0.21383	0.009478
O ₂	-16.81080	2.52253	-0.24907	0.011006
NO	-16.22737	2.25174	-0.21383	0.009478
N	-14.66642	1.57776	-0.12529	0.005601
O	-14.81564	1.80140	-0.15491	0.006909

At this point, the values of the single species viscosity are combined to compute the mixture viscosity. This operation is performed through Wilke’s mixing rule [68]:

$$\mu = \sum_{s=1}^{N_s} \frac{\mathcal{X}_s \mu_s}{\sum_{r=1}^{N_s} \mathcal{X}_r \phi_{sr}} \quad (2.25)$$

Table 2.2: Coefficients to compute the viscosity for $T > 1000$ K (obtained in kg/m s).

Species	A_{μ_s}	B_{μ_s}	C_{μ_s}
N ₂	0.0203	0.4329	-11.8153
O ₂	0.0484	0.1455	-8.9231
NO	0.0452	-0.0609	-9.4596
N	0.012	0.593	-12.3805
O	0.0205	0.4257	-11.5803

where

$$\phi_{sr} = \frac{1}{\sqrt{8}} \left(1 + \frac{\mathcal{M}_s}{\mathcal{M}_r} \right)^{-1/2} \left[1 + \left(\frac{\mu_s}{\mu_r} \right)^{1/2} \left(\frac{\mathcal{M}_r}{\mathcal{M}_s} \right)^{1/4} \right]^2 \quad (2.26)$$

Such a formulation is known to be accurate for temperatures in the dissociation range (< 9000 K). More sophisticated formulations exist [64, 73, 74]. One of the most important and widely developed follows the Chapman-Enskog analysis [64]. This method allows for accurate calculations of transport properties starting from the Boltzmann equation. On the other hand, when dealing with multi-species mixtures, the use of this sophisticated algorithm can be very costly. Mixing rules provide a good approximation of the transport properties in the range of dissociation phenomena. Fig. 2.5 illustrates the profiles of the viscosity calculated through the Chapman-Enskog formulation and through Wilke's mixing rule for an 11 species air mixture in a temperature range $500 \div 20000$ K. The two curves are coherent with each other when predicting the drop of the viscosity after the ionization temperature threshold: electrons present a much smaller mass with respect to classical heavy species and they do not really contribute to the transport of momentum. Moreover, the much larger cross sections typical of ions-ions interactions promote a reduction of the viscosity, explaining its drop when ionization phenomena are induced. The under prediction of the viscosity computed through Wilke's mixing rule is due to the neglected cross interactions (ions-heavy species). Nevertheless, the mixing rule is in a good agreement for temperatures below the ionization threshold. The present work does not deal with ionized flow, making Wilke's mixing rule an affordable strategy to exploit GPUs capabilities and reduce the computational cost.

2.3.2 Thermal conductivity

Pure thermal conductivities are evaluated for each species as a function of the viscosity. This is known as Eucken's formula [63, 70, 75, 76]:

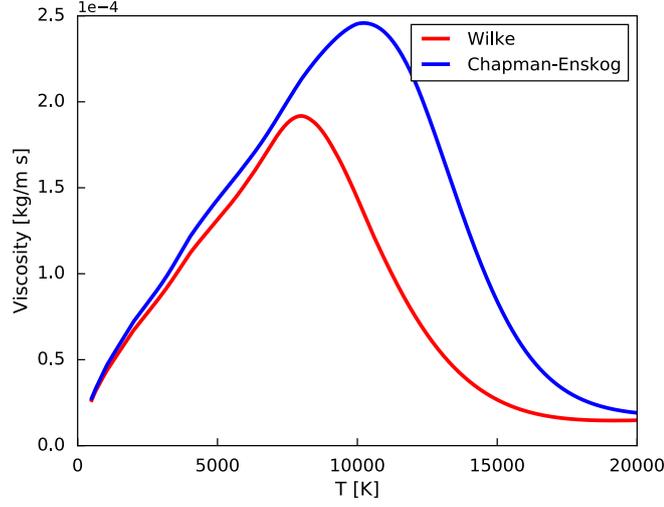


Figure 2.5: Comparison of viscosity computed through Chapman-Enskog algorithm and Wilke’s mixing rule ($p = 1$ atm).

$$\kappa_{tr,s} = \mu_s \left(\frac{5}{4} + \alpha_s \right) \frac{\mathfrak{R}}{\mathcal{M}_s} \quad (2.27)$$

whereas the vibrational thermal conductivities of the molecules are computed as follows [63, 75]:

$$\kappa_{vib,m} = \mu_m c_{vib} \quad (2.28)$$

where c_{vib} is the vibrational specific heat. Note that it is not needed to specify whether it is constant volume or constant pressure specific heat, since $h_{vib} = e_{vib}$ [9], as also derived in section 2.2.4. Eucken’s formulation is known to be accurate, even if a semi-empirical correction has been proposed by Hirschfelder [77] to improve even more the agreement with experiments. However, no assessment was made for air mixture in a high temperature range, including electronic excitation. Wilke’s mixture rule for viscosity is adapted for the conductivity, with an empirical correction [69]:

$$\kappa_{tr} = \sum_{s=1}^{N_s} \frac{\mathcal{X}_s \mu_s}{\sum_{r=1}^{N_s} 1.065 \mathcal{X}_r \phi_{sr}} \quad (2.29)$$

As for the viscosity, the Chapman-Enskog formulation can be applied to the thermal conductivity. Fig. 2.6 shows the thermal conductivity computed through the two methods. Again, the mixing rule under estimates the thermal conductivity. Nevertheless, preliminary simulation have shown no relevant differences in the final results.

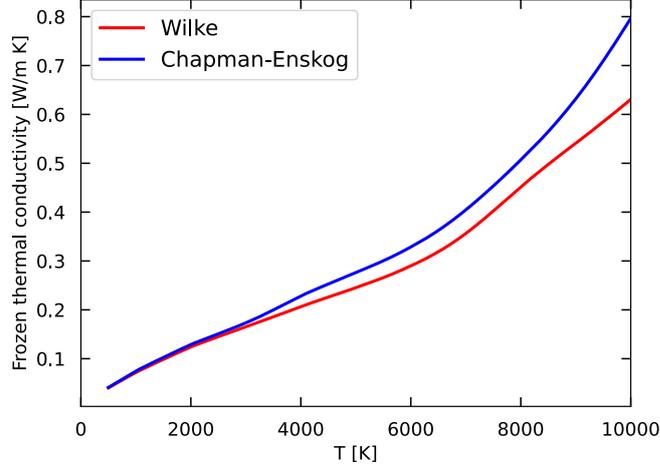


Figure 2.6: Comparison of frozen thermal conductivity computed through Chapman-Enskog algorithm and Wilke’s mixing rule ($p = 1$ atm).

2.3.3 Diffusivity

As anticipated in section 2.2.1, the diffusion coefficients should be evaluated by solving the system given in Eq. (2.4), but the Hirschfelder and Curtiss approximation allows to facilitate this calculation [61]. The diffusion coefficients are needed for the evaluation of the diffusion velocities, whose relation is here recalled (vibrational level subscript is here neglected for simplicity):

$$\rho_s \mathbf{u}_s^D = -\rho \mathcal{D}_s \nabla Y_s + \mathbf{V}_c \quad (2.30)$$

where $\mathbf{V}_c = \sum_s Y_s \mathcal{D}_s \nabla Y_s$ is a correction velocity to ensure mass conservation [62]. \mathcal{D}_s is an equivalent diffusion coefficient representing the diffusion of the species s into the entire mixture:

$$\mathcal{D}_s = \frac{1 - \mathcal{X}_s}{\sum_{r \neq s} \mathcal{X}_r / \mathcal{D}_{sr}} \quad (2.31)$$

so that only N_s diffusion coefficients must be computed. In this expression, \mathcal{D}_{sr} is the binary diffusion coefficient, that is obtained through the first order Chapman-Enskog approximation $\mathcal{D}_{sr} = \bar{\mathcal{D}}_{sr}/p$ where p is the pressure [61]. As for the viscosity, two range of temperatures are considered to compute $\bar{\mathcal{D}}_{sr}$:

$$\begin{cases} \ln \bar{\mathcal{D}}_{sr} = \sum_{n=1}^N D_n (\ln T)^{n-1} & \text{if } T \leq 1000 \text{ K} \\ \bar{\mathcal{D}}_{sr} = \exp(D_{\bar{\mathcal{D}}_{sr}}) T^{[A_{\bar{\mathcal{D}}_{sr}} (\ln T)^2 + B_{\bar{\mathcal{D}}_{sr}} (\ln T) + C_{\bar{\mathcal{D}}_{sr}}]} & \text{if } T > 1000 \text{ K} \end{cases} \quad (2.32)$$

where the first expression is taken from Chemkin library [71], with $N = 4$. The second expression represents the curve fitting by Gupta [72]. The coefficients for both formula are reported in tables 2.3 and 2.4.

Table 2.3: Coefficients to compute the diffusion coefficients for $T \leq 1000$ K (obtained in $\text{m}^2\text{Pa/s}$).

Species	D_1	D_2	D_3	D_4
N ₂ -N ₂	-15.2141	3.3481	-0.22337	0.0098178
N ₂ -O ₂	-15.4441	3.4349	-0.23400	0.0102503
N ₂ -NO	-15.2329	3.3488	-0.22347	0.0098226
N ₂ -N	-14.1963	3.0844	-0.19006	0.0084146
N ₂ -O	-14.4266	3.2164	-0.20696	0.0091353
O ₂ -O ₂	-15.7917	3.5721	-0.25185	0.0110253
O ₂ -NO	-15.4587	3.4335	-0.23380	0.0102416
O ₂ -N	-14.4686	3.1961	-0.20430	0.0090191
O ₂ -O	-14.7436	3.3501	-0.22454	0.0099065
NO-NO	-15.2485	3.3486	-0.22337	0.0098178
NO-N	-14.2314	3.0946	-0.19146	0.0084787
NO-O	-14.4686	3.2289	-0.20866	0.0092123
N-N	-12.9896	2.7092	-0.14047	0.0062300
N-O	-13.1483	2.8230	-0.15531	0.0068746
O-O	-13.2950	2.9390	-0.17060	0.0075474

2.4 Thermochemical models

This section is devoted to the thermochemical models. These are responsible for the modeling of the source terms in Eq. (2.1). In case of the mT model, Eq. (2.10) is also solved and the source terms modeled.

2.4.1 Multitemperature Park model

The multitemperature model proposed by Park [8] accounts for a 5 species neutral air mixture (N₂, O₂, NO, N, O) and considers the following dissociation reactions:



and two NO exchange (Zeldovich) reactions:



Table 2.4: Coefficients to compute the diffusion coefficients for $T > 1000$ K (obtained in $\text{m}^2\text{Pa/s}$).

Species	$A_{\overline{D}_{ST}}$	$B_{\overline{D}_{ST}}$	$C_{\overline{D}_{ST}}$	$D_{\overline{D}_{ST}}$
N ₂ -N ₂	0.0	0.0112	1.6182	-11.3091
N ₂ -O ₂	0.0	0.0465	0.9271	-8.11370
N ₂ -NO	0.0	0.0291	1.2676	-9.68780
N ₂ -N	0.0	0.0195	1.4880	-10.3654
N ₂ -O	0.0	0.0140	1.5824	-10.8819
O ₂ -O ₂	0.0	0.0410	1.0023	-8.35970
O ₂ -NO	0.0	0.0438	0.9647	-8.23800
O ₂ -N	0.0	0.0179	1.4848	-10.2810
O ₂ -O	0.0	0.0226	1.3700	-9.66310
NO-NO	0.0	0.0364	1.1176	-8.96950
NO-N	0.0	0.0185	1.4882	-10.3301
NO-O	0.0	0.0179	1.4848	-10.3155
N-N	0.0	0.0033	1.5572	-11.1616
N-O	0.0	-0.0048	1.9195	-11.9261
O-O	0.0	0.0034	1.5572	-11.1729

being X a generic partner in the mixture. Since each species can interact with any other one in the mixture, the total number of reactions is 17. For the evaluation of chemical source terms $\dot{\omega}_s$, the law of mass action is employed. It reads:

$$\dot{\omega}_s = \mathcal{M}_s \sum_{i=1}^{N_r} \nu_{is} RR_i \quad (2.35)$$

where N_r is the number of chemical reactions, \mathcal{M}_s is the molar weight of species s , ν_{is} is the difference between product and reactant stoichiometric coefficients of the species s in the reaction i . Rate coefficients of each reaction i , RR_i , are evaluated as:

$$RR_i = k_{f,i} \prod_{s=1}^{N_s} c_s^{\nu'_{is}} - k_{b,i} \prod_{s=1}^{N_s} c_s^{\nu''_{is}} \quad (2.36)$$

Here, ν'_{is} and ν''_{is} are reactant and product stoichiometric coefficients respectively, k_f and k_b are forward and backward rate coefficients and c_s is the concentration of species s . Forward coefficients are evaluated by means of Arrhenius formula [8, 78]:

$$k_{f,i} = A_i T_x^{m_i} \exp(-T_{d_i}/T_x) \quad (2.37)$$

where T_{d_i} is the characteristic activation temperature of reaction i , $T_x = T$ for NO exchange reactions, whereas for dissociation reactions a geometric average is

employed, thus $T_x = T_a = \sqrt{T_{vib}T}$ to take into account vibrational excitation in chemical activity. Note that T_{vib} refers to the vibrational temperature of the molecule considered for the dissociation. Parameters of such expression are given in table 2.5.

Table 2.5: Parameters to compute forward reaction rates [8, 10].

Reaction	X	T_x [K]	n_i	A_i [cm ³ /mol s]	T_d [K]
$N_2 + X \leftrightarrow 2N + X$	N ₂	T_a	-1.6	7.0^{21}	113200
	O ₂			7.0^{21}	
	NO			7.0^{21}	
	N			3.0^{22}	
	O			3.0^{22}	
$O_2 + X \leftrightarrow 2O + X$	N ₂	T_a	-1.5	2.0^{21}	59500
	O ₂			2.0^{21}	
	NO			2.0^{21}	
	N			1.0^{22}	
	O			1.0^{22}	
$NO + X \leftrightarrow N + O + X$	N ₂	T_a	0.0	5.0^{15}	75500
	O ₂			5.0^{15}	
	NO			1.1^{17}	
	N			1.1^{17}	
	O			1.1^{17}	
$NO + O \leftrightarrow N + O_2$	-	T	0.0	8.4^{12}	19450
$N_2 + O \leftrightarrow NO + N$	-	T	-1.0	6.4^{17}	38400

The following expression is used to evaluate the equilibrium constant ($Z = 10^4/T_x$):

$$\ln K_{eq,i} = \frac{A_1}{Z} + A_2 + A_3 \ln Z + A_4 Z + A_5 Z^2 \quad (2.38)$$

whose coefficients are given in table 2.6. Finally, backward rate coefficients are computed as $k_{b,i} = k_{f,i}/K_{eq,i}$, so that the reaction rates (and so the source terms) can be evaluated to solve the species continuity equations.

Table 2.6: Parameters to compute the equilibrium constant [8].

Reaction	A_1	A_2	A_3	A_4	A_5
$N_2 + X \leftrightarrow 2N + X$	1.4766	1.6291	1.2153	-11.457	-0.009444
$O_2 + X \leftrightarrow 2O + X$	0.50989	2.4773	1.7132	-6.5441	0.029591
$NO + X \leftrightarrow N + O + X$	0.50765	0.73575	0.48042	-7.4979	-0.016247
$NO + O \leftrightarrow N + O_2$	-0.002428	-1.7415	-1.2331	-0.95365	-0.04585
$N_2 + O \leftrightarrow NO + N$	0.96921	0.89329	0.73531	-3.9596	0.006818

In order to solve the vibrational energy conservation given in Eq. (2.10), two contributions are considered to model the vibrational source terms of the molecules, $\dot{\omega}_{vib,m}$. Specifically, it is decomposed in two contributions: the collisional ($\dot{\omega}_{LT,m}$) and chemical ($\dot{\omega}_{chem,m}$) one.

The collisional contribution $\dot{\omega}_{LT,m}$ represents the energy transfer between translational and vibrational modes, and is modeled by the Landau-Teller equation:

$$\dot{\omega}_{LT,m} = \rho_m \frac{e_{vib,m}(T) - e_{vib,m}(T_{vib})}{\tau_m} \quad (2.39)$$

This term is equal to zero in the case of thermal equilibrium ($T = T_{vib}$). Here, τ_m is the relaxation time, namely the time required to reach the thermal equilibrium condition. As first approximation, it is evaluated through the Millikan-White expression [8, 10], that reads:

$$\tau_{m,X}^{MW} = \frac{p_{atm}}{p} \exp [A_{m,X} (T^{-1/3} - B_{m,X}) - 18.42] \quad (2.40)$$

It accounts for the collision of the molecule m with a generic partner X. The coefficients $A_{m,X}$ and $B_{m,X}$ can be evaluated following simple expressions given in [79]: specific values of this work are reported in table 2.7.

Table 2.7: Parameters to compute the relaxation time [10].

Molecule	X	$A_{m,X}$	$B_{m,X}$
N ₂	N ₂	221.0	0.029
	O ₂	229.0	0.0295
	NO	225.0	0.0293
	N	180.0	0.0262
	O	72.4	0.015
O ₂	N ₂	134.0	0.0295
	O ₂	138.0	0.03
	NO	136.0	0.0298
	N	72.4	0.015
	O	47.7	0.059
NO	N ₂	49.5	0.042
	O ₂		
	NO		
	N		
	O		

Nevertheless, for temperatures above 5000 K, the prediction of the relaxation time obtained by using the Millikan-White expression does not well reproduce experimental results [8, 10, 12], since it does not account for the limits in collision cross sections. To overcome this drawback, Park proposed a correction in the evaluation of τ_m [8], taking into account the effective excitation cross section:

$$\tau_{m,X}^c = \frac{1}{n_m \sigma \sqrt{\frac{8RT}{\pi \mathcal{M}_{m,X}}}} \quad (2.41)$$

n_m being the number density of molecule m , σ the effective excitation cross section equal to $3 \times 10^{-17} (50000/T^2)$ cm² [10] and $\overline{\mathcal{M}}_{m,X} = \mathcal{M}_m \mathcal{M}_X / (\mathcal{M}_m + \mathcal{M}_X)$ the equivalent molecular weight of two colliding particles.

Hence, the relaxation time for the collision between a molecule and a generic partner X is given by $\tau_{m,X} = \tau_{m,X}^{MW} + \tau_{m,X}^c$ and its mean value is evaluated through a weighted harmonic average:

$$\frac{1}{\tau_m} = \frac{1}{n_t} \sum_{s=1}^{N_s} \frac{n_s}{\tau_{m,X}} \quad (2.42)$$

Here n_t is the total number density, while n_s is the number density of the species s .

The second contribution of the vibrational source terms is $\dot{\omega}_{chem,m}$, that takes into account the influence of the dissociation on the vibrational energy. In this work, the rigid-rotor-harmonic oscillator model (RRHO) is employed, thus the energy exchanged in dissociation process is equally divided into vibrational and translational degrees of freedom [8]:

$$\dot{\omega}_{chem,m} = \frac{D_m}{2} \dot{\omega}_m \quad (2.43)$$

where D_m is the dissociation energy per unit mass of molecule m . Finally, vibrational temperatures are evaluated by inverting the following expression for the molecular vibrational energy:

$$e_{vib,m} = \frac{R_m \theta_{v,m}}{\exp(\theta_{v,m}/T_{vib}) - 1} \quad (2.44)$$

where R_m is the gas constant and $\theta_{v,m}$ is the characteristic vibrational temperature of molecule m . Table 2.10 summarizes the characteristics of the species useful for such computations.

2.4.2 State-to-State model

The mT model is based on the harmonic oscillator assumption to describe the vibrational mode. Specifically, the population of the vibrational states follows a Boltzmann distribution. Such an assumption is not suitable in many cases [21, 22, 80]. Hence, a more detailed description of the vibrational internal states is needed. The State-to-State (StS) model [15] represents a very powerful tool to deal with strong non-equilibrium conditions, known to be relevant across strong shock waves [81, 82] and in the boundary layer of a entry vehicle [83–85]. As a consequence, the global rates of chemical mechanisms are strongly affected [17, 83, 86], especially in the presence of atom recombination [87, 88].

In this work, the StS model is focused on a 5 species neutral air mixture (N_2 , O_2 , NO , N , O), but only N_2 and O_2 are treated through a StS kinetics. The vibrational energy of each level v is described by an anharmonic polynomial expansion [15]:

$$e_{vib}(v) = hc \left(\omega_e \bar{v} + \omega_e x_e \bar{v}^2 + \omega_e y_e \bar{v}^3 + \omega_e z_e \bar{v}^4 \right)_{|\bar{v}=v+1/2} \quad (2.45)$$

being h the Planck constant and c the speed of light. The spectroscopic constants are summarized in table 2.8.

Table 2.8: Spectroscopic constants expressed in m^{-1} for N_2 [88–90] and O_2 [90].

Constant [m^{-1}]	N_2	O_2
ω_e	237244.63054	158058.2
$\omega_e x_e$	-1810.16777	1020.365
$\omega_e y_e$	1.2755162435	-5.223211
$\omega_e z_e$	-0.0079594872485	-0.08941839

The first order approximation of Eq. (2.45) provides the harmonic oscillator model. On the other hand, the StS model describes the vibrational mode through a finite number of levels v_{max} , obtained from the condition $e_{vib}(v_{max}) < D_m < e_{vib}(v_{max} + 1)$, where D_m is the dissociation energy of the molecule m . Such an approach leads to 68 vibrational levels for N_2 and 47 vibrational levels for O_2 , whereas only the fundamental state is considered for N , O and NO .

The main difference from the mT model is in the vibrational temperature. Indeed, in the mT model translational and rotational modes are considered at equilibrium, whereas the vibrational mode follows a Boltzmann distribution at the vibrational temperature (T_{vib}). The role of such a parameter is to affect the reaction rates. Indeed, as previously mentioned, a geometrical mean between roto-translational and vibrational temperature is used to calculate the reaction rates, so that chemical activity is accelerated in the presence of vibrational excitation. This is suitable in the dissociation range, but is not that effective in the recombination range.

In the StS formulation, the energy exchange could occur through different elementary processes, classified as:

- Vibration-Vibration (VV): $\text{A}_2(v) + \text{B}_2(w) \leftrightarrow \text{A}_2(v - 1) + \text{B}_2(w + 1)$
- Vibration-Translation by molecules (VTm): $\text{A}_2(v) + \text{B}_2 \leftrightarrow \text{A}_2(v - 1) + \text{B}_2$
- Vibration-Translation by atoms (VTa): $\text{A}_2(v) + \text{B} \leftrightarrow \text{A}_2(w) + \text{B}$
- Dissociation-Recombination by molecules (DRm): $\text{A}_2(v) + \text{B}_2 \leftrightarrow 2\text{A} + \text{B}_2$
- Dissociation-Recombination by atoms (DRa): $\text{A}_2(v) + \text{B} \leftrightarrow 2\text{A} + \text{B}$

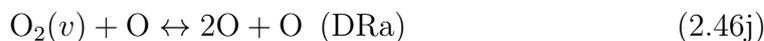
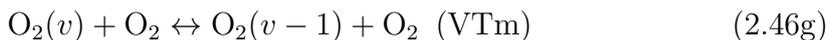
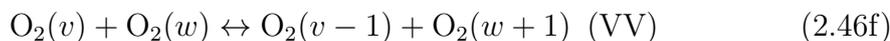
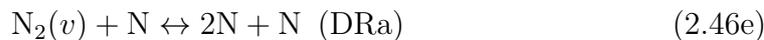
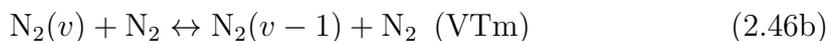
being v and w generic vibrational levels. VV mechanisms express energy exchange just through interactions between two energy levels, whereas VT mechanisms are based on cross interactions (vibrational mode/translational mode). These interactions can lead to a single-quantum transition, as in the case of VTm, or to multi-quantum transition, more probable in the case of interaction with atoms (VTa).

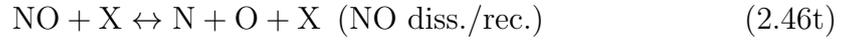
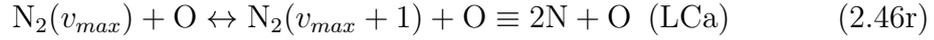
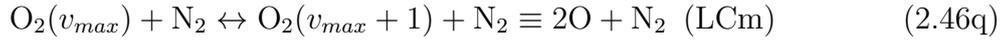
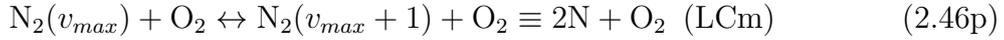
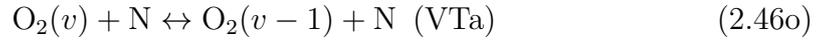
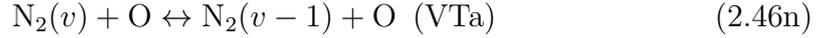
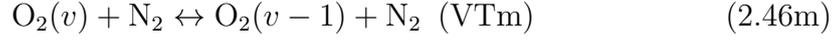
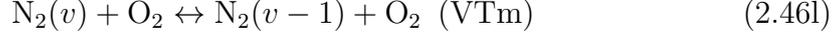
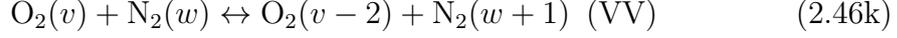
Concerning the DR mechanisms, it is worth stressing the point that rates for each vibrational level must be known. When no data are available, the *Ladder Climbing* model (LC) can be applied [15]. In such an approach, a pseudo level above the last one is considered to represent the dissociation. Hence, DRm and DRa processes are adapted as follows:

- LCm: $A_2(v_{max}) + B_2 \rightarrow A_2(v_{max} + 1) + B_2 \equiv 2A + B_2$
- LCa: $A_2(v_{max}) + B \rightarrow A_2(v_{max} + 1) + B \equiv 2A + B$

It is known that the Ladder Climbing approach can introduce large errors in the evaluation of the reaction rates [91], but this is an affordable strategy to account for these kind of mechanisms when no rate can be evaluated.

Hence, the total set of reactions involved is:





Few comments about this set of reactions are mandatory. First of all, one can notice that VTm interactions do not include NO. These interactions are bypassed in order to save computational time, as the mass fraction of NO is often very small in most of hypersonic flow applications. Anyway, its dissociation and exchange reactions are involved (last three mechanisms) [14]. Furthermore, mixed VV mechanism between O_2 and N_2 presents a multi-quantum transition for the oxygen. This is due to the energy distribution among the levels: the energy needed for a mono-quantum transition in N_2 is approximately doubled-up in O_2 . Lastly, the LC approach is employed only in case of mixed dissociation/recombination processes, as dissociation rates for homo-nuclear interactions are computed [92].

Contrary to the mT model, the vibrational temperature has no physical meaning, as the vibrational modes do not follow a Boltzmann distribution. In the mT model it is used to account for vibrational excitation on the chemical activity. In the case of the StS model, the source terms are just function of the

roto-translational temperature [15, 88, 92]. Nevertheless, it can be useful to define a vibrational temperature to give an indication of the vibrational energy content. Since the first levels (≈ 10) approximately follow a Boltzmann distribution, it can be defined from the first 2 levels:

$$T_{vib,m} = \frac{e_{vib,m,2} - e_{vib,m,1}}{R_m \ln \left(\frac{\rho_{m,2}}{\rho_{m,1}} \right)} \quad (2.47)$$

being R_m the gas constant of molecule m . Knowing the vibrational energy of each level of each molecule, one can evaluate the total contribution as:

$$e_{vib} = \frac{1}{\rho} \sum_{s=1}^{N_s} \sum_{l=1}^{V_s} e_{vib,m,l} \quad (2.48)$$

For the sake of brevity, expressions for reaction rates are not reported in this thesis as it would require a dedicated chapter. In the following, references for reactions rates are given for the mixture considered in this work. For simplicity, homo-nuclear and hetero-nuclear mechanisms are split: the first ones involve interactions between species of the same nature (N_2/N or O_2/O); the second ones involve mixed interactions (N_2/O_2 , N_2/O , $O_2/N...$).

Homo-nuclear N_2 mechanisms

For VV and VTm mechanisms of N_2 (processes 2.46a and 2.46b), a curve fitting from Billing semi-classical collision model calculations [93] is used [94]. VTa and DRa rates (processes 2.46c and 2.46e) are obtained by fitting quasi-classical trajectory (QCT) calculations performed by Esposito et al. [92, 95]. Concerning DRm rates (process 2.46d), no detailed StS calculations exist. Hence, a scaling of the DRa rates is used [92], which implicitly assumes that the state-to-state rates behave like the global rates. Such an approach follows the formulation by Shatalov [96], which makes the rates depend on the temperature.

Homo-nuclear O_2 mechanisms

For VV and VTm mechanisms of O_2 (processes 2.46f and 2.46g), a curve fitting from Billing semi-classical collision model calculations [97] is used [94]. Rates for VTa and DRa mechanisms (processes 2.46h and 2.46j) are fitted following QCT calculations by Esposito et al. [98]. Similarly to N_2 , DRm rates for O_2 (process 2.46i) are obtained by scaling DRa.

Hetero-nuclear and NO mechanisms

Hetero-nuclear interactions involve state-to-state N_2/O_2 mechanisms and NO mechanisms. Specifically, mixed nitrogen and oxygen processes (from 2.46k to 2.46s)

are taken from [99]. Concerning NO dissociation (process 2.46t) and exchange reactions (processes 2.46u and 2.46v), the rates follow the formulation given in [13, 99, 100].

2.5 Gas-Surface Interaction (GSI)

Thermal protection systems (TPS) of space vehicles are designed for the heat mitigation during re-entry flights. Such a mitigation exploits specific materials whose decomposition allows to reduce the heat load on the thermal shield. For this reason, source terms arising from such gas-surface interaction (GSI) mechanisms must be taken into account. A generic description of the GSI mechanisms is sketched in Fig. 2.7, where dashed lines identify an infinitesimal section between gas and solid phases. Because of the extreme thermal and chemical conditions undergone by the body entering in the atmosphere, internal material consumption occurs, making the solid porous. Hence, pyrolysis gases percolate toward the surface, together with gas product due to solid-solid reactions promoted by several kind of materials (such as carbon and silica in silica-phenolic composites). Moreover, part of solid residual (char) can reach the surface, where it can sublime or react with the species in the boundary layer. These processes provoke a substantial modification of the bulk mass next to the surface because of a mass addition. The flow gases are then convected away from the surface (*blowing* effect). Furthermore, mechanical removal might occur because of extreme shear stresses or particle impingement.

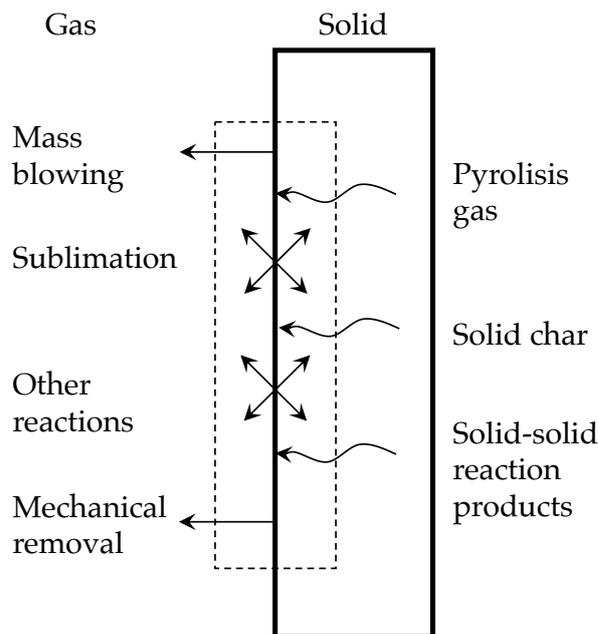


Figure 2.7: Sketch of a general GSI mechanisms.

Such phenomena are described by a simple mathematical formulation. Assuming a steady state and considering the balance for the infinitesimal volume between the gas and the solid phase, the total flux contribution of the generic conservative variable is expressed as:

$$(\mathbf{F}_{gas} - \mathbf{F}_{sol}) \cdot \mathbf{n} = \mathbf{W}_{surf} \quad (2.49)$$

being \mathbf{F}_{gas} and \mathbf{F}_{sol} the fluxes in the gas and in the solid (body) phase, considering a positive surface normal vector \mathbf{n} coming out from the solid; finally, \mathbf{W}_{surf} is the sum of all the source terms arising from the GSI mechanisms. Such a formulation allows to re-adapt the mass balance, by opportunely substituting the fluxes expressions.

2.5.1 Mass balance

Starting from the species continuity law, Eq. (2.49) can be written for each species. In this thesis, mechanical degradation is linked to ablation and catalysis only, thus $\dot{\omega}_{surf,s} = \dot{\omega}_{abla,s} + \dot{\omega}_{cat,s}$. Therefore, following Fig. 2.8, for each species s the mass balance reads:

$$\underbrace{\rho_s (u_g - u_r) + j_s}_{F_{gas,s}} = \underbrace{\dot{\omega}_{abla,s} + \dot{\omega}_{cat,s}}_{\dot{\omega}_{surf,s}} \quad (2.50)$$

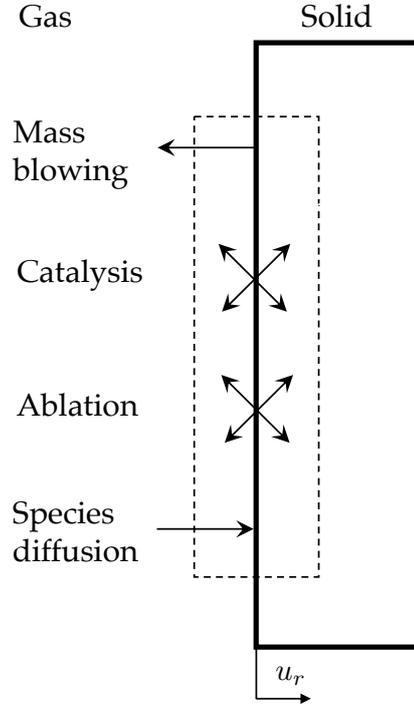


Figure 2.8: Overall mass fluxes.

since $F_{sol,s}$ is null. In Eq. (2.50), u_g is the gas velocity coming out from the wall, whereas u_r is the recession velocity, namely the velocity the solid is moving with. Since no mechanical erosion is considered, it is reasonable to assume $u_g \gg u_r$, so that u_r can be neglected. Therefore, the mass balance reads:

$$\boxed{\rho_s u_g = \dot{\omega}_{abla,s} + \dot{\omega}_{cat,s}} \quad (2.51)$$

By summing Eq. (2.51) over all the species, the catalytic source term disappears since $\sum_s \dot{\omega}_{cat,s} = 0$ and the expression of the gas velocity coming from the surface, u_g , is obtained::

$$u_g = \frac{\dot{m}_{blow}}{\rho_{wall}} \quad (2.52)$$

being ρ_{wall} the mixture density on the wall and $\dot{m}_{blow} = \sum_s \dot{\omega}_{abla,s}$ the massblowing rate.

2.5.2 Modeling of catalysis

A wall is classified as *catalytic* when it promotes chemical reactions. Catalysis can be modeled in different ways [101], but the most common one is the γ model [102], widely used in many current applications due to its simplicity [103, 104]. The formulation in the following is presented in the framework of macroscopic models (multitemperature), being the StS approach a specific case [84, 86, 105]. A generic atomic species X recombines on the surface in the molecule X_2 :



For such a mechanism, a recombination probability must be defined. It is interpreted as the ratio between the actually recombined particles mass flux and the impinging particles mass flux:

$$\gamma = \frac{\mathcal{F}_{rec}}{\mathcal{F}_{imp}}$$

This coefficient γ can be defined for each separate species, even if a unique value is often considered for all the species. One can distinguish a *non-catalytic* wall ($\gamma = 0$), completely inert to kinetics. On the other extreme, a *fully-catalytic* wall ($\gamma = 1$) promotes a complete recombination. All the values of γ in the middle identify a *partially-catalytic* wall ($0 < \gamma < 1$). This is an input parameter of the model, needed to evaluate the source term in Eq. (2.51):

$$\dot{\omega}_{cat,a} = \gamma m_a \mathcal{F}_{imp,a} \quad (2.53)$$

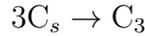
the subscript a referring to an atomic species with particle mass m_a . A similar source term is defined for the corresponding molecule m , so that $\dot{\omega}_{cat,m} = -\dot{\omega}_{cat,a}$. In the presence of an isothermal wall and an approximately Maxwellian internal states distribution, the impinging mass flux of a generic species s is given by [106]:

$$\mathcal{F}_{imp,s} = n_s \sqrt{\frac{k_B T_w}{2\pi m_s}} \quad (2.54)$$

where k_B is the Boltzmann constant and T_w is the wall temperature.

2.5.3 Modeling of ablation

Ablation can occur because of three different kind of reactions [37]:



representing respectively phenomena of nitridation, oxidation and sublimation. Note that, in these reactions, subscript s stands for 'surface', since an ablative material releases carbon from the surface in the flow field. The production term for CN, CO and C_3 follows a formulation similar to Eq. (2.53). To distinguish ablation products from catalysis, coefficient γ is substituted by coefficient β .

Several analysis have been carried out to provide a proper expression of the ablation probability coefficient. For the nitridation reaction, probability function by Helber et al. [107] is used:

$$\beta_{CN} = 0.0791 \exp(-5653/T_w) \quad (2.56)$$

that turned out to be effective for a wide range of wall temperatures. Concerning the oxidation reaction, Park [108] proposed a probability coefficient based on an Arrhenius type function, given by:

$$\beta_{CO} = 0.63 \exp(-1160/T_w) \quad (2.57)$$

Lastly, the only sublimation process accounted for is the production of C_3 , neglecting other pure carbon species due to their smaller production. The production term is given by a formulation proposed by Park [4]:

$$\dot{\omega}_{subl,C_3} = \beta_{C_3} (\rho_{C_3,eq} - \rho_{C_3}) \sqrt{\frac{k_B T_w}{2\pi m_{C_3}}} \quad (2.58)$$

being $\rho_{C_3,eq}$ the equilibrium partial density of C_3 obtained from the saturated vapor pressure of carbon (in Pa), given by [4]:

$$p_{sat,C_3} = 5.19 \times 10^{14} \exp(-90845/T_w) \quad (2.59)$$

For this kind of reaction, it has been found that coefficient β_{C_3} can vary from 0.1 to 1, with no relevant differences in the results. In this thesis $\beta_{C_3} = 1$ has been considered. Such a formulation finds application in many CFD problems carried out in different researches [36, 38, 109–111]. As mentioned previously, the injection of carbon species in the flow field leads to chemical reactions with heavy species in the boundary layer. For this reason, the set of chemical reactions given in table 2.5 must be extended. The 11 species mixture contains N_2 , O_2 , NO , N , O , C , C_2 , C_3 , CN , CO , CO_2 , and the complete set of reactions is summarized in table 2.9, along with the coefficients to compute forward rates through Eq. (2.37). Finally, table 2.10 summarizes the characteristics of the species, including molar weight, formation enthalpy and characteristic vibrational temperature. Note that, for C_3 and CO_2 , 4 different characteristic vibrational temperatures are specified. In general, the number of vibrational modes depends on the number of atoms in the molecule. Indicating with N_A the number of atoms in a molecule, the vibrational modes are $3N_A - 3 - \mathcal{L}$, where $3N_A - 3$ represents the degrees of freedom and \mathcal{L} represents the rotational modes (2 for linear molecules). Since in the considered mixture all the molecules are linear, $\mathcal{L} = 2$ and the only species presenting more than one vibrational modes are C_3 and CO_2 . For the sake of brevity, coefficients to compute the equilibrium constant with Eq. (2.38) and vibrational relaxation time with Eq. (2.40) are not reported here, but the reader can refer to the literature [11, 111].

Table 2.9: Parameters to compute forward reaction rates in ablative cases [36]. 'Mol' and 'At' stand for molecules and atoms respectively.

Reaction	X	T_x [K]	n_i	A_i [cm ³ /mol s]	T_d [K]
$N_2 + X \leftrightarrow 2N + X$	Mol	T_a	-1.6	7.0^{21}	113200
	At	T_a	-1.6	3.0^{22}	113200
$O_2 + X \leftrightarrow 2O + X$	Mol	T_a	-1.5	2.0^{21}	59750
	At	T_a	-1.5	1.0^{22}	59750
$NO + X \leftrightarrow N + O + X$	Mol	T_a	0	5.0^{15}	75500
	At	T_a	0	1.0^{17}	75500
$CO_2 + X \leftrightarrow CO + O + X$	Mol	T_a	-1.5	6.9^{21}	63275
	At	T_a	-1.5	1.4^{22}	63275
$CO + X \leftrightarrow C + O + X$	Mol	T_a	-1.0	2.3^{20}	129000
	At	T_a	-1.0	3.4^{20}	129000
$C_2 + X \leftrightarrow 2C + X$	Mol	T_a	0.0	3.7^{14}	69000
	At	T_a	0.0	3.7^{14}	69000
$C_3 + X \leftrightarrow C_2 + C + X$	Mol	T_a	-0.5	6.3^{16}	101200
	At	T_a	-0.5	6.3^{16}	101200
$CN + X \leftrightarrow C + N + X$	Mol	T_a	0.0	2.5^{14}	71000
	At	T_a	0.0	2.5^{14}	71000
$NO + O \leftrightarrow N + O_2$	-	T	0.0	8.4^{12}	19450
$N_2 + O \leftrightarrow NO + N$	-	T	-1.0	6.4^{17}	38370
$CO + O \leftrightarrow O_2 + C$	-	T	-0.18	3.9^{13}	69200
$CO_2 + O \leftrightarrow O_2 + CO$	-	T	0.0	2.1^{13}	27800
$CO + C \leftrightarrow C_2 + O$	-	T	-1.0	2.0^{17}	58000
$CO + N \leftrightarrow CN + O$	-	T	0.0	1.0^{14}	38600
$N_2 + C \leftrightarrow CN + N$	-	T	-0.11	1.1^{14}	23200
$CN + O \leftrightarrow NO + C$	-	T	0.10	1.6^{13}	14600
$CN + C \leftrightarrow C_2 + N$	-	T	0.0	5.0^{13}	13000

Table 2.10: Characteristics of the species ($T_{ref} = 298$ K, $p = 1$ atm) [112].

Species	\mathcal{M} [kg/mol]	h^f [J/mol]	θ_v [K]
N ₂	0.028014	0.0	3393
O ₂	0.031998	0.0	2273
NO	0.030006	91089	2739
N	0.014007	472440	-
O	0.015999	249229	-
C	0.012011	716680	-
C ₂	0.024022	828374	2604
C ₃	0.036033	823630	213, 213, 1723, 2990
CN	0.026018	439970	2993
CO	0.02801	-110530	3083
CO ₂	0.044009	-393472	932, 932, 1914, 3373

Chapter 3

Methodology and numerical formulation

The solver developed in the framework of this PhD project is called CHES (Cuda HypErSonic Solver). It is a C based code, extended to CUDA libraries to exploit GPU capabilities. The executable is standalone with no need of external libraries for thermodynamics and physics. Nevertheless, in its recent version it has been coupled with MUTATION⁺⁺ [112] to benefit of advanced algorithm for transport properties and gas-surface interaction mechanisms.

3.1 Finite volume approach

The numerical solver is finite-volume based. Such an approach consists in discretizing the computational domain in a finite number of control volumes, whose shape is arbitrary. Classical approaches exploit hexahedral cells that allow an easy computation of the volume and normal surface vectors. The integral form of the system of equations to solve can be written as follows:

$$\int_{\mathcal{V}_0} \frac{\partial}{\partial t} \mathbf{U} d\mathcal{V} + \oint_{S_0} \mathbf{F} \cdot \mathbf{n} dS = \int_{\mathcal{V}_0} \mathbf{W} d\mathcal{V} \quad (3.1)$$

where \mathbf{U} and \mathbf{W} are the conservative variables and source terms vectors, whereas \mathbf{F} is the fluxes tensor, respectively. They read:

$$\mathbf{U} = \underbrace{\begin{bmatrix} \rho_1 \\ \vdots \\ \rho_{N_s} \\ \rho u \\ \rho v \\ \rho w \\ \rho E \\ \rho_m e_{vib,m} \\ \vdots \\ \rho_{N_m} e_{vib,N_m} \end{bmatrix}}_{mT \text{ model}} \quad \mathbf{U} = \underbrace{\begin{bmatrix} \rho_{1,1} \\ \vdots \\ \rho_{1,V_1} \\ \vdots \\ \rho_{N_s,V_{N_s}} \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix}}_{StS \text{ model}} \quad (3.2a)$$

$$\mathbf{F} = (\mathbf{F}_I^x - \mathbf{F}_V^x, \mathbf{F}_I^y - \mathbf{F}_V^y, \mathbf{F}_I^z - \mathbf{F}_V^z) \quad (3.2b)$$

$$\underbrace{\begin{bmatrix} \rho_1 u \\ \vdots \\ \rho_{N_s} u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ (\rho E + p) u \\ \rho_m e_{vib,m} u \\ \vdots \\ \rho_{N_m} e_{vib,N_m} u \end{bmatrix} \quad \begin{bmatrix} \rho_1 v \\ \vdots \\ \rho_{N_s} v \\ \rho u v \\ \rho v^2 + p \\ \rho v w \\ (\rho E + p) v \\ \rho_m e_{vib,m} v \\ \vdots \\ \rho_{N_m} e_{vib,N_m} v \end{bmatrix} \quad \begin{bmatrix} \rho_1 w \\ \vdots \\ \rho_{N_s} w \\ \rho u w \\ \rho v w \\ \rho w^2 + p \\ (\rho E + p) w \\ \rho_m e_{vib,m} w \\ \vdots \\ \rho_{N_m} e_{vib,N_m} w \end{bmatrix}}_{mT \text{ model}} \quad (3.2c)$$

$$\underbrace{\begin{bmatrix} \rho_{1,1} u \\ \vdots \\ \rho_{1,V_1} u \\ \vdots \\ \rho_{N_s,V_{N_s}} u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ (\rho E + p) u \end{bmatrix} \quad \begin{bmatrix} \rho_{1,1} v \\ \vdots \\ \rho_{1,V_1} v \\ \vdots \\ \rho_{N_s,V_{N_s}} v \\ \rho u v \\ \rho v^2 + p \\ \rho v w \\ (\rho E + p) v \end{bmatrix} \quad \begin{bmatrix} \rho_{1,1} w \\ \vdots \\ \rho_{1,V_1} w \\ \vdots \\ \rho_{N_s,V_{N_s}} w \\ \rho u w \\ \rho v w \\ \rho w^2 + p \\ (\rho E + p) w \end{bmatrix}}_{StS \text{ model}} \quad (3.2d)$$

$$(\mathbf{F}_V^x, \mathbf{F}_V^y, \mathbf{F}_V^z) = \underbrace{\begin{bmatrix} -\rho_1 \mathbf{u}_1^D \\ \vdots \\ -\rho_{N_s} \mathbf{u}_{N_s}^D \\ \overline{\overline{\sigma}} \\ \mathbf{u} \cdot \overline{\overline{\sigma}} - \mathbf{q} \\ -\mathbf{q}_{vib,m} \\ \vdots \\ -\mathbf{q}_{vib,N_m} \end{bmatrix}}_{mT \text{ model}} \quad \underbrace{(\mathbf{F}_V^x, \mathbf{F}_V^y, \mathbf{F}_V^z) = \begin{bmatrix} -\rho_{1,1} \mathbf{u}_{1,1}^D \\ \vdots \\ -\rho_{1,V_1} \mathbf{u}_{1,V_1}^D \\ \vdots \\ -\rho_{N_s,V_{N_s}} \mathbf{u}_{N_s,V_{N_s}}^D \\ \overline{\overline{\sigma}} \\ \mathbf{u} \cdot \overline{\overline{\sigma}} - \mathbf{q} \end{bmatrix}}_{StS \text{ model}} \quad (3.2e)$$

$$\mathbf{W} = \underbrace{\begin{bmatrix} \dot{\omega}_1 \\ \vdots \\ \dot{\omega}_{N_s} \\ 0 \\ 0 \\ 0 \\ 0 \\ \dot{\omega}_{vib,m} \\ \vdots \\ \dot{\omega}_{vib,N_m} \end{bmatrix}}_{mT \text{ model}} \quad \underbrace{\mathbf{W} = \begin{bmatrix} \dot{\omega}_{1,1} \\ \vdots \\ \dot{\omega}_{1,V_{N_s}} \\ \vdots \\ \dot{\omega}_{N_s,V_{N_s}} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{StS \text{ model}} \quad (3.2f)$$

Here, two comments are mandatory. First of all, a distinction between the mT and the StS model is needed. Indeed, the number of species continuity equations is different in the two approaches (note that for the mT model the subscript for the vibrational levels is omitted). Secondly, N_m transport equations are solved for the vibrational energy in case of the mT model.

In this work, multi-block structured meshes are employed, each cell within a block being uniquely identified by three integers (i, j, k) . Hence, the discretized form of Eq. (3.1) is:

$$\mathcal{V}_{i,j,k} \frac{d\mathbf{U}_{i,j,k}}{dt} + \sum_{f=1}^{N_f} \mathbf{F}_f \cdot \mathbf{n}_f S_f = \mathcal{V}_{i,j,k} \mathbf{W}_{i,j,k} \quad (3.3)$$

where $\mathcal{V}_{i,j,k}$ is the volume of the cell, N_f is the number of faces (4 in 2D and 6 in 3D), whereas \mathbf{n}_f and S_f are the outward unit normal vector and the surface of face f . The evaluation of the conservative variables stored in \mathbf{U} is performed at the center of each cell, whereas the fluxes \mathbf{F} are evaluated on cell's surfaces, as shown in Fig. 3.1. Surface normal vectors can be computed following a general formulation based on the cross product between the diagonals of the face [113]. The example of a generic hexahedral volume is given in Fig. 3.2. Considering the face identified by vertices A, D, E and H, the surface normal vector is given by:

$$\mathbf{S}_{\text{AEHD}} = \frac{1}{2} (\mathbf{r}_{\text{ED}} \times \mathbf{r}_{\text{HA}}) \quad (3.4)$$

whose magnitude gives the area of the surface, S_f . The Cartesian components of the unit normal vector, \mathbf{n}_f , are obtained by normalizing \mathbf{S}_{AEHD} with respect to S_f . By performing the same procedure on surfaces ABFE and EFGH, one can compute \mathbf{S}_{ABFE} and \mathbf{S}_{EFGH} . Finally, the volume of the cell is computed by subdividing the hexahedron into three pyramids that share the same diagonal EC as common edge, thus:

$$\mathcal{V} = \frac{1}{3} (\mathbf{S}_{\text{AEHD}} + \mathbf{S}_{\text{ABFE}} + \mathbf{S}_{\text{EFGH}}) \cdot \mathbf{r}_{\text{EC}} \quad (3.5)$$

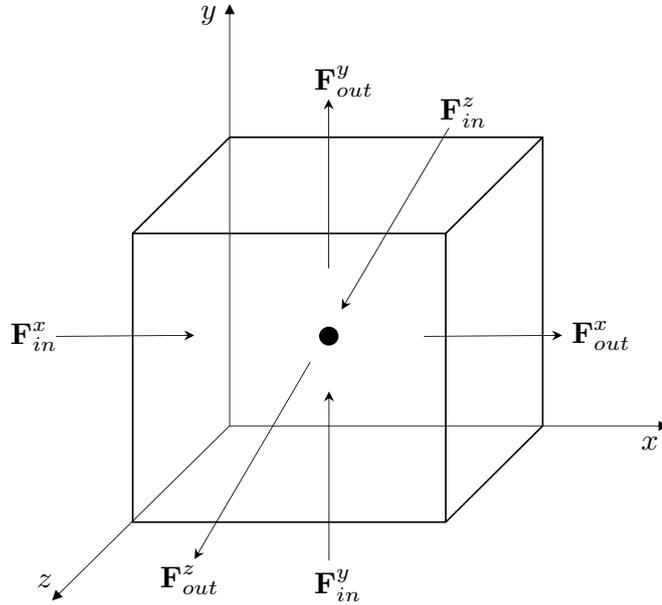


Figure 3.1: Numerical fluxes on the faces of the cell.

3.2 Advection fluxes discretization

Dealing with hypersonic flows, shock waves propagate in the flow field and create discontinuities in the numerical solution. This leads to the need of evaluating the sign of the eigenvalues to know the propagation direction of the waves. In order to do this, one can solve an approximate Riemann problem at each interface between the control volumes. As to do so, the solution is represented by an arbitrary polynomial function that reduces to an average between the surrounding values of the solution in case of first order schemes [114]. In such cases, upwind schemes are suitable to handle the numerical solution even in case of mixed sign eigenvalues, resulting in a more stable formulation if compared to centered schemes.

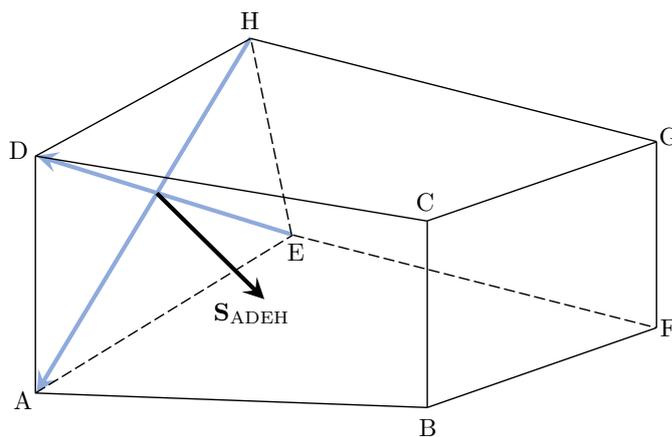


Figure 3.2: Example of a three-dimensional hexahedral volume.

Moreover, upwind schemes provide an affordable strategy if implicit time integration is employed. Indeed, the inversion of tridiagonal or pentadiagonal matrices is mandatory in case of centered schemes, resulting in a very tedious and expensive operation; on the contrary, the treatment of triangular matrices obtained through upwind schemes allows for saving computational time. Despite these pros, upwind schemes are based on the reconstruction of the solution from one side with respect to the interface. This obviously represents a limit of such formulations since one should account for the sign of the eigenvalues of the Jacobian matrix of the inviscid fluxes, responsible of the speed and direction of the signals propagation. For this reason, a *flux vector splitting* (FVS) is needed: this approach allows one to properly apply upwinding.

In this work, the formulation proposed by Steger and Warming is employed [114, 115]. The main idea is the splitting of the inviscid flux vectors into two contributions: one is linked to the positive eigenvalues (+) and the other is linked to the negative eigenvalues (-). In such a way, the fluxes can be reconstructed along the effective propagation direction of the signals, assumed to be normal to the interface. The total flux will be the sum of the two contributions, allowing for a proper formulation even in case of subsonic regions (mixed sign eigenvalues). As an example, Fig. 3.3 shows a 1D flux splitting. Considering Euler's equations, the system of partial differential equations (PDEs) to solve in each cell i is:

$$\frac{\partial \mathbf{U}_i}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \quad (3.6)$$

being $\mathbf{U} = [\rho, \rho u, \rho E]$ and $\mathbf{F} = [\rho u, \rho u^2 + p, (\rho E + p)u]$. After the spatial discretization, the system given by Eq. (3.6) becomes a system of ordinary differential equations (ODEs):

$$\frac{d\mathbf{U}_i}{dt} = -\frac{\mathbf{F}_{i+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}}}{\Delta x} \quad (3.7)$$

where $\mathbf{F}_{i\pm\frac{1}{2}}$ are the total flux vector contributions, computed as follows:

$$\mathbf{F}_{i\pm\frac{1}{2}} = \mathcal{F}_{i\pm\frac{1}{2}}^+ \left(\phi_{i\pm\frac{1}{2}}^l \right) + \mathcal{F}_{i\pm\frac{1}{2}}^- \left(\phi_{i\pm\frac{1}{2}}^r \right) \quad (3.8)$$

where the two contributions (+ and -) are functions of the the left and right reconstruction of the generic flow variable ϕ with respect to the surface, as shown in Fig. 3.3. Following the formulation by Steger and Warming [115], the generalized form of the splitting term \mathcal{F} for a 3D configuration (mT model) is:

$$\mathcal{F}^\pm = \frac{\rho}{2\bar{\gamma}} \begin{bmatrix} \beta \\ \beta u + c (\lambda_2^\pm - \lambda_3^\pm) n_x \\ \beta v + c (\lambda_2^\pm - \lambda_3^\pm) n_y \\ \beta w + c (\lambda_2^\pm - \lambda_3^\pm) n_z \\ \beta \left(\frac{u^2+v^2+w^2}{2} + \frac{\varepsilon}{\rho} \right) + cV (\lambda_2^\pm - \lambda_3^\pm) + c^2 \frac{\lambda_2^\pm + \lambda_3^\pm}{\bar{\gamma}-1} \\ \beta e_{vib,N_2} \\ \beta e_{vib,O_2} \\ \beta e_{vib,NO} \end{bmatrix} \quad (3.9)$$

where $\beta = 2(\bar{\gamma}-1)\lambda_1^\pm + \lambda_2^\pm + \lambda_3^\pm$, c is the speed of sound and $V = u n_x + v n_y + w n_z$ is the velocity normal to the face, being (n_x, n_y, n_z) the components of the unit vector normal to the cell face. For simplicity, in Eq. (3.9) symbols + and - have been reported only for the eigenvalues, λ , to distinguish positive and negative ones. Nevertheless, these eigenvalues are in turn linked to the reconstruction of the solution in a certain direction: indeed they read $(\lambda_1^\pm, \lambda_2^\pm, \lambda_3^\pm) = (V^\pm, V^\pm + c, V^\pm - c)$. Always recall that the last three terms in Eq. (3.9) are not considered in case of the StS model. Also, the splitting term of the species continuity equation can be obtained by substituting the mixture density, ρ , with the species density, ρ_s .

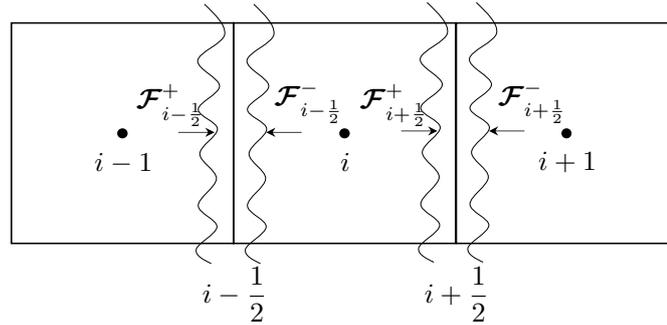


Figure 3.3: Interpretation of a 1D FVS.

The reconstruction of the solution is based on a second order MUSCL (*monotone upstream-centered schemes for conservation laws*) formulation [116]. Fig. 3.4 illustrates the spatial evolution of the generic flow variable ϕ . For each cell, the mean value is reported, justifying the step trend. This leads to discontinuities at each interface, whether they are expansions ($\phi_i > \phi_{i-1}$) or shocks ($\phi_i < \phi_{i-1}$), as highlighted by the blue lines. Assuming the the signal travels toward the positive

direction of the x axis, positive eigenvalues (+) are linked to the left state, whereas negative ones (−) to the right state, as also reported in Fig. 3.3. Following the notation in Fig. 3.4 and taking as an example the interface at $i + \frac{1}{2}$, the left and right states are:

$$\phi_{i+\frac{1}{2}}^r = \begin{cases} \phi_{i+1} & \text{first order reconstruction} \\ \phi_{i+1} - \frac{1+\varphi}{4}\delta\phi_{i+\frac{1}{2}} - \frac{1-\varphi}{4}\delta\phi_{i+\frac{3}{2}} & \text{higher order reconstruction} \end{cases} \quad (3.10a)$$

$$\phi_{i+\frac{1}{2}}^l = \begin{cases} \phi_i & \text{first order reconstruction} \\ \phi_i + \frac{1-\varphi}{4}\delta\phi_{i-\frac{1}{2}} + \frac{1+\varphi}{4}\delta\phi_{i+\frac{1}{2}} & \text{higher order reconstruction} \end{cases} \quad (3.10b)$$

where

$$\delta\phi_{i-\frac{1}{2}} = \phi_i - \phi_{i-1} \quad \delta\phi_{i+\frac{1}{2}} = \phi_{i+1} - \phi_i \quad \delta\phi_{i+\frac{3}{2}} = \phi_{i+2} - \phi_{i+1}$$

and φ is an integer that defines the accuracy of the scheme. In case of higher order schemes, a second order fully upwind reconstruction is ensured when $\varphi = -1$, as in this work.

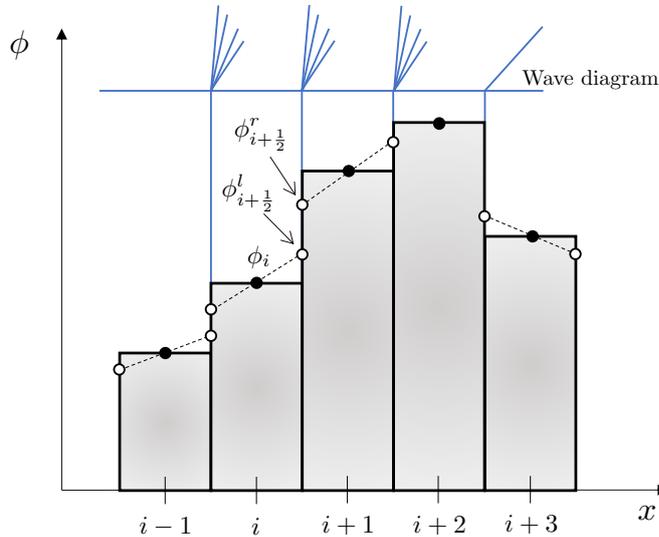


Figure 3.4: Piecewise linear reconstruction for MUSCL formulation. Blue lines represent the wave diagram (three expansions and one shock).

Despite the higher accuracy with respect to first order schemes, it is known from Godunov's theorem that linear higher order schemes do not ensure a monotone solution. It is evident also from Fig. 3.4: the reconstruction on interface $i + \frac{3}{2}$ would not be monotone as it can be evinced from the wave diagram (shock+expansion) and this might produce spurious oscillations near the discontinuity [117]. It is

then affordable exploiting a TVD (*total variation diminishing*) formulation: *flux limiter* functions are introduced in order to reduce the accuracy of the scheme near the discontinuities. This allows to preserve a globally second order accurate scheme, with stable properties near discontinuities (first order). Hence, higher order reconstruction expressions in Eq. (3.10) become:

$$\phi_{i+\frac{1}{2}}^r = \phi_i - \frac{1+\varphi}{4}\Phi(r_1)\delta\phi_{i+\frac{1}{2}} - \frac{1-\varphi}{4}\Phi(r_2)\delta\phi_{i+\frac{3}{2}} \quad (3.11a)$$

$$\phi_{i+\frac{1}{2}}^l = \phi_{i-1} + \frac{1-\varphi}{4}\Phi(r_3)\delta\phi_{i-\frac{1}{2}} + \frac{1+\varphi}{4}\Phi(r_4)\delta\phi_{i+\frac{1}{2}} \quad (3.11b)$$

where

$$r_1 = \frac{\delta\phi_{i+3/2}}{\delta\phi_{i+1/2}} \quad r_2 = \frac{\delta\phi_{i+1/2}}{\delta\phi_{i+3/2}} \quad r_3 = \frac{\delta\phi_{i+1/2}}{\delta\phi_{i-1/2}} \quad r_4 = \frac{\delta\phi_{i-1/2}}{\delta\phi_{i+1/2}}$$

and $\Phi(r)$ is the *flux limiter* function. Following *minmod* formulation, $\Phi(r) = \frac{r^2+r}{1+r^2}$. In such a way, the numerical scheme turns out to be stable also in case of mixed sign eigenvalues. It provides high accuracy and stability at the same time, preserving the upwind nature of the fluxes formulation.

3.3 Diffusive fluxes discretization

While discretizing viscous terms, interface gradients must be evaluated. Specifically, velocity gradients and temperature gradients are needed for the calculation of the stress tensor and the heat flux. For this purpose, generalized Gauss' theorem is employed. It allows for transforming a volume integral into a surface integral, as follows:

$$\int_{\mathcal{V}} \nabla\phi d\mathcal{V} = \int_S \phi \mathbf{n} dS \quad (3.12)$$

where ϕ is the generic flow variable. The main idea is to compute gradients on each face of the control volumes. Fig. 3.5 shows two adjacent control volumes (in the computational domain) having a common interface, highlighted in red. As an example, fluxes in the i direction are considered (index i is kept constant).

Discretizing Eq. (3.12), one obtains:

$$\nabla\phi = \frac{1}{\mathcal{V}_G} \sum_{f=1}^{N_f} \phi_f \bar{\mathbf{S}}_f \quad (3.13)$$

where \mathcal{V}_G and $\bar{\mathbf{S}}_f$ are the volume and the surface normal vector of the *Gauss cell*, composed of N_f faces. For i fluxes formulation, one has to operate on the jk plane, as shown in Fig. 3.6. Dashed lines identify the *Gauss cell*, extending from point 5

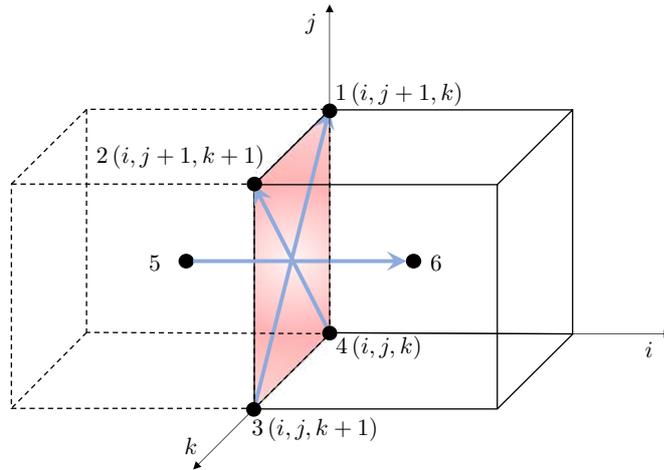


Figure 3.5: Sketch of the computational domain to apply Gauss' theorem.

to point 6. Surface normal vectors can be computed by means of cross products, as follows:

$$\bar{\mathbf{S}}_1 = \mathbf{r}_{42} \times \mathbf{r}_{56} \quad \bar{\mathbf{S}}_2 = \mathbf{r}_{56} \times \mathbf{r}_{31} \quad \bar{\mathbf{S}}_6 = \mathbf{r}_{31} \times \mathbf{r}_{42}$$

and obviously $\bar{\mathbf{S}}_3 = -\bar{\mathbf{S}}_1$, $\bar{\mathbf{S}}_4 = -\bar{\mathbf{S}}_2$ and $\bar{\mathbf{S}}_5 = -\bar{\mathbf{S}}_6$. In Eq. (3.13), ϕ_f represents the value of the generic flow variable in the point f . Since points 5 and 6 correspond to cell centers, ϕ_5 and ϕ_6 are obtained directly from the flow field, whereas values in the nodes (ϕ_1 , ϕ_2 , ϕ_3 and ϕ_4) are evaluated by interpolating the values of ϕ in the surrounding cells, in order to obtain second order accuracy.

3.4 Time integration and source terms evaluation

Once the spatial discretization is performed, governing equations must be integrated in time. In this scenario, it is worth specifying that, dealing with non-equilibrium flows, kinetics characteristic time can be much shorter than the fluid dynamics one, making the problem stiff. This would imply a reduction of the time step size to ensure stability while evaluating thermochemical source terms. This reduction could be very restrictive, making the simulation too demanding. For this reason, a splitting approach is employed [118]. Time advancement is separated into two different steps: the first one is devoted to the fluid dynamics, focusing on the solution of the homogeneous equations; the second one concerns the computation of thermochemical terms to account for non-equilibrium. More details are given in the next subsections.

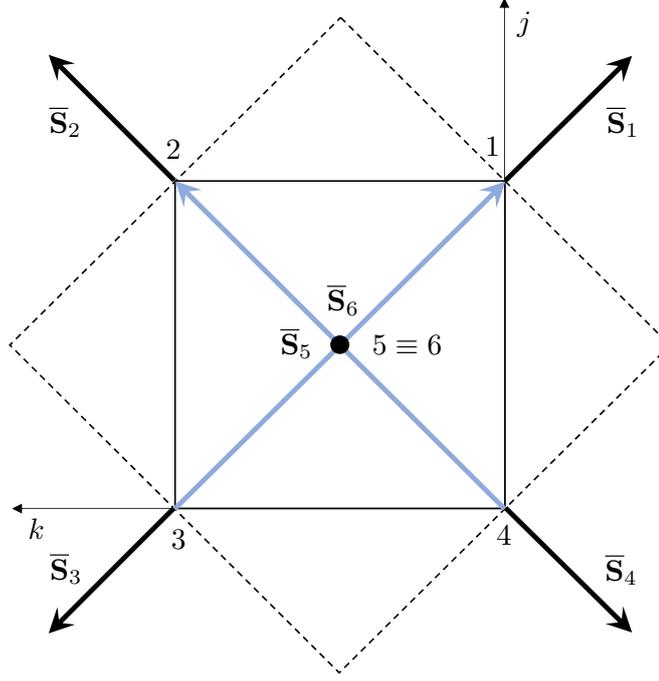


Figure 3.6: Interpretation of Gauss' theorem for viscous fluxes formulation along i direction.

3.4.1 First step: fluid dynamics

In the first step the homogeneous equations are solved, thus a non-reacting mixture is considered and vibrational excitation is neglected (null source terms). In this way, advective and diffusive characteristics are advanced in time with a time step size based on the CFL (Courant-Friedrichs-Lewy) condition [119]:

$$\Delta t_{i,j,k} = \frac{\text{CFL } \mathcal{V}_{i,j,k}}{|\mathbf{u}_{i,j,k} \cdot \bar{\mathbf{S}}_i| + |\mathbf{u}_{i,j,k} \cdot \bar{\mathbf{S}}_j| + |\mathbf{u}_{i,j,k} \cdot \bar{\mathbf{S}}_k| + c_{i,j,k} (\bar{\mathbf{S}}_i + \bar{\mathbf{S}}_j + \bar{\mathbf{S}}_k)} \quad (3.14)$$

where CFL is the Courant number, whereas $c_{i,j,k}$ and $\mathbf{u}_{i,j,k}$ are the speed of sound and the flow velocity vector in the cell (i, j, k) , respectively; $\bar{\bullet}$ stands for average values between two consecutive faces ($\bar{\mathbf{S}}$ are the surface normal vectors, whereas $\bar{\mathbf{S}}$ are the surface extension). Eq. (3.14) provides the time step size for each control volume: at this stage, two possible strategy can be applied, depending on the nature of the problem. In case of stationary problems, a *local time stepping* approach can be employed, so that the solution in each control volume is advanced in time with the corresponding Δt (faster convergence). On the other hand, when dealing with unsteady flows, time accuracy is necessary to opportunely capture the transient state: for this purpose, a *global time stepping* approach allows to set a unique time step size for all the control volumes, computed as the minimum among all the cells.

Once the time step size is evaluated, an explicit third order Runge-Kutta scheme [120] is employed to solve the system of homogeneous ordinary differential equations (ODEs) given by:

$$\frac{d\mathbf{U}_{i,j,k}}{dt} = -\frac{1}{\mathcal{V}_{i,j,k}} \sum_{f=1}^{N_f} \mathbf{F}_f \cdot \mathbf{n}_f S_f \quad (3.15)$$

whose integration is:

$$\begin{aligned} \mathbf{U}^{(1)} &= \mathbf{U}^n + \Delta t \mathcal{R}(\mathbf{U}^n) \\ \mathbf{U}^{(2)} &= \frac{3}{4}\mathbf{U}^n + \frac{1}{4}\mathbf{U}^{(1)} + \frac{1}{4}\Delta t \mathcal{R}(\mathbf{U}^{(1)}) \\ \mathbf{U}^{n+1} &= \frac{1}{3}\mathbf{U}^n + \frac{2}{3}\mathbf{U}^{(2)} + \frac{2}{3}\Delta t \mathcal{R}(\mathbf{U}^{(2)}) \end{aligned}$$

where $\mathcal{R}(\mathbf{U})$ is the spatial discretization operator (spatial indices i , j and k are omitted for simplicity); n and $n + 1$ are respectively the old and new time step, whereas quotes (1) and (2) indicates the first and second Runge-Kutta substeps. Despite the limitations on the time-step size introduced by an explicit formulation, this allows to fully exploit GPU capabilities: indeed, CUDA parallel programming is less efficient when complicated floating point operations are requested, such as a matrix inversion in case of an implicit formulation.

3.4.2 Second step: thermochemistry

In the second step, thermochemical terms, namely the species densities (and the molecule vibrational energies in case of the mT model), are updated. Hence, the system of equations to solve for each cell is:

$$\frac{d\mathbf{y}}{dt} = \mathbf{P}(\mathbf{y}) - \bar{\bar{\mathbf{L}}}(\mathbf{y}) \cdot \mathbf{y} \quad (3.17)$$

where $\mathbf{y} = [\rho_1, \dots, \rho_{N_s}, e_{vib,m}, \dots, e_{vib,N_m}]^T$ for the mT model, whereas $\mathbf{y} = [\rho_{1,1}, \dots, \rho_{1,V_1}, \dots, \rho_{N_s,V_{N_s}}]^T$ for the StS model. \mathbf{P} and $\bar{\bar{\mathbf{L}}}$ are, respectively, a vector and a diagonal matrix whose components are nonnegative and represent production and loss terms for a certain species. After discretizing Eq. (3.17), one obtains:

$$\left(\bar{\bar{\mathbf{I}}} + \Delta t_c^{(v)} \bar{\bar{\mathbf{L}}}(\mathbf{y})\right) \cdot \mathbf{y}(t + \Delta t_c^{(v)}) = \Delta t_c^{(v)} \mathbf{P}(\mathbf{y}) + \mathbf{y}(t) \quad (3.18)$$

The calculation of production and loss terms is straightforward in case of the mT model, for which \mathbf{P} and \mathbf{L} collect all the terms multiplying k_b and k_f respectively in Eq. (2.36). On the other hand, they are harder to write in case of the

StS model, as the reaction rates follow different kind of formulations depending on the nature of the mechanism (VV, VTm, VTa, etc.) and they need an opportune expression that also involves the vibrational levels. Lastly, despite the non linearity of Eq. (3.18), its inversion is trivial since this represents a system of uncoupled equations. Indeed, the solution of Eq. (3.18) is found applying an iterative Gauss-Seidel scheme, that provides the solution of the generic species s at the inner iteration k for each control volume (indices i , j and k are omitted for simplicity) as:

$$y_s^k(t + \Delta t_c^{(v)}) = \frac{\Delta t_c^{(v)} P_s(\mathbf{y}^{k-1}) + y_s(t)}{1 + \Delta t_c^{(v)} L_s(\mathbf{y}^{k-1})} \quad (3.19)$$

The number of inner iterations depends on the relevance of thermochemical non-equilibrium: in the present work, it is 4 for most of the cases, having demonstrated that 8 inner iterations provide the same results. In Eqs. (3.18,3.19), $\Delta t_c^{(v)}$ represents the 'chemical' time step size. In order to account for the stiffness of the problem, thermochemical source terms are advanced in time with a time step size that is a portion of the fluid dynamics one, Δt_f , computed from Eq. (3.14):

$$\sum_{v=1}^{N_{kin}} \Delta t_c^{(v)} = \Delta t_f$$

where N_{kin} is the number of sub-steps. This normally varies in the range $1 \div 8$, depending on the stiffness of the problem. In general, it is reasonable to set $\Delta t_c^{(v)} = \Delta t_f / N_{kin}$. Such a splitting approach has been found to be suitable when solving reactive flows [118]. This allows for a higher time step size when solving the first step, while preserving accuracy in the evaluation of the kinetic source terms. Moreover, a fully explicit algorithm is suitable in GPU programming, as previously mentioned. Finally, it was found that its accuracy is perfectly comparable to that of an implicit formulation [23].

3.5 Boundary conditions

The boundary conditions are applied on the conservative variables, species partial densities and, in case of isothermal wall, on the temperature. In order to be consistent with the numerical scheme, two ghost cells are defined at the boundaries.

3.5.1 General boundary conditions

The imposition of the boundary conditions is based on the propagation of the characteristics (characteristic boundary conditions [121–124]).

Supersonic inflow

Supersonic flow condition is imposed at the inlet. Velocity, pressure and species concentration (densities) are given as input. All the other variables are then calculated. These are never altered during the simulation.

Supersonic outflow

Since the characteristics travel in a unique way, all the waves at the boundaries leave the domain without reflection. Hence, the conservative variables are copied from the flow field to the ghost cells. Indicating with ϕ the generic conservative variables:

$$\phi_{g_1} = \phi_{g_2} = \phi_{f_1}$$

being g_1 and g_2 the two ghost cells, whereas f_1 is the first internal cell (fluid domain).

Symmetry

As for the supersonic outflow, this boundary condition copies the values of the conservative variables from the flow field to the ghost cells. Nevertheless, to ensure symmetry, a specific velocity component is overturned depending on the shape of the body. For example, considering the generic flow velocity vector $\mathbf{u} = (u, v, w)$, symmetry is ensured along the stagnation line of a cylinder or a sphere by imposing:

$$\begin{aligned}(\rho v)_{g_1} &= -(\rho v)_{f_1} \\ (\rho v)_{g_2} &= -(\rho v)_{f_2}\end{aligned}$$

This allows to simulate a quarter of a cylinder or a sphere. Obviously, in order to simulate an eighth of a sphere, the same condition must be imposed for w along k direction, namely:

$$\begin{aligned}(\rho w)_{g_1} &= -(\rho w)_{f_1} \\ (\rho w)_{g_2} &= -(\rho w)_{f_2}\end{aligned}$$

3.5.2 Wall boundary conditions

No-slip

No-slip boundary condition is imposed to account for a viscous wall. The only difference with respect to the supersonic outflow condition is that the velocity components are all overturned to ensure zero velocity on the wall:

$$\begin{aligned} (\rho u)_{g_1} &= -(\rho u)_{f_1} & (\rho v)_{g_1} &= -(\rho v)_{f_1} & (\rho w)_{g_1} &= -(\rho w)_{f_1} \\ (\rho u)_{g_2} &= -(\rho u)_{f_2} & (\rho v)_{g_2} &= -(\rho v)_{f_2} & (\rho w)_{g_2} &= -(\rho w)_{f_2} \end{aligned}$$

Moreover, in case of isothermal wall, wall temperature is kept constant by extrapolating its value in the first ghost cell:

$$T_{g_1} = 2T_w - T_{f_1}$$

being T_w the fixed wall temperature value. Since thermal equilibrium is imposed at wall ($T = T_{vib}$), the same formula is applied for vibrational temperatures. On the other hand, in case of adiabatic wall, null temperature gradient is imposed along the wall normal direction by imposing null viscous fluxes for the energy equation and, in case of the mT model, for the vibrational energy equations (this *de facto* corresponds to impose a null heat flux). Note that, in this formulation, the grid spacing of the first ghost cell is the same of the first fluid cell to ensure that the gradients are based on the same local discretization. Also, copying the value of the energy conservative variable in the ghost cells yields to a zero pressure gradient along the wall normal direction, that is known to be approximately correct for boundary layers.

Finally, a boundary condition for the species mass fractions is also needed. As previously mentioned, it is imposed on the species partial densities by copying their values from the flow field to the ghost cells. Since the mixture density is also copied, null mass fractions gradient is ensured along the wall normal direction.

Catalysis

In the standalone version of the code, a fully-catalytic model is implemented. Since the wall temperatures considered in this work never exceed 1000 K, the equilibrium composition is imposed for N_2 and O_2 . Hence, the mass fractions in the ghost cells are extrapolated from the fluid:

$$Y_{s,g_1} = 2Y_{s,w} - Y_{s,f_1}$$

where $Y_{s,w}$ is the wall mass fraction of species s (0.767 for N_2 and 0.233 for O_2). Of course, in case of a binary mixture, $Y_{s,w}$ is 1 for the molecule involved. Moreover, the GSI module from MUTATION⁺⁺ allows to handle partially-catalytic walls.

Ablation

Ablative boundary condition is applied thanks to MUTATION⁺⁺ GSI module. The mass balance is solved to compute the massblowing rate and the mass fractions to be imposed at wall (see section 2.5.1). This allows to evaluate the gas velocity u_g in Eq. (2.51), which is imposed on the wall to account for the blowing effect.

Chapter 4

GPU programming

Nowadays, computational algorithms are becoming more and more efficient thanks to advanced techniques that allow to speed up the calculations. One of the best way to increase the efficiency of a fluid dynamics solver is the General-Purpose computing on Graphical Processing Units (GPGPUs). Such an approach allows to exploit GPU capabilities to execute thousands of tasks in parallel.

In principle, GPUs were born for video rendering. However, several libraries have been developed to extend their use to general purposes, such as mathematics and physics problems. One of the best environment is the *Compute Unified Device Architecture* (CUDA) that allows to write a GPU code in a very simple way, starting from a C/Fortran based prototype [125, 126]. The thousands of parallel tasks executed by a GPU lead to an effective speed-up that represents the kill feature of many current solvers [127, 128].

This chapter is dedicated to the description of CUDA programming and hardware characteristics. As it will be explained, the software configuration of the code depends on the hardware characteristics.

4.1 CUDA programming

By the time, GPUs have become competitive in the scientific computing field. One of the most important companies is NVIDIA Corporation, which is still actively taking care of the graphical user interface of many daily devices, such as laptops and gaming consoles. The first version of CUDA was developed many years ago and NVIDIA has been working to improve its performances and compatibility with classical programming languages such as C/C++ or Fortran. Indeed, the development of a GPU function (*device kernel* or simply *kernel*) does not actually require a relevant effort, as it will be shown in the following.

In order to better understand how a CUDA program works, the reader can refer to Fig. 4.1. A GPU program always needs a standard initialization on CPUs, whose memory is referred to as *host*. Then, the most computationally intensive part is assigned to GPUs, where memory is also allocated (*device* memory). Once

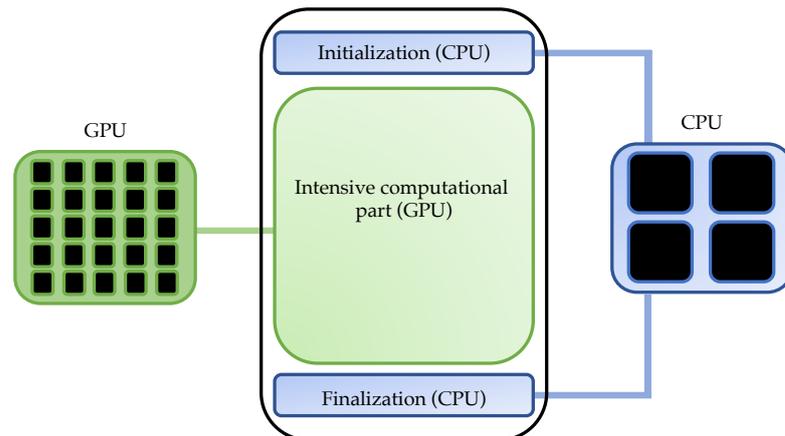


Figure 4.1: Overall assignment of the tasks to GPUs and CPUs.

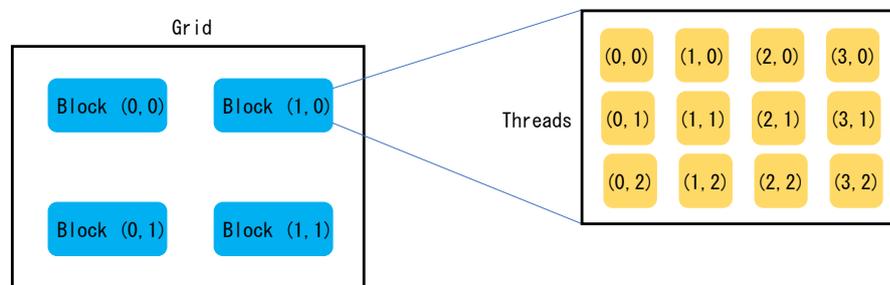


Figure 4.2: Hierarchy of a GPU.

the calculation is over, CPUs handle the finalization including management of output files.

In Fig. 4.1 one can notice that the GPU is represented by many 'black elements'. They represent the *threads* of the GPU within a *block*. Along with the *grid* of the blocks, these represent the main components in the hierarchy of a GPU and their definitions are here given. Fig. 4.2 illustrates a generic structure of these components.

- *Thread*: the lowest level in the hierarchy, a thread represents a programming abstraction to which a certain operation is assigned (for instance, a sum or a product).
- *Block*: it is a group of threads that can communicate with each other. Currently, a block can contain up to 1024 threads, but it depends on the hardware [125].
- *Grid*: ensemble of blocks to which the kernel invocation is assigned. Currently, a grid can contain up to 65535 blocks [125].

In CUDA language threads and blocks are identified by an integer `threadIdx` and `blockIdx`. The total number of thread in a block is identified by `blockDim`,

whereas the total number of blocks in a grid is identified by the integer `gridDim`. They are built-in parameters of CUDA, and are used for the setting of loop counters. More detail about the development of a kernel are given in section 4.2.

4.2 Example of a kernel and indexing

In this section, a basic example of a kernel is given in listing 4.1 to highlight the few differences between CPU and GPU programming. In such an example, two arrays `a` and `b` are composed of integer elements and must be summed each other to obtain a third array `c`. So, once the dimension is defined ($N=1000$) they are declared and allocated. Note that for CPUs a static allocation is performed, whereas for GPUs they are dynamically allocated by means of the built-in CUDA function `cudaMalloc`. Before performing the sum, they are initialized on the CPU with certain values. At this point, they are copied from host to device memory with the command `cudaMemcpy`. Such a function is useful also in the other way around, since output files have to be written on CPU memory, for which a copy of the variables from device to host is needed.

The call to the function to perform the sum is quite similar between CPU and GPU, except for the lettering `<<<B,T>>>` that represents the number of blocks and threads assigned to this kernel. The number of threads must never exceed the `maxThreadsPerBlock` parameter. The total number of threads that will execute the kernel is given by $B \times T$.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 __global__
5 void sum_GPU(int *a, int *b, int *c, int N) {
6     int i = threadIdx.x;
7     if(i < N){
8         c[i] = a[i] + b[i];
9     }
10 }
11
12 void sum_CPU(int *a, int *b, int *c, int N) {
13     int i = 0;
14     while(i < N){
15         c[i] = a[i] + b[i];
16         i += 1;
17     }
18 }
19
20 int main(void) {
21
22     // Declaration of the arrays
23     const int N = 1000;           // This will be the size
24     int a[N], b[N], c[N];       // Host variables (CPU)
```

```

25     int *a_dev , *b_dev , *c_dev;      // Device variables (GPU)
26     int B=1, T=N;                    // Blocks and Threads for kernel
execution
27
28     // Initialize host variables
29     for (int i = 0; i < N; ++i)
30     {
31         a[i] = i + 1;
32         b[i] = i + 2;
33         c[i] = 0;
34     }
35
36     // Allocate device variables and copy from host
37     cudaMalloc((void**)&a_dev ,(N)*sizeof(int));
38     cudaMalloc((void**)&b_dev ,(N)*sizeof(int));
39     cudaMalloc((void**)&c_dev ,(N)*sizeof(int));
40     cudaMemcpy(a_dev ,a ,(N)*sizeof(int) ,cudaMemcpyHostToDevice);
41     cudaMemcpy(b_dev ,b ,(N)*sizeof(int) ,cudaMemcpyHostToDevice);
42     cudaMemcpy(c_dev ,c ,(N)*sizeof(int) ,cudaMemcpyHostToDevice);
43
44     // Perform the sum
45     sum_CPU(a ,b ,c ,N);
46     sum_GPU<<<B,T>>>(a_dev , b_dev , c_dev ,N);
47
48     cudaFree(a_dev);
49     cudaFree(b_dev);
50     cudaFree(c_dev);
51
52     return 0;
53
54 }

```

Listing 4.1: Example code: sum of two arrays

As one can notice, the differences between a CPU function and a GPU kernel are minimal. The lettering `__global__` identifies the beginning of a `void()` kernel. This becomes `__device__` in case of a specific kernel type (`int()`, `double()`, etc.) and it are callable only by other `__device__` or `__global__` functions, making impossible their call from host [125]. The loop is almost the same of a classical CPU function, except for the definition of the counter. This is linked to a specific thread, and so initialized as `threadIdx.x`. The suffix `.x` represents a certain direction in the topology of a block. Fig. 4.2 reports a 2D topology, but the threads are generally distributed into a 3D structure. Their distribution can be simplified performing the 1D indexing [125], for which a sketch is given in Fig. 4.3. A generic array is composed of 1000 elements as in listing 4.1. At this stage there are many combinations of blocks and threads to deal with such a computation. The simplest one is to assign all the elements to the same block and run the kernel with 1000 threads. In such a case there would be only 1 block (`gridDim.x=1,blockIdx.x=0`) with 1000 threads (`blockDim.x=1000`). In such a case, the loop counter is just the

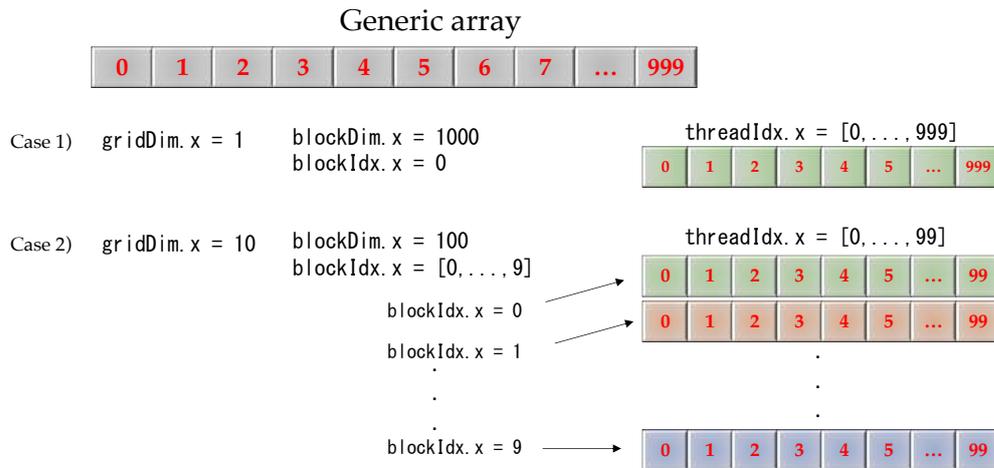


Figure 4.3: Examples of 1D indexing.

integer identifying the thread, namely `threadIdx.x`. However, since the number of threads per block depends on the hardware, another possibility is to distribute the elements among more blocks. In this case, 10 blocks can be chosen (`gridDim.x=10`) each of which would contain 100 elements (`blockDim.x=100`). Then, the counter must be defined as a combination of `threadIdx.x`, `blockDim.x` and `blockIdx.x`, as shown in listing 4.2.

```

1 __global__
2 void sum_GPU(int *a, int *b, int *c, int N) {
3     int i = threadIdx.x + blockIdx.x * blockDim.x;
4     while(i < N){
5         c[i] = a[i] + b[i];
6         i += blockDim.x * gridDim.x;
7     }
8 }

```

Listing 4.2: Example of a different definition of the loop counter

At this point, two considerations about the number of blocks involved are mandatory. First, in the example of Fig. 4.3, there are actually many other possibilities to distribute the elements, for example imposing `gridDim.x=5` and `blockDim.x=200`. For this reason, the estimation of the number of blocks and threads is often generalized basing on the number of elements. A common way is to fix the number of threads, thus $T=128$. If an array is composed of N elements, the number of blocks can be computed as $B=(N+T-1)/T$. This operation ensures to launch the kernel with a suitable number of threads. However, it is very common that the number of threads is larger than the actual size of the array. For this reason, the condition for which $i < N$ avoids to perform the calculations if the thread is out of the bound of the array.

The second consideration is linked to the maximum number of blocks that can be invoked, that is 65535 as previously mentioned. Hence, the formula just shown

to compute the number of blocks must be revised. A possibility is $B = \min(65535, (N+T-1)/T)$. In this scenario, a limitation on the number of blocks does not ensure a suitable number of threads. Hence, as shown in listing 4.2, an increment of the counter is needed if the total number of threads is lower than the total number of elements of the array, namely $B \times T < N$. Then the loop counter increment is given by the total number of threads in the grid. This allows to use the threads more than once.

4.3 MPI-CUDA environment

Fluid dynamics solvers often take advantage of external libraries that allow parallel computing. Especially when dealing with complex fluid flows, parallelization is mandatory to reduce the computational cost of a simulation. *Message Passing Interface* (MPI) or *Open Multi-Processing* (OpenMP) libraries represent a powerful tool for making simulations faster. Thanks to their interface with basic programming languages, these libraries make processors communicate with each other for information exchange. In the framework of CUDA programming, multi-GPU executions are also possible [34, 129, 130].

When operating on GPUs, the main drawback of the MPI library is the inability to make GPUs directly communicate with each other since such a library is designed to work on classical CPU (host memory). Hence, even when dealing with GPU programming, one should always perform the communications on the host memory, thus requiring the copy of object variables from the device memory. The recent CUDA-Aware-MPI library allows to overcome this tedious operation. This approach allows GPU-GPU direct communications, with no need of copying variables from device to host memory. An indicative 2D communicator is shown in Fig. 4.4, being the 3D extension straightforward.

First of all, a Cartesian communicator is created on CPUs through the MPI command `MPI_Cart_comm`, along with `MPI_Cart_coords` and `MPI_Cart_shift` that allow to identify the adjacent processors. This operation assigns a specific portion of the computational domain to a specific CPU, identified by 2 integers (3 in a 3D configuration) representing the position of the processor in the Cartesian topology. At this point, each GPU is associated to a specific CPU. The command `cudaGetDeviceCount` allows to find the total number of GPU within the same cluster node; they are then assigned to the tasks through the command `cudaSetDevice`.

In order to make GPU allocations compatible with the CUDA-Aware-MPI library, the object variables should be allocated with the command `cudaMallocHost`, so that the information exchange travels through GPUs with no need of involving CPUs. Nevertheless, it was found that this kind of allocations drastically slows down the calculation. For this reason, the main communications object arrays are allocated using the `cudaMalloc` command, whereas their 'twins' are allocated with

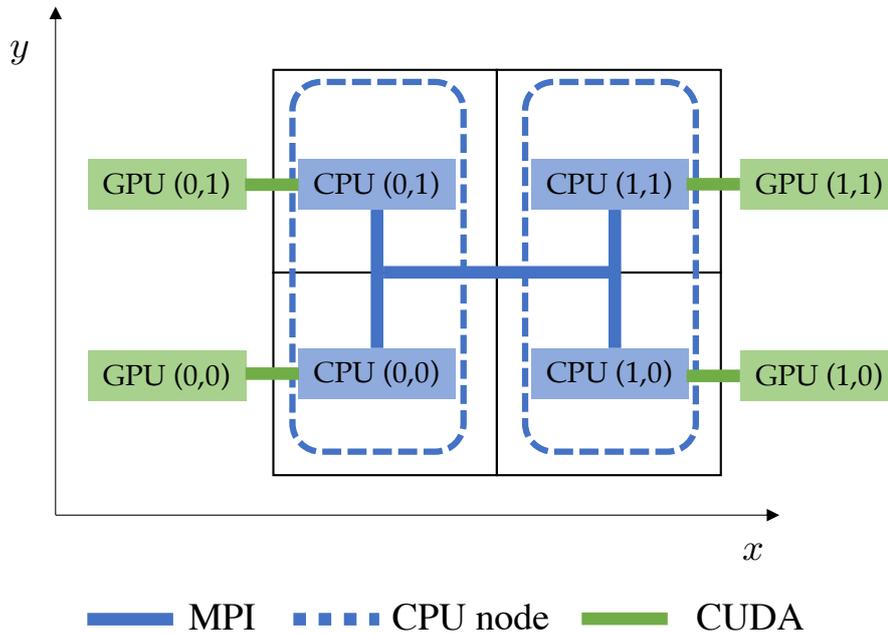


Figure 4.4: Example of a 2D MPI-CUDA environment.

`cudaMallocHost`. Before communications, their values are copied to allow values transfer.

4.4 CPU vs GPU

The benefit of using GPUs in the computational science field is not only linked to the smaller simulation time. Indeed, Graphics Processing Units require a lower energy consumption with respect to classical Central Processing Units. This section is devoted to a comparison between CPU and GPU executions.

The efficiency of the GPUs turns out to be more and more relevant when the complexity of the simulation progressively increases. A 2D flow was simulated on the cluster at Politecnico di Bari, composed of 6 GPU nodes, each one is equipped with two NVIDIA Tesla K40m (235 W), two Intel(R) Xeon(R) CPU E5-2630 v2 2.60 GHz (80 W) and 64 GB of RAM memory. The computational grid involved 1024×512 control volumes. Non-equilibrium was modeled with the mT and the StS approach for a 5 species neutral air mixture. A perfect gas flow (frozen) was also simulated. The times per iteration (TpI) are reported in table 4.1. As expected, the computational cost of a frozen simulation is not too high if compared to mT and StS simulations. Serial CPU executions with the mT model are still feasible, but the application of the StS model is prohibitive. However, single GPU execution times are quite smaller. More in detail, the speed-up obtained using the StS model is about 140 when employing one GPU rather than one CPU core. Also for the mT model the speed-up is relevant (about 44 times faster on one GPU).

Concerning the parallel executions, the code shows an almost linear scalability. Indeed, simulations performed on 12 GPUs are approximately 10 times faster with respect to the single GPU simulations for all the three models. In such a case, the benefit of using GPUs becomes extreme for the StS model: the computational cost is 25 times lower when using 12 GPUs instead of 72 CPU cores (6 nodes). This ratio is just 5 in case of mT and frozen simulations. A detailed scalability analysis of the 3D code is given in section 4.5.

Table 4.1: Times per iteration (TpI) for a 1024×512 computational domain. Times are given in seconds.

Model	Single-core CPU	72 cores CPU	1 GPU	12 GPUs
Frozen	2.36	0.0412	0.0545	0.0087
mT	15.9	0.246	0.361	0.0348
StS	107000	1700	764	68.7

Finally, it is noteworthy that, even if the power of the GPUs (235 W) is higher than that of the CPUs (80 W), the total energy consumption required by GPUs is lower. The energy consumption (in Joule) and the time per iteration (in seconds) for the serial execution of the StS model in table 4.1 are shown in a bar plot in Fig. 4.5. GPU execution is not only 140 times faster, but also 48 times more efficient of the CPU execution.

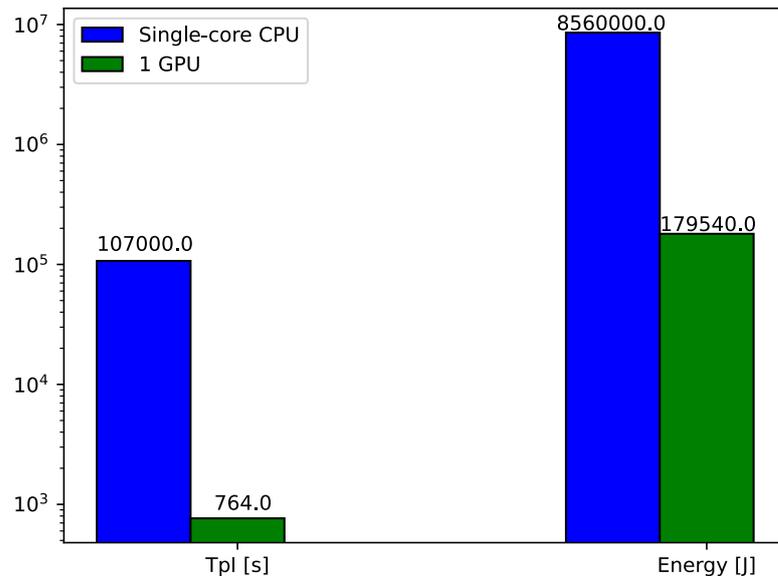


Figure 4.5: Comparison of TpI and energy consumption (serial StS run from table 4.1).

4.5 Scalability analysis

A preliminary performance analysis was conducted to test the scalability of the code. Two kind of scalability are distinguished, namely the *strong* and the *weak* scalability [131]. The first one describes the reduction of the execution time by increasing the number of processors (GPUs in this case) while keeping the size of the problem (namely the mesh for CFD problems) constant. Indicating with t_{ex} the execution time (wall-clock time), with S_p the size of the problem and with n_p the number of processors, the obtained speed-up can be defined as:

$$\text{speed-up} = \frac{t_{ex}(S_p, 1)}{t_{ex}(S_p, n_p)} \quad (4.1)$$

being $\text{speed-up}_{id} = n_p$ in the ideal case. In other words, for the same size of the problem, the time required by a 2 processors simulation should be the half of that required for serial execution. Hence, the efficiency is defined as:

$$\eta_s = \frac{\text{speed-up}}{\text{speed-up}_{id}} = \frac{t_{ex}(S_p, 1)}{n_p \times t_{ex}(S_p, n_p)} \quad (4.2)$$

On the other hand, the *weak* scalability gives indications about the ability of the code to keep the same execution time when changing the size of the problem and the number of processors of the same factor. Specifically, in the ideal case it should turn out that $t_{ex}(S_p, n_p) = t_{ex}(2 \times S_p, 2 \times n_p) = t_{ex}(3 \times S_p, 3 \times n_p) = \dots$. Therefore, its efficiency is defined as:

$$\eta_w = \frac{t_{ex}(S_p, 1)}{t_{ex}(n_p \times S_p, n_p)} \quad (4.3)$$

Table 4.2 reports the problem sizes for the performance analysis. Both the strong and the weak scalability have been investigated on Marconi100, a cluster at Cineca in Italy. It is based on POWER9 chips accelerated by Volta NVIDIA GPUs and includes 980 nodes equipped with two 16 cores IBM POWER9 AC922 at 2.6 (3.1) GHz and four NVIDIA Volta V100 GPUs with 16 GB of dedicated RAM [132].

Table 4.2: Problem sizes for the performance analysis.

Strong scalability (3D domain)	Weak scalability (volumes per GPU)
$1024 \times 1024 \times 512$	262144
$1024 \times 512 \times 512$	524288
$1024 \times 512 \times 256$	1048576
$512 \times 256 \times 256$	

Fig. 4.6 shows the results for the strong scalability analysis. Even if not linear, the scalability provides satisfactory efficiencies for all the cases. Fig. 4.7 shows

the results for the weak scalability analysis to perform 100 iterations. The results demonstrate that the execution time is practically constant by increasing the size of the problem and the number of GPUs, confirming the capability of the algorithm.

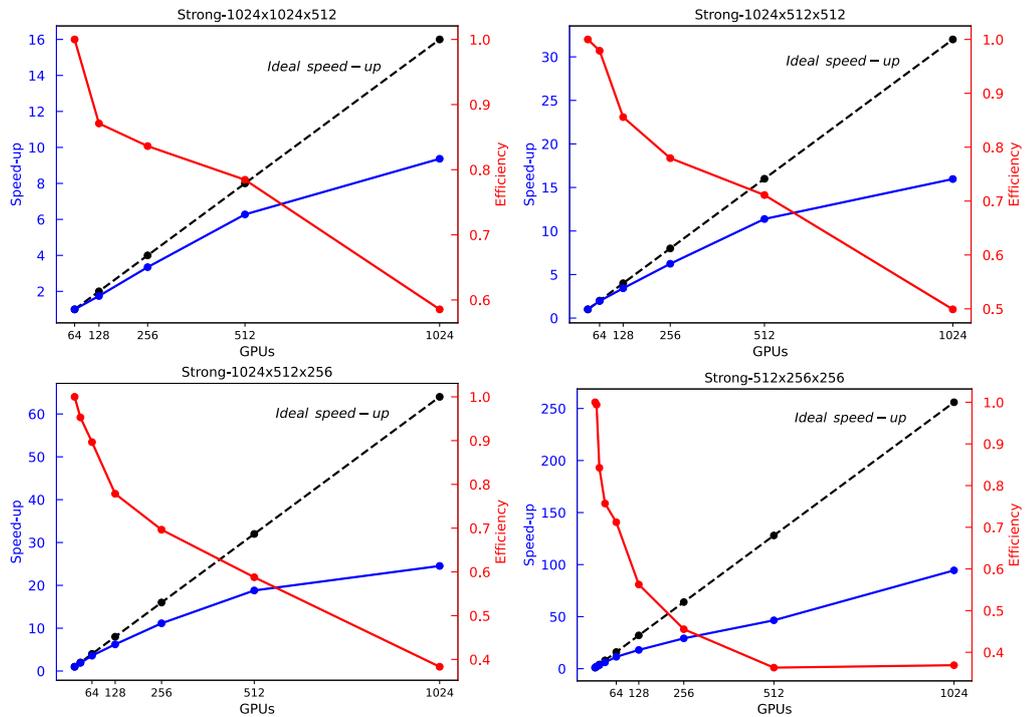


Figure 4.6: Strong scalability for the four cases.

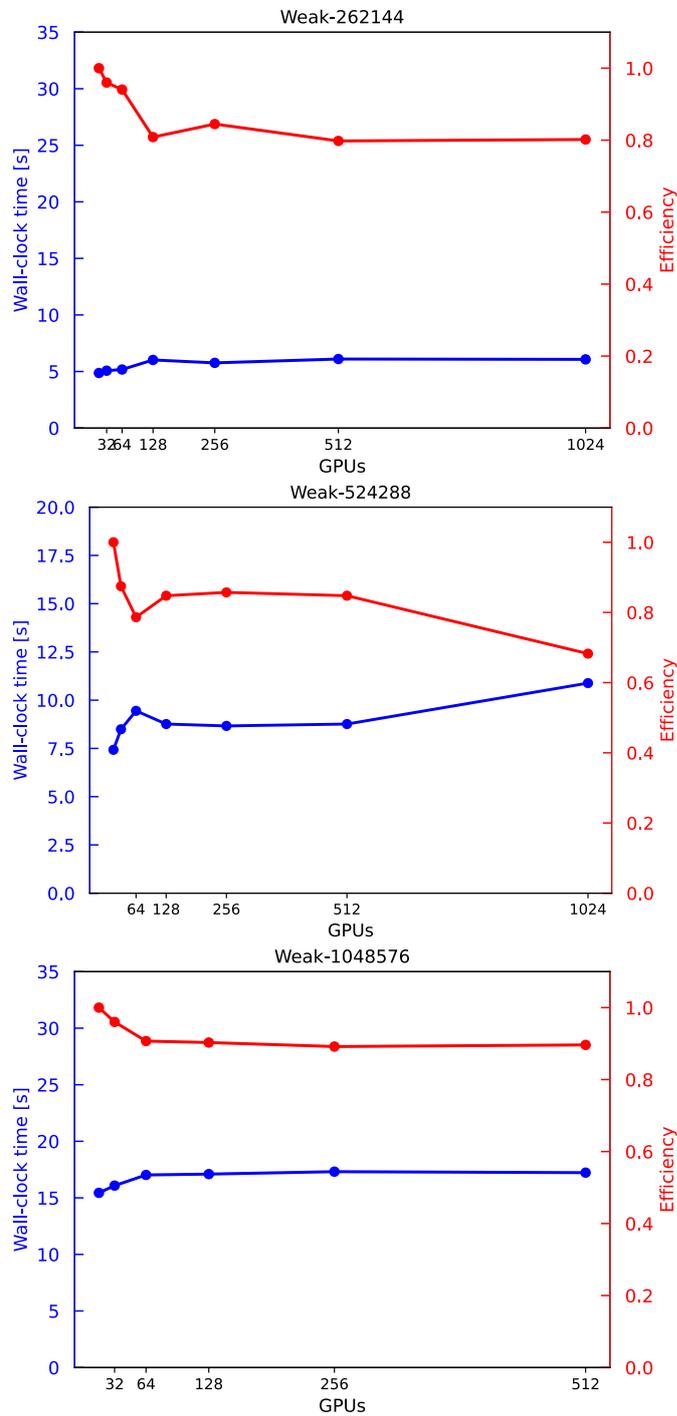


Figure 4.7: Weak scalability for the three cases.

Chapter 5

Results

In this section, the main results are presented. Even if the solver structure is 3D, most of the configurations considered are 2D to reduce the computational cost. The correct implementation of the third dimension has been verified for an hypersonic flow past a sphere, reported in appendix B. Grid convergence studies for cases in sections 5.2, 5.3 and 5.4 are all collected in appendix C. Lastly, note that the results obtained through the standalone version of the code are referred to as CHESS, whereas results obtained through the code coupled with MUTATION⁺⁺ are referred to as CHESS-M⁺⁺

5.1 Verification studies

Preliminary analysis have been carried out to verify the correct treatment of each aspect of hypersonic flows. They are presented in the next subsections. Specifications about the test cases are summarized in table 5.1. Note that all these preliminary analysis are carried out through the mT model.

Table 5.1: Test cases selected for verification studies.

Test	Assessment
0D reactor	Chemistry
1D diffusion with catalysis	Mass diffusion and GSI
1D shock tube	Shock capturing

5.1.1 0D reactor

Chemistry and vibrational relaxation have been verified by simulating a 0D test case. An initial 5 species neutral air mixture at fixed pressure is left to reach the equilibrium condition. Two test cases have been analyzed: the initial conditions are summarized in table 5.2. The first one accounts for thermal equilibrium, so

that CHES and CHES-M⁺⁺ can be both tested and compared to the theoretical values, obtained through the analytical solution implemented in MUTATION⁺⁺. The second case accounts also for thermal non-equilibrium. Since the standalone version of the code treats each thermal bath independently (3 different vibrational temperatures) only CHES-M⁺⁺ is tested as it involves one single thermal bath for the mixture. In such a case, a comparison with the well-known US3D [133–135] solver is provided.

Fig. 5.1 reports the relaxation in terms of translational temperature and species mass fractions for the CNE case. A satisfactory agreement is found. Black crosses indicate theoretical values obtained through the analytical solution in MUTATION⁺⁺, confirming the correct behavior of the chemical processes. Other methods to compute the equilibrium condition are available in the literature [136].

Fig. 5.2 shows the temporal evolution of the two temperatures and the species mass fractions for the TCNE case. The curves obtained through the coupling with MUTATION⁺⁺ are in perfect agreement with the reference from US3D.

Table 5.2: Initial conditions for 0D reactor test cases.

Case	T [K]	T_{vib} [K]	ρ [kg/m ³]	Y_{N_2}	Y_{O_2}
CNE	7000	7000	0.01	0.767	0.233
TCNE	15000	300	0.48	0.767	0.233

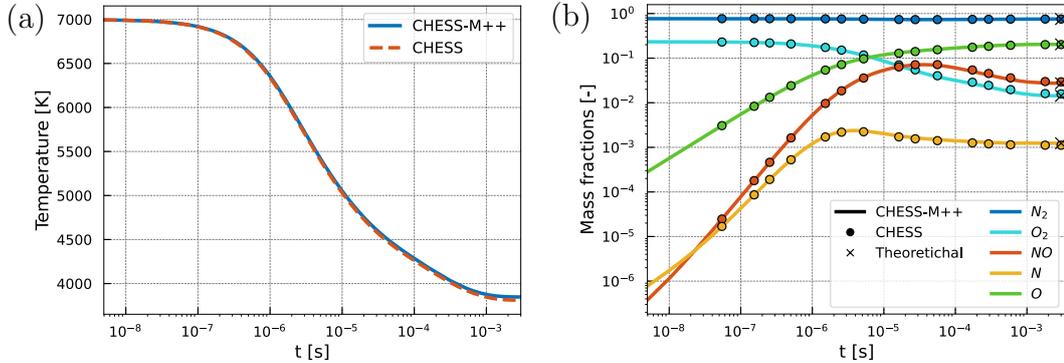


Figure 5.1: Temporal evolution of the translational temperature (a) and species mass fractions (b) for the CNE case.

5.1.2 1D diffusion problem with catalysis

Mass diffusion and gas-surface interaction are assessed through a 1D diffusion test case. A pure nitrogen mixture is injected from a reservoir into a 0.2 m long tube, at constant temperature (3000 K) and pressure (100 Pa). On the other side of the tube, the wall is isothermal ($T_w = 3000$ K) and catalytic. Four different

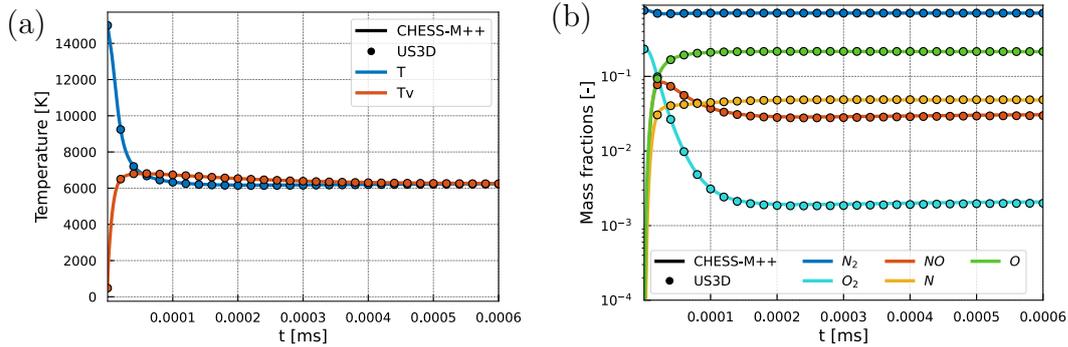


Figure 5.2: Temporal evolution of the translational and vibrational temperatures (a) and species mass fractions (b) for the TCNE case.

catalysis efficiencies are analyzed, as shown in Fig. 5.3. This test case is very interesting for two main reasons. First of all, the GSI is investigated in four different cases, starting from a quasi-inert wall ($\gamma = 0.001$) to a fully-catalytic wall ($\gamma = 1$). Secondly, the only phenomenon contributing to the heat flux is the mass diffusion at wall, as the wall temperature and the gas temperature are the same (i.e., conduction will not affect the heat flux). The mixture is considered frozen and the variation in terms of mass fractions will be promoted only by catalysis effects.

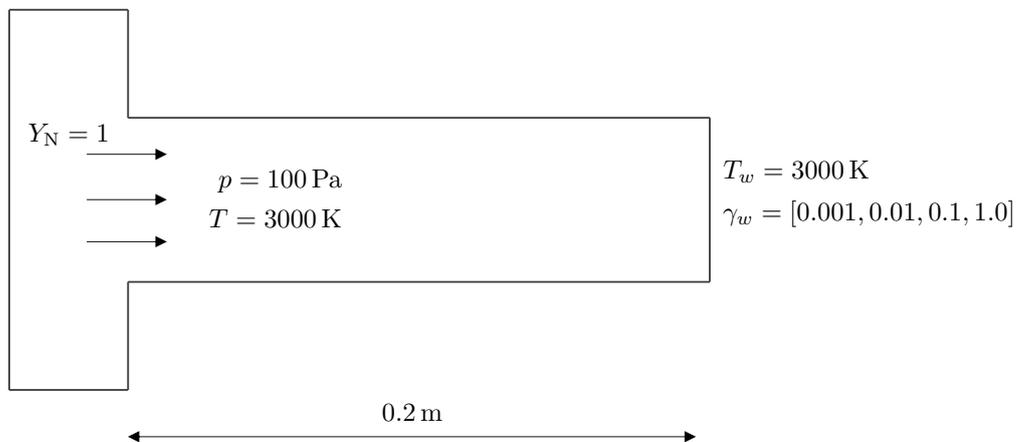


Figure 5.3: Sketch of the 1D diffusion problem with catalysis.

A binary mixture has been chosen for simplicity, since the analytical formulation of such a problem (reported in appendix A) can be easily obtained. Numerical results are compared with the analytical solutions. These simulations are performed through CHES5-M⁺⁺ to deal with partial catalysis. Nevertheless, as previously mentioned, CHES5 presents a standalone fully-catalytic boundary condition. For this reason, for the fully-catalytic case the simulation is performed through CHES5 also, in order to assess the correct implementation of the catalytic boundary condition. The results, illustrated in Fig. 5.4, present a perfect

agreement with the analytical solutions. Also the standalone boundary condition implemented in CHES for the fully-catalytic wall is coherent with references, with slight differences due to the boundary condition imposition.

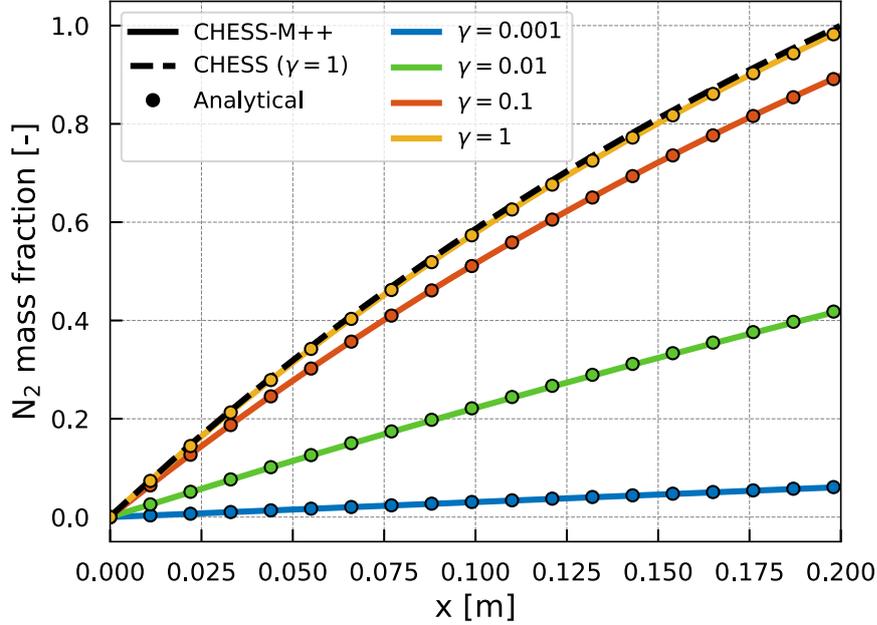


Figure 5.4: N_2 mass fraction profiles for the four catalysis efficiencies.

5.1.3 1D shock tube

The shock tube test case analyzed by Grossmann and Cinnella [137] is here reproduced. The tube is 1 meter long, discretized with 600 control volumes. The diaphragm separating the right and left state is placed in the middle of the tube. The initial conditions, under the assumption of equilibrium, are given in table 5.3. Thermal non-equilibrium is assumed. However, it is found to be negligible, so that both versions, CHES and CHES-M⁺⁺, can be employed for this test.

Table 5.3: Initial conditions for the shock tube test case.

u_L [m/s]	T_L [K]	p_L [Pa]	u_R [m/s]	T_R [K]	p_R [Pa]
0	9000	195256	0	300	10000

According to the reference [137], the results are analyzed at $99 \mu s$. An excellent agreement is found, with minor differences in the mass fractions profiles.

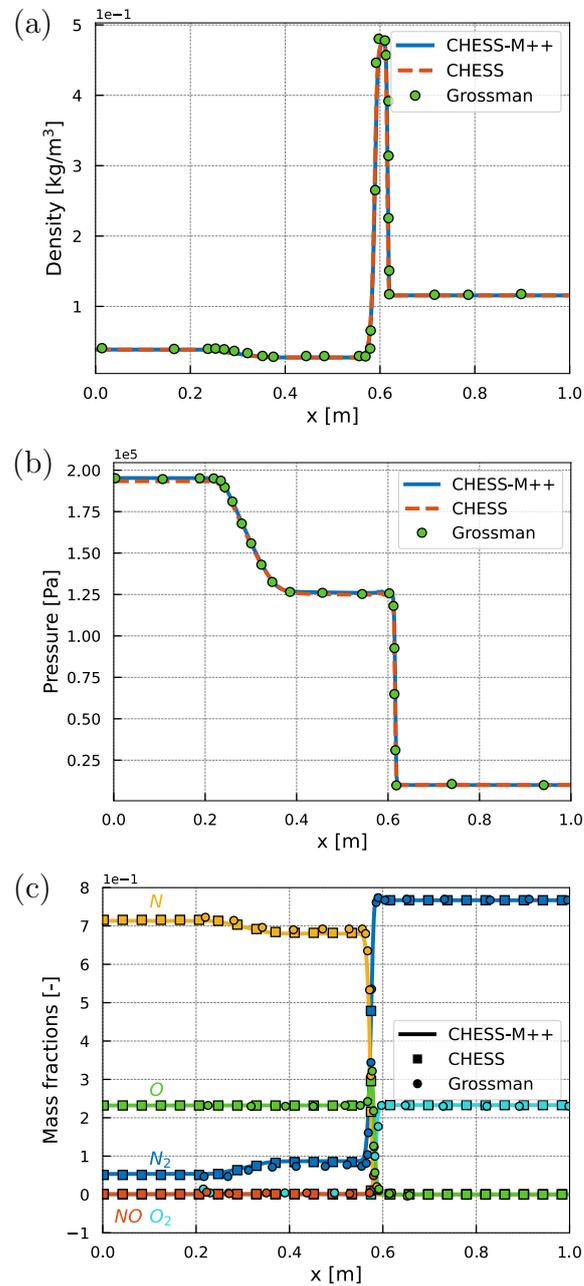


Figure 5.5: Results for the 1D shocktube: pressure (a), density (b) and species mass fraction (c) profiles.

5.2 Hypersonic shock wave/boundary layer interaction

In the hypersonics community, one of the major topics is the shock wave/boundary layer interaction (SWBLI). The correct prediction of the boundary layer separation extent is fundamental for the analysis of the wall quantities such as pressure and heat flux distributions. In the recent years, many authors devoted efforts to the analysis of hypersonic double-wedge [27–29, 138, 139] or double-cone [32, 140–151] flows, often supported by experimental campaigns [25, 26, 31, 152, 153]. These geometrical configurations represent simplified models of fuselage or wings, thus very attractive in aerospace engineering. The interest is increasing also in the framework of Mars entry flows, for which carbon mixture have been analyzed [154]. Moreover, three-dimensionality effects occur and affect the general flow field [25, 27, 28, 155].

In this section, results for hypersonic double-wedge and double-cone flows are presented. For each geometrical configuration, a low enthalpy and a high enthalpy regime have been considered to assess the relevance of non-equilibrium and unsteady behavior of the flow. A grid independence analysis, reported in appendix C, was carried out for both double-wedge and double-cone configurations at high enthalpy regime. Note that, only for these test cases, a 2D formulation of CHESH has been adopted, thus requiring an additional source term in the y – *momentum* equation to deal with axial-symmetry of the double-cone. This term reads $\dot{\omega}_{ax} = \frac{p - \sigma_{\theta,\theta}}{y}$, being p the pressure, $\sigma_{\theta,\theta}$ the tangential stress component and y the spatial coordinate in the radial direction.

The sketch of a generic flow field (either for double-wedge or double-cone configuration) is illustrated in Fig. 5.6. The deflection of the first wall induces the generation of an attached shock (AS). Given the higher deflection of the second wall, a detached shock (DS) propagates and interact with the AS, resulting in a transmitted shock (TS). The induced adverse pressure gradient is responsible for the boundary layer separation (separation region, SR) near the compression corner, leading to the formation of a separation shock (SS). Especially in case of the double-wedge, the TS plays a key role in inducing the unsteadiness when a low enthalpy regime is considered [23, 29]. On the other hand, due to the axial-symmetry of the body, the TS shock is too weak in the double-cone case and the flow is expected to be stationary. Finally, non-equilibrium affects the macroscopic flow features only at high enthalpy regime. For this reason, four different cases are analyzed, referred to as LEDW (low enthalpy double-wedge), LEDC (low enthalpy double-cone), HEDW (high enthalpy double-wedge) and HEDC (high enthalpy double-cone). They allow to assess the relevance on non-equilibrium for different mixtures, and the (potential) unsteady behavior of such flows. Table 5.4 reports the geometrical configurations, whereas the free stream conditions are listed in table 5.5. The free stream Reynolds number have been computed considering the

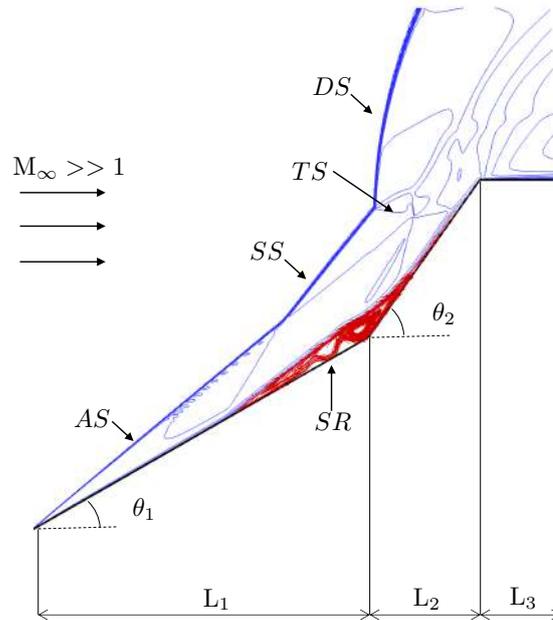


Figure 5.6: General flow field (blue lines) of the shock wave/boundary layer interaction: DS (detached shock), AS (attached shock), SS (separation shock), TS (transmitted shock), SR (separation region, red lines).

sum of the three axial contributions as reference length, namely $L = L_1 + L_2 + L_3$: as found in the experiments [26, 142], the flow can be considered laminar. Note that an air mixture and a binary oxygen mixture are considered for the double-wedge and the double-cone flows, respectively. Concerning the double-wedge, a nitrogen mixture were also considered, but the macroscopic characteristics were comparable to those reported for an air mixture. Hence, for seek of brevity, only results for an air mixture are illustrated in this thesis; the reader may refer to [23] for a complete description of the nitrogen mixture.

Table 5.4: Geometrical reference values: lengths are in meters, angles in degrees.

Quantity	Double-wedge	Double-cone
L_1	0.04399	0.0921
L_2	0.01458	0.0921
L_3	0.01082	0.0433
θ_1	30	25
θ_2	55	55

Before going into the details, it is worth specifying that all the temporal instants mentioned in the HEDW analysis indicate a relative time. Indeed, in the experimental setup [26] the flow presents an already developed shock structure at the initial time ($t = 0$ s), namely the time when the Schlieren pictures started being collected. Comparing the Schlieren image at $t = 0$ s with the numerical

Table 5.5: Flow free stream conditions for the selected test cases.

Quantity	LEDW	LEDC	HEDW	HEDC
h_0 [MJ/kg]	2.1	4.0	8.0	9.8
M_∞	7.1	10.4	7.1	8.1
Re_∞	7.38×10^4	7.88×10^4	2.94×10^4	2.91×10^4
p_∞ [Pa]	391	91	780	165
T_∞ [K]	191	190	710	625
ρ_∞ [kg/m ³]	0.007	0.0018	0.0038	0.00094
Y_{N_2}	0.78	0.0	0.78	0.0
Y_{O_2}	0.22	0.9986	0.22	0.9245
Y_O	0.0	0.0014	0.0	0.0755

solution, it was found that such a structure is obtained after 0.01 *ms* of simulation, with the whole flow field initialized with the free stream values. Hence, the values of the time in the following refer to the temporal instants after the initial shock structure. Concerning the LEDW flow, since the goal is the analysis of the unsteady behavior, the time scaling is absolute.

5.2.1 Low enthalpy double-wedge flow

When the free stream enthalpy is low enough, non-equilibrium phenomena can be neglected. It has been demonstrated by many researchers that chemical activity is too weak to induce molecular dissociation in both double-wedge [23, 29] and double-cone [32, 156] configurations. Hence, these flows are considered frozen, and only the mT model has been employed for their analysis. In both geometrical configurations, no-slip boundary condition is imposed at the wall, assumed non-catalytic and isothermal ($T_w = 300$ K).

The transient state of the LEDW flow is reported in Fig. 5.7, where Schlieren images are illustrated at different snapshots. They represent the contour of the density gradient magnitude (the legend is omitted for a better visualization). Specifically, up to $t = 2.4$ *ms* the flow tries to reach the steady state. Nevertheless, as soon as the TS interacts with the wall, the DS is pushed toward the expansion corner, giving rise to a periodic behavior. This is linked to the interaction of the TS with the wall. As the SS keeps moving, the DS follows the evolution of the SR due to the adverse pressure gradient in the boundary layer. This makes the TS in turn slide down the expansion corner, interacting with the wall and inducing the unsteadiness. For the purpose of capturing such a periodic behavior, four probe points have been distributed along the wall to analyze the temporal evolution of the heat flux. Their axial coordinates are $x_1 = 0.021993$ *m*, $x_2 = 0.03299$ *m*, $x_3 = 0.0503$ *m* and $x_4 = 0.056607$ *m*. Fig. 5.8 shows the trend of the heat flux in the four stations. As expected, most of the activity occurs near the expansion corner, whereas in the three probes placed near the compression corner

the heat transfer is quite weak due to the boundary layer separation. This result was already found by Durna and Celik [29] for a nitrogen mixture. The present investigation confirms the same behavior for an air mixture. Specifically, the flow is periodic with a period of approximately 2.7 ms .

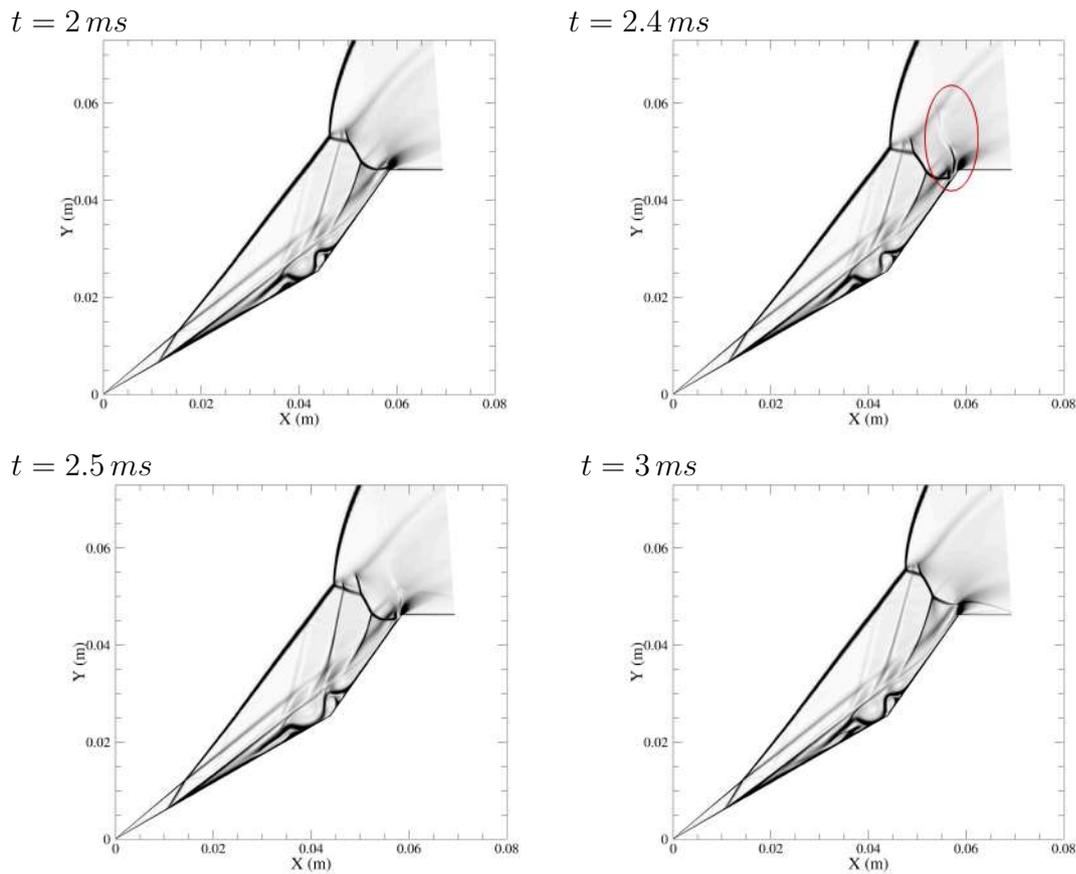


Figure 5.7: Schlieren images at different instants for the LEDW flow. Legend is omitted for a better view.

Given this unsteadiness, a time-step size independence analysis has been carried out to be sure of the present outcome. The main simulation was performed with $CFL = 0.2$. Following the definition given in Eq. (3.14), the resulting time-step size was of the order of $4 \times 10^{-10}\text{ s}$. Part of the simulation (up to $6\ \mu\text{s}$) was performed again imposing $CFL = 0.1$, thus corresponding to a time-step size of approximately $2 \times 10^{-10}\text{ s}$. The comparison of the heat fluxes obtained from the two simulations (Probe 4) are shown in Fig. 5.9: it emerges that a value of $CFL = 0.2$ provides a time-step independent solution and can be exploited to save computational time.

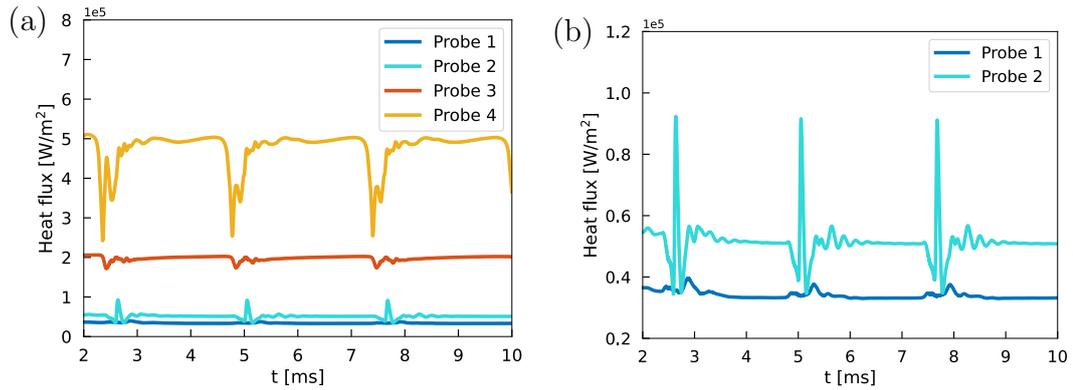


Figure 5.8: Temporal evolution of the heat flux in the probe points (a), zoom for probes 1 and 2 (b), for the LEDW flow.

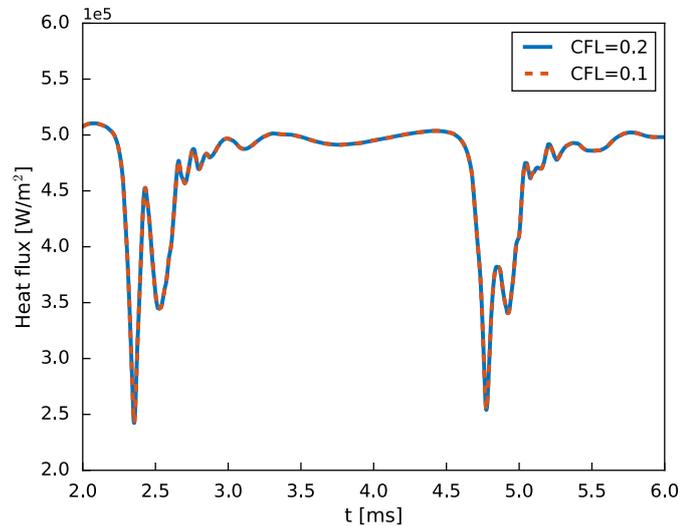


Figure 5.9: Temporal evolution of the heat flux in probe 4 for two different CFL values for the LEDW flow.

5.2.2 Low enthalpy double-cone flow

Since the flow around the double-cone is axial-symmetric, the shock waves propagating in the flow field are weaker than those forming in a double-wedge configuration and therefore a steady state is expected. In order to compare the present results, the work by Nompelis et al. [32] has been taken as reference, where the case considered is referred to as *Run-90*. The wall pressure and heat flux distributions are reported in Fig. 5.10. The separation extent is well predicted and an excellent agreement is found with both numerical results by Nompelis and experimental measurements. Also the peak of pressure and heat flux are well captured. It deserves to be pointed out that a little difference in the geometry led to a slight mismatch in terms of expansion zone ($x \approx 0.15 m$); nevertheless, it does not influence the upstream field as the flow is supersonic in that zone, as shown

in Fig. 5.11.

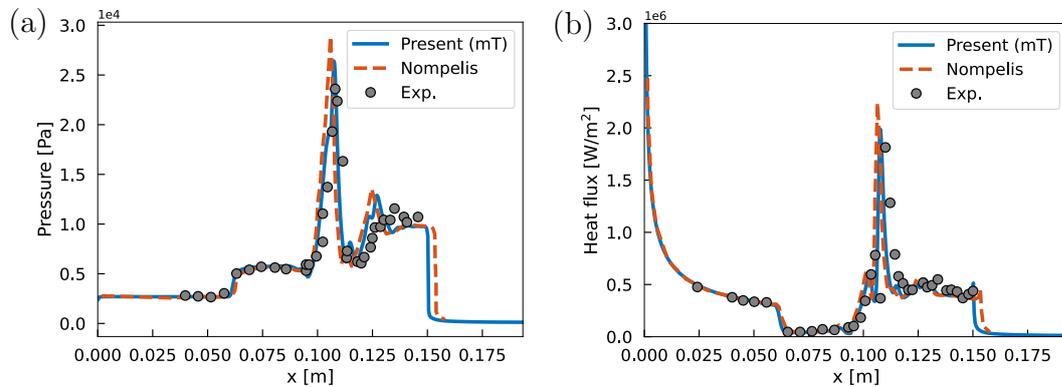


Figure 5.10: Wall pressure (a) and wall heat flux (b) distributions for the LEDC flow.

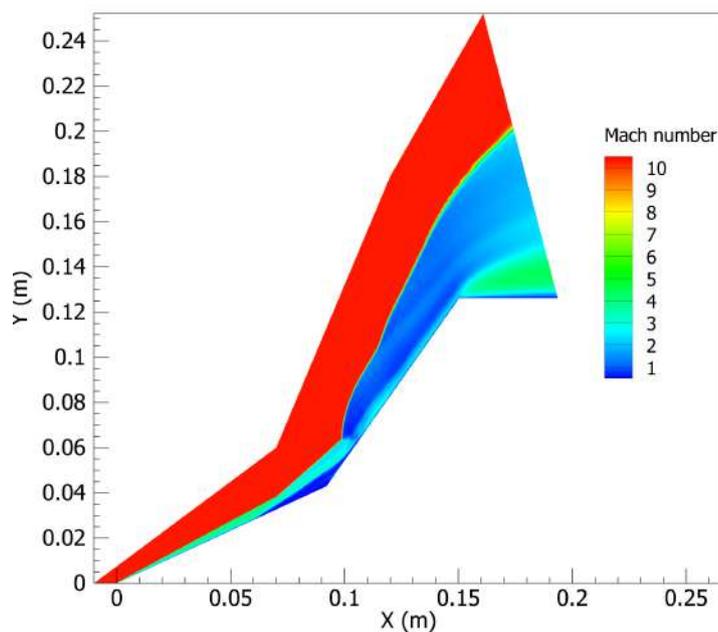


Figure 5.11: Contour of the Mach number for the LEDC flow.

5.2.3 High enthalpy double-wedge flow

Differently from the low enthalpy regimes, the high free stream values of the HEDW and HEDC flows induce thermochemical non-equilibrium. Chemistry is activated due to the high temperature reached downstream of the shock waves. In such a case, it is reasonable to employ the State-to-State model to analyze the actual

internal state distributions of molecular nitrogen and molecular oxygen. In the case of the double-wedge, reference results are taken from the work by Hao et al. [27].

The numerical set up is the same of the low enthalpy case: no-slip boundary condition is imposed at wall, considered isothermal ($T_w = 300$ K) and non-catalytic. Experimental campaigns conducted for this case suggest that the flow could still be unsteady [26]. Nevertheless, the time covered by the experiments was actually limited to less than 0.3 ms. In this time range, Schlieren pictures were collected to describe the shock structure evolving in front of the sample and it was found that at the last snapshot ($t = 0.242$ ms) the steady state was not reached yet. Next numerical investigations [25,27,28,155] assessed the influence of three-dimensional effects on the flow field. Despite the regularity of the geometry, waves propagate also in the span-wise direction (z axis). Such a behavior could hamper the flow in reaching the steady state.

Based on these considerations, the present analysis aims at addressing two points. First of all, the verification of the results in the transient state, by providing an accurate temporal evolution of the Schlieren contours. Also, wall pressure and heat flux distributions are provided and compared with numerical results available in the literature [27]. Such simulations are performed through both the mT and the StS model as non-equilibrium is relevant. Secondly, the steadiness of the flow is assessed by simulating a temporal range up to 10 ms. In such a case, given the high computational cost required to cover such a long period of time, only the mT model is employed.

Fig. 5.12 shows the wall pressure and heat flux profiles at $t = 242$ μ s, that is the experimental time. A comparison with experimental measurements is not suitable for this test case, as the aforementioned three-dimensional effects influence the numerical solution. Since the present formulation is 2D, only the comparison with numerical results available in literature is presented. As 3D calculations require a huge computational resources, these are perspectives for future works. A good agreement is found between the present mT model and numerical results reported by Hao et al. [27]. The StS model provides a larger separation region, due to the weaker dissociation (evident in Fig. 5.13) that leads to a smaller energy dissipation. This in turn promotes stronger shock waves, as it emerges from the wall pressure profile in Fig. 5.15, where a higher pressure peak is observed for the StS model. Schlieren images in Fig. 5.14 confirm this assessment.

In order to better understand this aspect and justify the difference in terms of separation extent, the analysis of the internal state distributions through the StS model is crucial. Such an analysis is carried out on O_2 since nitrogen dissociation is weak. For this purpose, six probe points have been selected in the flow field. Their coordinates are reported in table 5.6 and in Fig. 5.15.

Population profiles illustrated in Fig. 5.16(a) highlight that in probes 3, 4 and 5 the assumption of a Boltzmann distribution is reasonable, except for the most energetic levels as they are more involved in chemical dissociation. Indeed, the

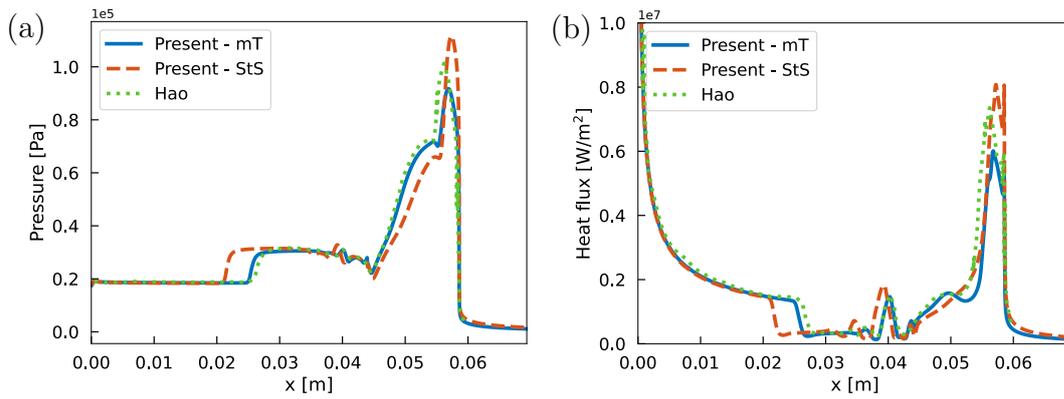


Figure 5.12: Wall pressure (a) and wall heat flux (b) distributions for the HEDW flow at $242 \mu\text{s}$.

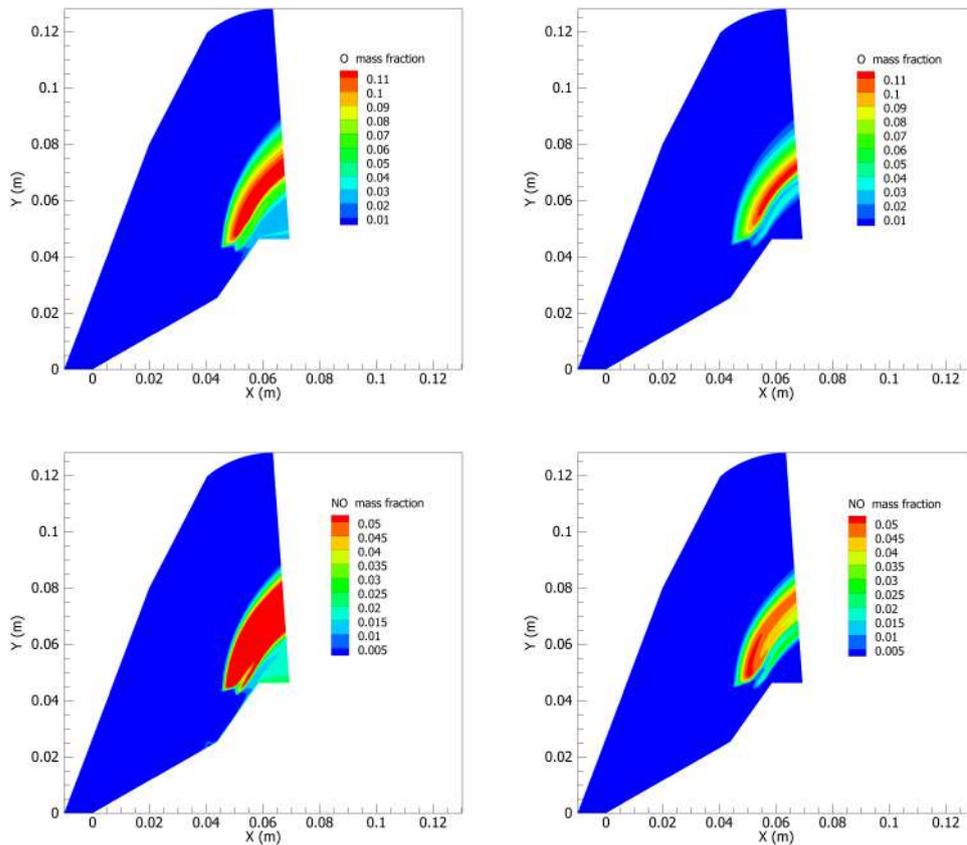


Figure 5.13: Contours of the mass fractions of O and NO for the HEDW flow: mT (left) and StS (right).

actual to Boltzmann type ratio reported in Fig. 5.16(b) is almost around 1 for all the levels, whereas the highest levels are slightly underpopulated as expected [87, 88]. On the contrary, distributions in probes 1, 2 and 6 strongly depart from

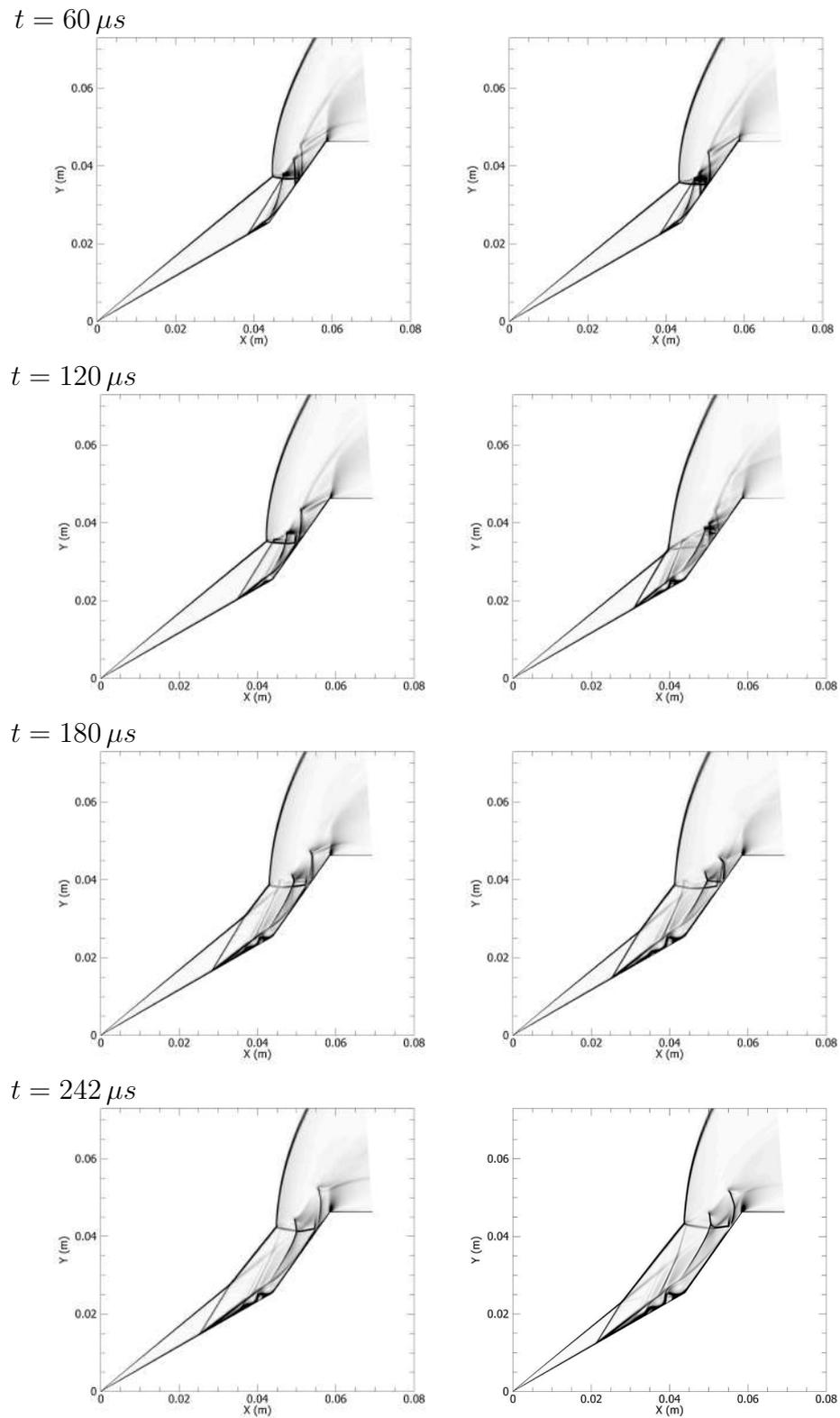


Figure 5.14: Schlieren images at different instants for the HEDW flow: mT (left) and StS (right). Legend is omitted for a better view.

Table 5.6: Spatial coordinates for the probe points.

Probe	x [m]	y [m]
1	0.0215	0.0126
2	0.0275	0.0228
3	0.0437	0.0433
4	0.058	0.06
5	0.05	0.05
6	0.0273	0.016

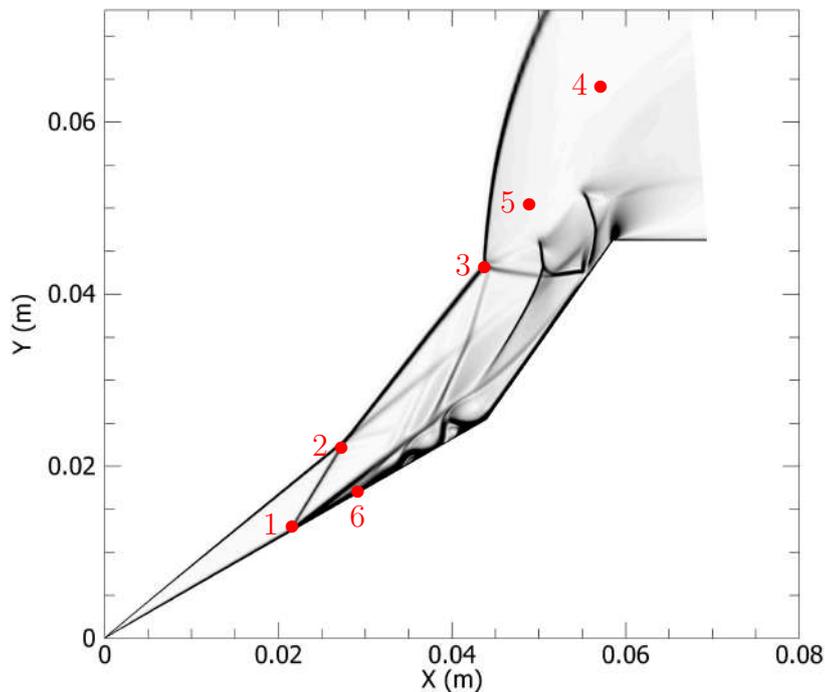


Figure 5.15: Probe points to analyze internal state distributions for the HEDW flow.

the Boltzmann one: the recombination occurring near the wall due to the lower temperature leads to an overpopulation of the highest levels.

Finally, the stationarity of the flow is assessed. Given the long temporal range covered, this aspect is investigated through the mT model only. The results for the same probe points of the LEDW case are shown in Fig. 5.17. After the transient state, the curves tend to an asymptotic value, confirming that the flow reaches a steady state on the long period.

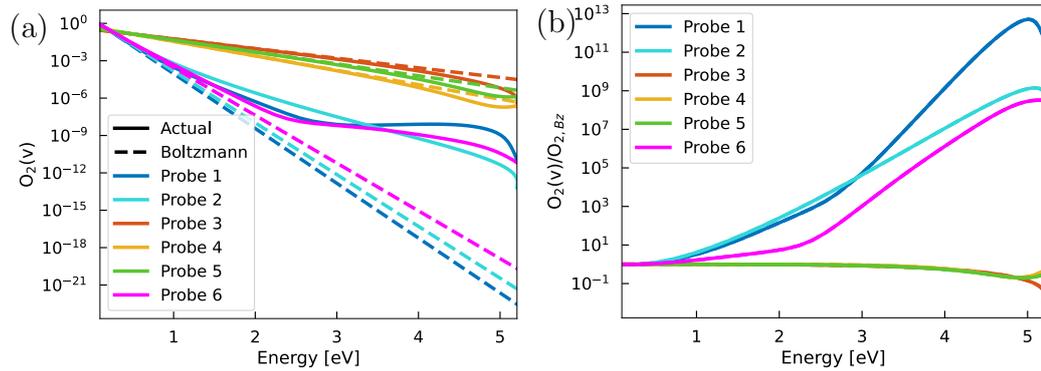


Figure 5.16: Distributions in the probe points (a) and actual to Boltzmann type ratio (b) for the HEDW flow.

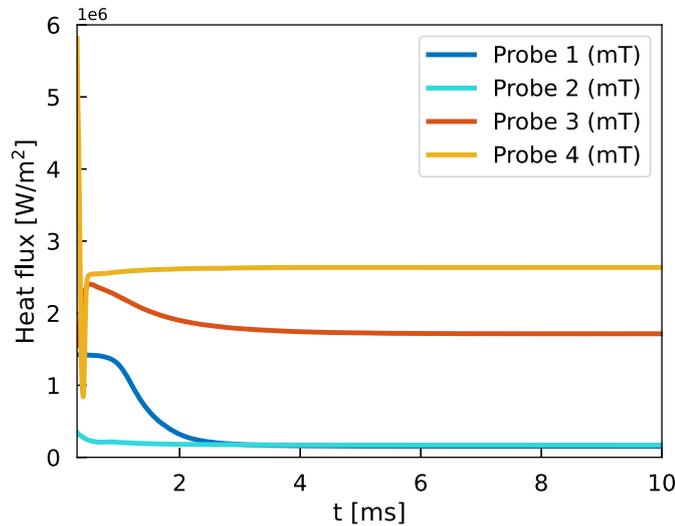


Figure 5.17: Temporal evolution of the heat flux in the probe points for the HEDW flow.

5.2.4 High enthalpy double-cone flow

As for the low enthalpy regime, the HEDC flow is expected to be stationary. Differently from the HEDW flow, the transient state does not take long time and the stationarity of the flow has been found experimentally [152]. Numerical results by Nompelis et al. [32], whose work reports also experimental measurements, are taken as reference for the verification/validation. The case under consideration is referred to as *Run-87*. As for the HEDW flow, mT and StS model are compared. Fig. 5.19 shows the contour the the molecular oxygen mass fraction. As for the HEDW flow, the StS model predicts a weaker dissociation. The distributions of wall pressure and heat flux obtained are reported in Fig. 5.18, along with the comparison with numerical results obtained by Nompelis [32] and experimental measurements. A very good agreement is found with numerical results from the

literature. Also, the mT and the StS models predict a comparable behavior of the solution, except for the peaks ($x \approx 0.1 m$). Nevertheless, numerical findings strongly depart from experimental measurements, especially in terms of separation extent. Indeed, contrarily to the low enthalpy condition, it is under predicted by the numerical solver. Different researches have been devoted to the investigation of the boundary layer separation for double-cone flows, found to be strongly sensitive to the Reynolds number [157, 158] and to the wall temperature [159], but not to kinetics [160]. Indeed, it is relevant only in the detached shock region (second wall).

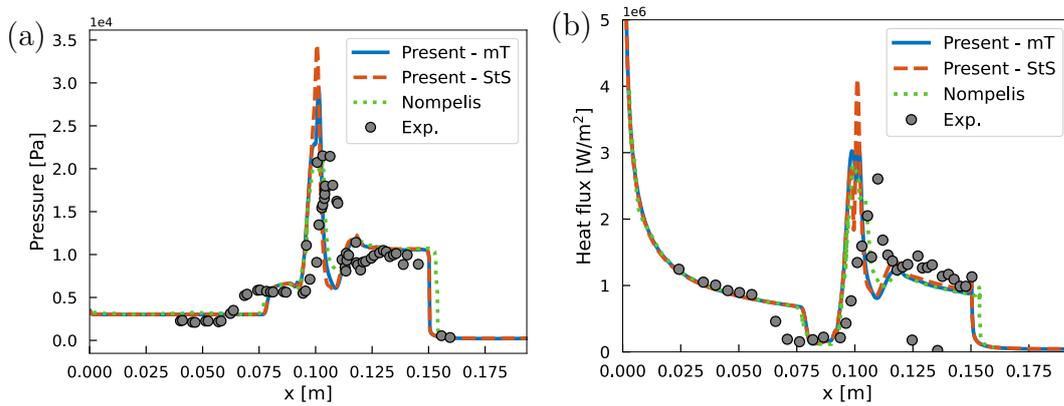


Figure 5.18: Wall pressure (a) and wall heat flux (b) distributions for the HEDC flow.

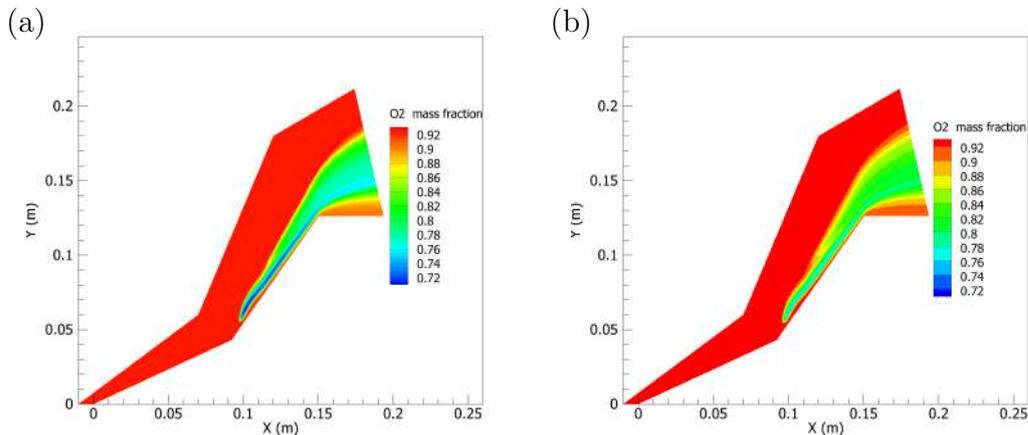


Figure 5.19: Contour of the mass fraction of O_2 for the HEDC flow: mT (a) and StS (b).

In order to assess the influence of wall kinetics, a fully-catalytic case is also analyzed. Obviously, the wall pressure remains unchanged, but the heat flux increases due to species diffusion at wall, as shown in Fig. 5.20. Again, numerical solutions present a good agreement, but still missing the experimental measurements. Specifically, the StS provides an even smaller separation region, as the

attached shock moves toward the compression corner due to the recombination. Furthermore, the overall heat flux is over predicted, so that it can be assessed that catalysis does not play a key role in the under estimation of the separation extent [140, 145, 156, 161].

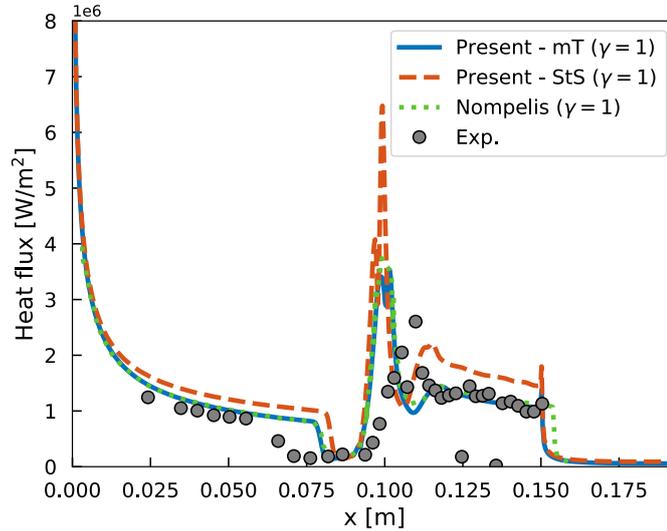


Figure 5.20: Wall heat flux profile for the HEDC flow with fully-catalytic wall.

By the way, a strange behavior of the wall pressure is observed in Fig. 5.18. In the attached shock region ($x \approx 0.05$ m) the flow is chemically frozen as also confirmed in Fig. 5.19 and it is not interacting with other shocks nor with the boundary layer. For these reasons, the calculation of the pressure downstream of the attached shock should be straightforward. Despite this, a mismatch with the experimental measurements is observed, which induces to think that the numerical free stream conditions summarized in table 5.5 could be different from the real ones [156]. They were computed by simulating an expansion nozzle flow [144] and employing a mT model. Their correct prediction is fundamental to well describe the shock wave/boundary layer interaction near the body [162, 163]. Dealing with an oxygen mixture, chemistry is activated at a lower temperature and a mT approach might be not suitable to describe the non-equilibrium [152]. On the other hand, a StS approach represents a much more accurate alternative to predict the flow conditions at the outlet of the nozzle, namely the free stream conditions for the double-cone flow, in the presence of strong non-equilibrium phenomena. Previous work have demonstrated the importance of using a StS approach when dealing with nozzle flows [22, 34, 35, 88]. For this reason, the present analysis has been extended to the analysis of the nozzle expansion, to compare the free stream conditions obtained employing the mT and the StS model. The results are presented in the following.

The nozzle employed in the experimental setup is the LENS I facility developed in the Calspan-University at Buffalo Research Center (CUBRC), able to ensure

outlet velocity regimes with Mach number up to 18 [142]. The difficulty in finding a precise technical documentation about the geometry and stagnation conditions led to investigate the expansion through an alternative nozzle, governed by the same stagnation enthalpy and keeping the overall values of the free stream quantities comparable to those of the *Run-87*. The geometry is axis-symmetric: a planar view is illustrated in Fig. 5.21 with geometrical values summarized in table 5.7. Stagnation pressure and enthalpy are 5 MPa and 9.85 MJ/kg respectively. No-slip and adiabatic wall boundary conditions are imposed.

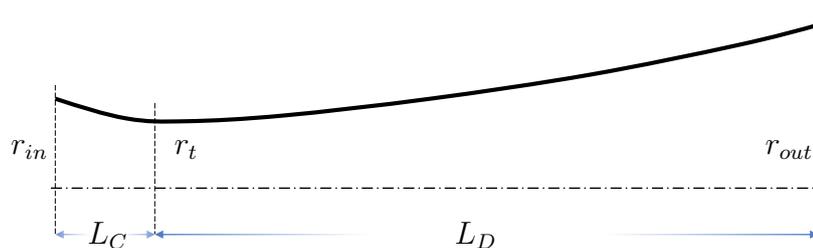


Figure 5.21: Geometry of the nozzle considered (not in scale).

Table 5.7: Reference values for the nozzle geometry: lengths in meters.

L_C	L_D	r_{in}	r_t	r_{out}
0.025	0.5	0.00175	0.016	0.061

The results in terms of temperature and molecular oxygen mass fraction profiles are shown in Fig. 5.22. Even if the temperature at the exit section is approximately the same, the recombination of atomic oxygen during the expansion is stronger in case of the mT model. Moreover, Fig. 5.23 shows the populations of the internal states: the actual distribution departs from the Boltzmann type. The complete set of exit conditions is summarized in table 5.8.

For the aforementioned reasons, several double-cone simulations have been carried out, according to the different free stream conditions and to the model employed. The first simulation involves the mT model (*Sim. 1*), whereas for the StS model two different distribution types are considered for the free stream, namely the non-Boltzmann (*Sim. 2*) and the Boltzmann one (*Sim. 3*). In the latter case, the Boltzmann distribution is evaluated at the free stream vibrational temperature. Finally, the two thermochemical models are combined (*Sim. 4* and *Sim. 5*). The set of simulations is summarized in table 5.9

The results for *Sim. 1*, *Sim. 2* and *Sim. 3* are reported in Fig. 5.24. Firstly, the non-Boltzmann distribution does not influence the shock structure. The separation extent predicted in *Sim. 2* is perfectly overlap to that of *Sim. 3*. The explanation of such a result lies into the internal state populations. Fig. 5.25 illustrates the probe points where the internal state distributions have been analyzed. Their profiles are shown in Fig. 5.26. The probes are located along the

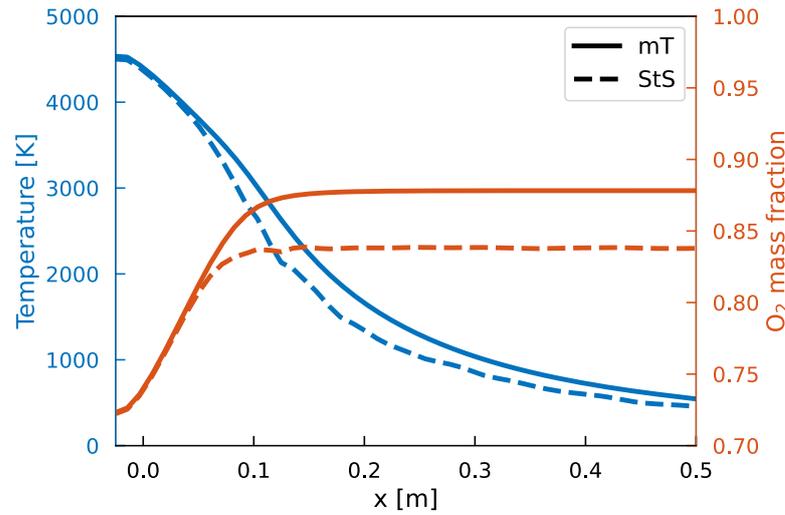


Figure 5.22: Profiles of temperature and molecular oxygen mass fraction along the center-line of the nozzle.

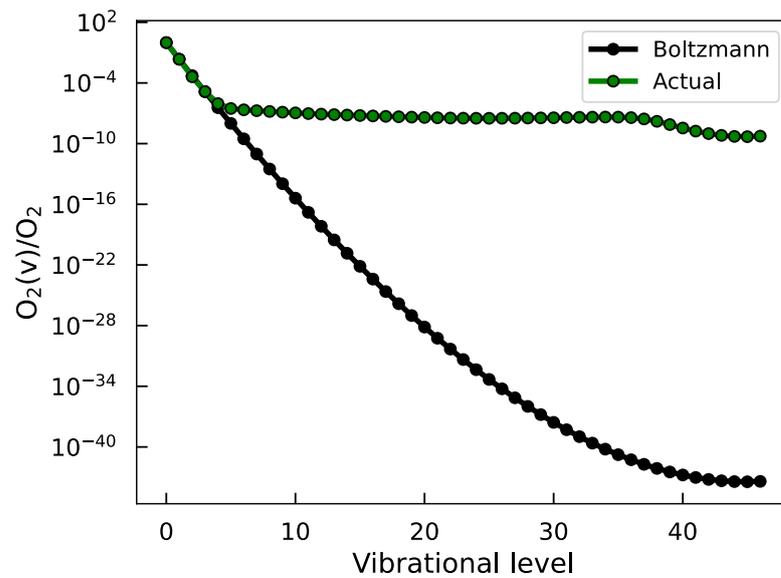


Figure 5.23: Internal state population at the outlet section of the nozzle.

first wall, downstream of the attached shock, justifying the non-Boltzmann trend of both simulations. The sudden variation of the fluid dynamics conditions pushes the populations to follow the same kind of trend, resulting in the same shock structure evolution. However, a relevant difference is observed with respect to the behavior described by the mT approach in *Sim. 1* (see Fig. 5.24). The different free stream conditions calculated from at the outlet section of the nozzle lead to a different evolution of the shock wave/boundary layer interaction on the double-cone. This can be the key in the view of understanding the mismatch reported for the previous high enthalpy case (*Run-87*). Fig. 5.27 illustrates the shock struc-

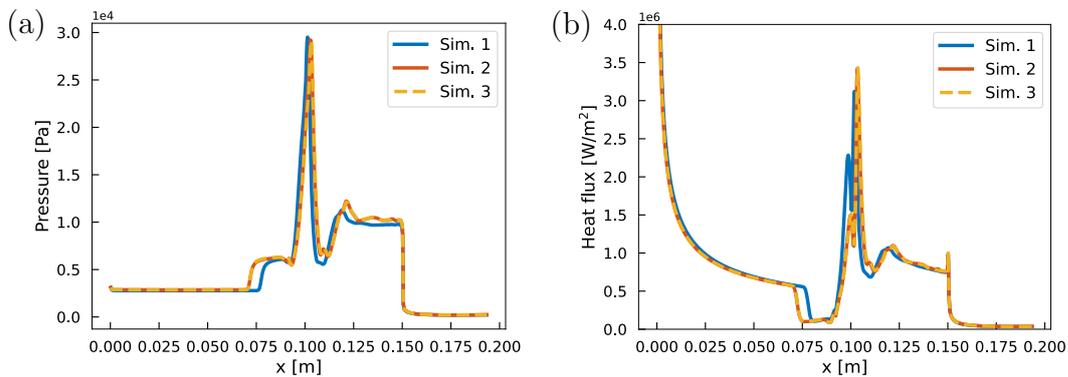
Table 5.8: Free stream conditions computed through the nozzle simulations ($h_0 = 9.85 \text{ MJ/kg}$).

Quantity	mT	StS
Mach number	8.09	8.37
u_∞ [m/s]	3845	3729
p_∞ [Pa]	148	142
ρ_∞ [kg/m ³]	0.000942	0.00104
T_∞ [K]	540	453
T_{v,∞,O_2} [K]	1075	593
Y_{O_2}	0.878	0.838
Y_{O}	0.122	0.162

Table 5.9: Simulations performed combining the two models.

	Nozzle	Double-cone
<i>Sim. 1</i>	mT	mT
<i>Sim. 2</i>	StS	StS (non-Boltzmann)
<i>Sim. 3</i>	StS	StS (Boltzmann)
<i>Sim. 4</i>	StS	mT
<i>Sim. 5</i>	mT	StS

tures of *Sim. 1* and *Sim. 2*, with a detailed detection of the different shocks in the flow field. The StS model (green lines) predicts a stronger bow shock promoted by a weaker dissociation and, as a consequence, the adverse pressure gradient is stronger than that provided by the mT model (red lines). This generates a larger 'separation bubble' and, of course, a stronger separation shock, moving the triple point away from the wall.

Figure 5.24: Pressure (a) and heat flux (b) distributions at wall for *Sim. 1*, *Sim. 2* and *Sim. 3*.

To conclude this analysis, a comparison of the heat flux obtained from *Sim. 1*,

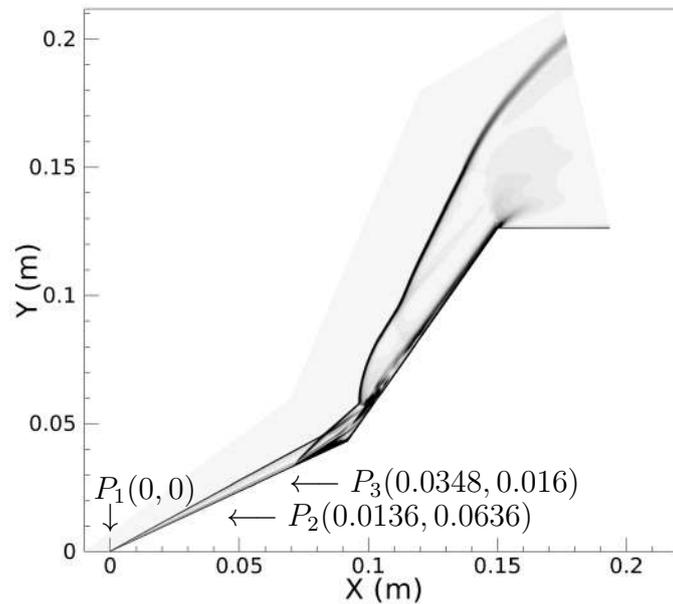


Figure 5.25: Schlieren image of *Sim. 2* and probe points (axial distances not in scale).

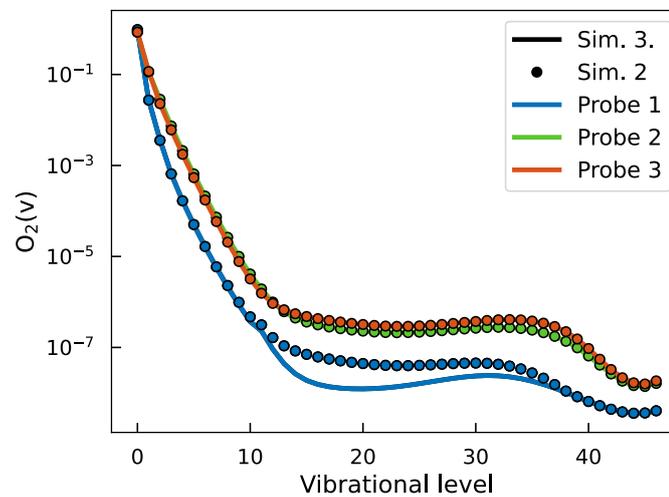


Figure 5.26: Oxygen populations among the vibrational levels in the probe points.

Sim. 2, *Sim. 4* and *Sim. 5* is shown in Fig. 5.28. The curve obtained from *Sim. 4* presents a smaller separation region. The mT model provides a stronger chemical activity than the StS model, as shown in Fig. 5.29. Hence, the total energy of the flow is mostly dissipated because of dissociation, thus explaining the smaller separation extent.

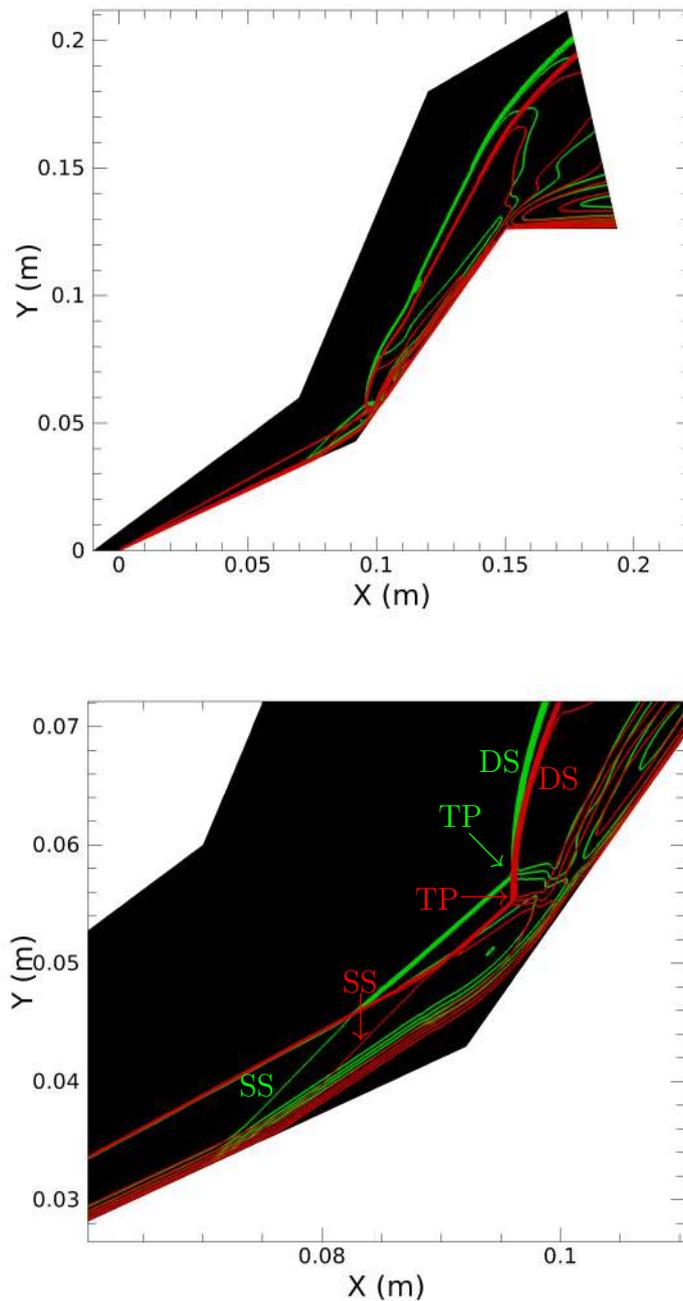


Figure 5.27: Shock structure (top) of *Sim. 1* (—) and *Sim. 2* (—). Zoom in the compression corner (bottom).

5.3 Reactive hypersonic flow past a cylinder

Space vehicles often present a bluff shape that promotes the propagation of a strong bow shock in front of the nose. Geometries as cylinders or spheres represent simplified models of the actual space bodies and are widely used in hypersonic

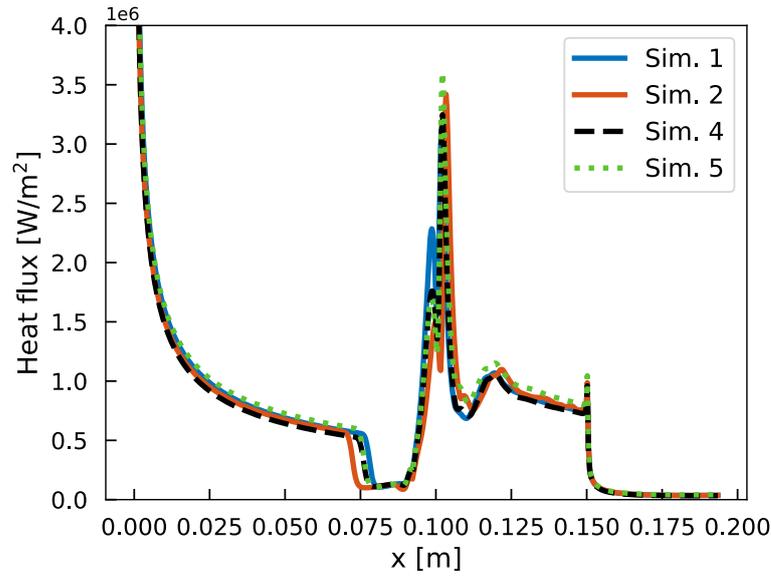


Figure 5.28: Heat flux profiles for *Sim. 1*, *Sim. 2*, *Sim. 4* and *Sim. 5*.

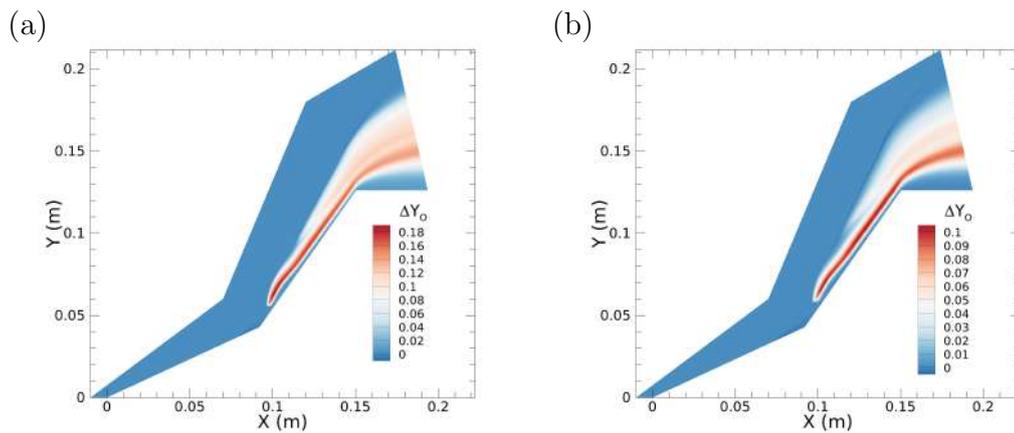


Figure 5.29: Contours of $\Delta Y_O = Y_O - Y_{O,\infty}$: *Sim. 1* (a) and *Sim. 2* (b).

community for numerical simulations. In recent years, particular efforts have been devoted to a specific test case concerning the flow past a cylinder [30], that will be referred to as *Knight case*. Many researchers compared numerical results for code-to-code verification and validation with experimental measurements of the wall pressure and heat flux distributions.

The sample considered is a cylinder with radius $r = 0.045 \text{ m}$, invested by an air flow in thermal equilibrium whose free stream conditions are reported in table 5.10. The free stream Reynolds number is small enough to assume a laminar flow [164]. The wall is assumed isothermal ($T_w = 300 \text{ K}$). However, two different simulations are performed: in the first one, the wall is inert, whereas in the second one it is considered fully-catalytic. Comparison with the well-known US3D solver and with

experimental measurements are provided. Moreover, note that this test case is performed with both CHES and CHES-M⁺⁺. Finally, it is noteworthy that the resolution employed in US3D is the same of the present analysis (grid convergence analysis is shown in appendix C).

Table 5.10: Free stream conditions for the cylinder test case.

M_∞	Re_∞	T_∞ [K]	p_∞ [Pa]	Y_{N_2}	Y_{O_2}	Y_{NO}	Y_N	Y_O
8.98	9820	901	476	0.7543	0.00713	0.01026	6.5×10^{-7}	0.2283

The normalized shock stand-off distance is well predicted: Fig. 5.30 illustrates the temperature profiles along the stagnation streamline. An excellent agreement is found for all the three codes, except for the temperature peak, where the temperature is slightly under estimated in US3D. Of course, in the case of fully-catalytic wall the shock stand-off distance remains approximately unchanged, but the mass fractions are influenced by the wall kinetics. Their profile are shown in Fig. 5.31 for both the inert and the fully-catalytic wall cases. Again, the three solver show a very good agreement, predicting the same behavior of the chemical activity.

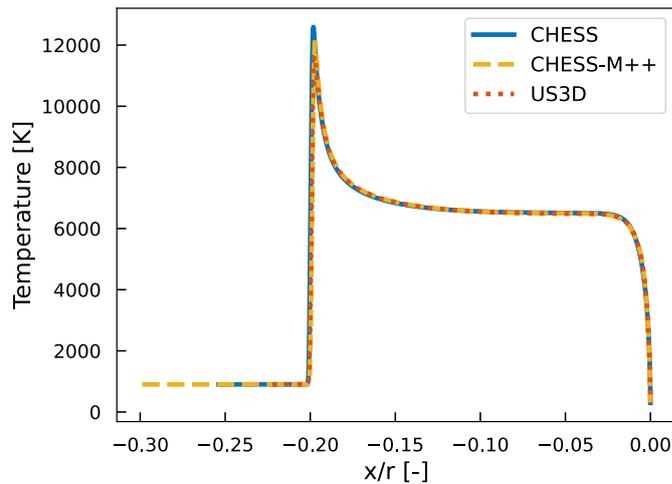


Figure 5.30: Temperature profiles along the stagnation streamline.

The most critical issue of this test case is the calculation of the wall quantities, in particular for the heat flux. The sensitivity of the solution to the grid resolution makes this task very hard [30]. Indeed, the wall resolution employed for these simulations ($5 \times 10^{-8} m$) is very challenging, especially for an explicit time integration scheme. However, the computed wall quantities are in a very good agreement with experimental measurements. The pressure distribution is not affected by the choice of the solver and perfectly match the experiments, as illustrated in Fig. 5.32(a). The calculation of the heat flux is more delicate: Fig. 5.32(b) and Fig. 5.32(c) report the results for the inert and the fully-catalytic cases. One can notice that

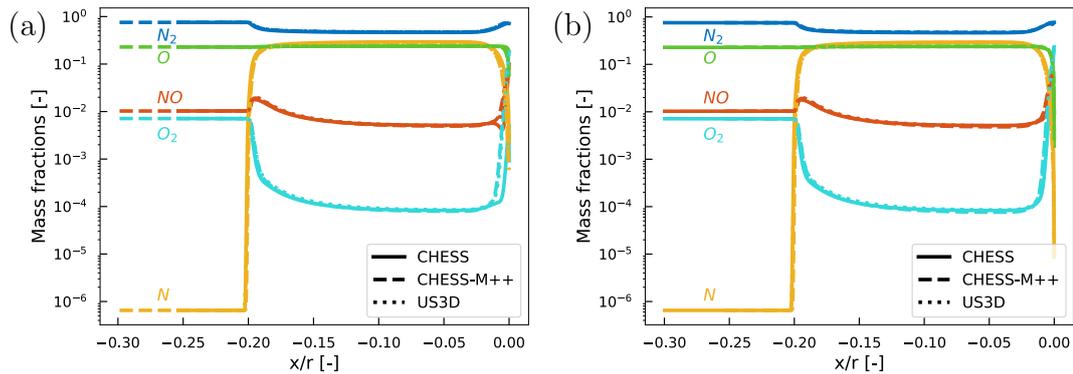


Figure 5.31: Mass fraction profiles along the stagnation streamline: inert wall (a) and fully-catalytic wall (b).

the wall recombination remarkably affects the heat transfer, especially in the stagnation point. The overall agreement is good, except for CHES, which provides an important under estimation. On the other hand, CHES-M⁺⁺ matches very well the experimental findings, confirming the higher accuracy of the Chapman-Enskog formulation in the presence of strong temperature gradients.

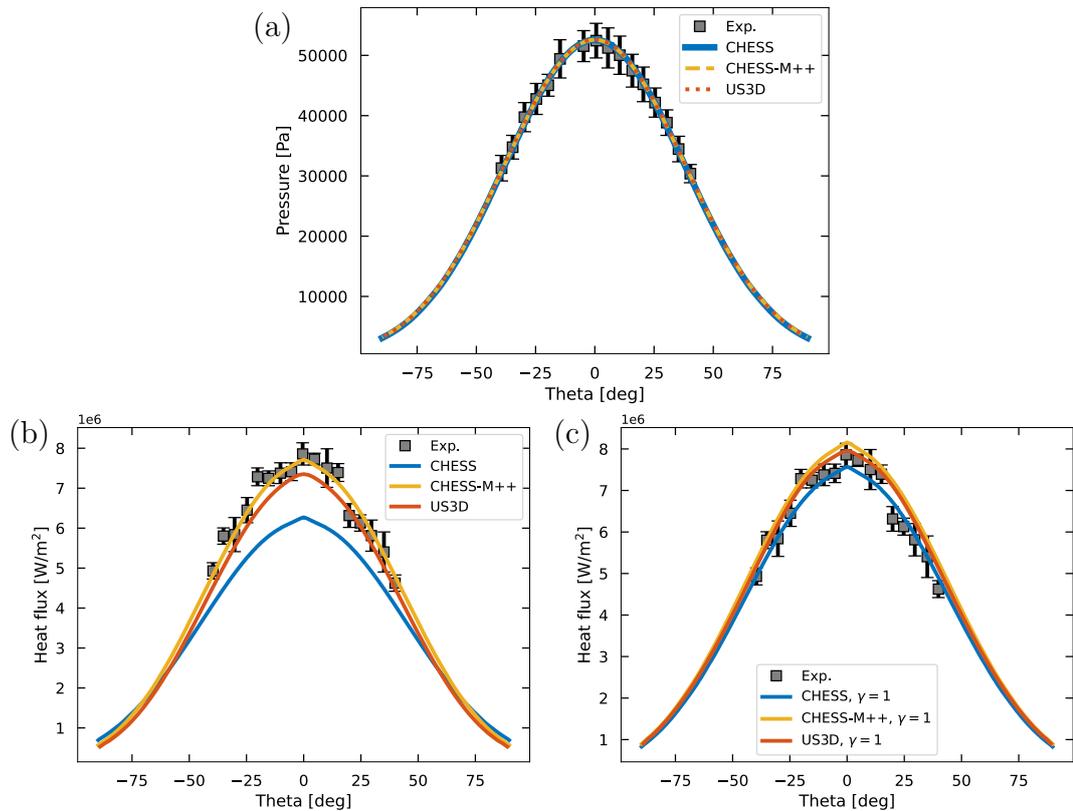


Figure 5.32: Wall pressure (a) and heat flux distributions over the wall: inert wall (b) and fully-catalytic wall (c).

5.4 Effects of ablation

This test case is a simplification of an experimental campaign conducted in the Plasmatron facility at the von Karman Institute [107] to determine the mass-blowing rate for a nitrogen subsonic flow around an axis-symmetric hemispherical sample ($r = 0.025\text{ m}$). Experimental data were then used to verify the coupling of the well-known US3D solver with MUTATION⁺⁺ GSI module [165]. Numerical results from the simulations were also compared to a 1D solver developed at the von Karman Institute, reporting a very good agreement.

The flow conditions selected are referred to as *G7* in [107]. Nevertheless, since the goal of the present test case is a code-to-code verification with the US3D solver, two main simplifications were adopted to reduce the computational cost [166]: electrons are neglected in the mixture and the geometry of the sample is planar. Free stream conditions are reported in table 5.11. The wall is considered isothermal and non catalytic. Only nitridation boundary condition is imposed, following the reaction illustrated in section 2.5.3. The nitridation efficiency follows the formulation found by Helber et al. [107], shown in Eq. (2.56).

Table 5.11: Free stream conditions for the ablative test case.

u_∞ [m/s]	Re_∞	T_∞ [K]	T_w [K]	p_∞ [Pa]	Y_{N_2}	Y_N
1570	36.7	10280	2407	1500	9.77659e-05	0.9999022341

The computational domain considered extends for $25r$ along both directions from the wall, as shown in Fig. 5.33, where boundary condition specification is also reported: the blue line indicates the inflow and the green line indicates the outflow. Along the stagnation line (black line) symmetry boundary condition is imposed, whereas the red line identifies the body sample, where no-slip condition is imposed.

Contours of the mass fractions of the major species, pressure and Mach number are shown in Fig. 5.34. Given the low wall temperature, atomic nitrogen particles tend to recombine near the wall, where Y_N reaches the lowest value. Nevertheless, the molecular nitrogen mass fraction remains poor. This is the consequence of the gas-surface interaction: particles of C are released into the gas phase, some reacting with N to form CN. As expected, the pressure is maximum in the stagnation point and decreases along the wall due to the acceleration of the gas, as confirmed the Mach number contour.

For the purpose of verification, profiles of species mass fractions, pressure and velocity have been extracted along the stagnation line and compared to those obtained through US3D. Moreover, the massblowing rate due to ablation is computed and compared between the two solvers. Fig. 5.35 shows the results, reporting a satisfactory agreement for all the analyzed quantities.

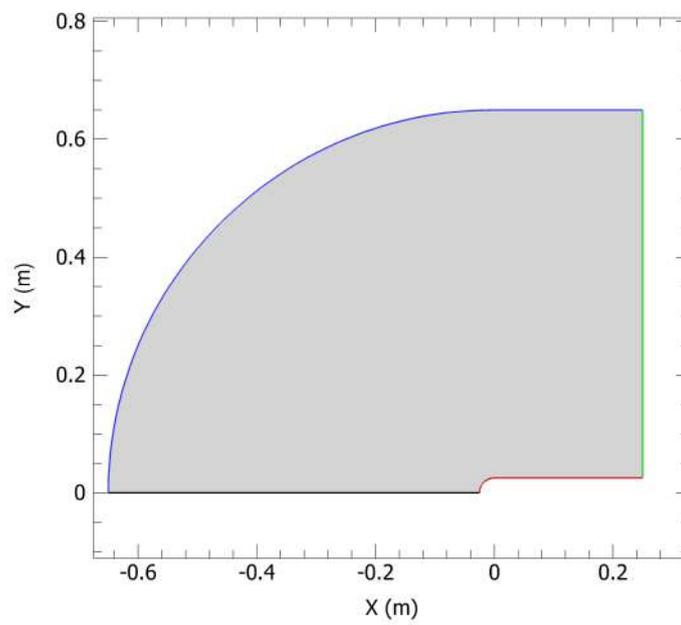


Figure 5.33: Computational domain for the subsonic ablative test case.

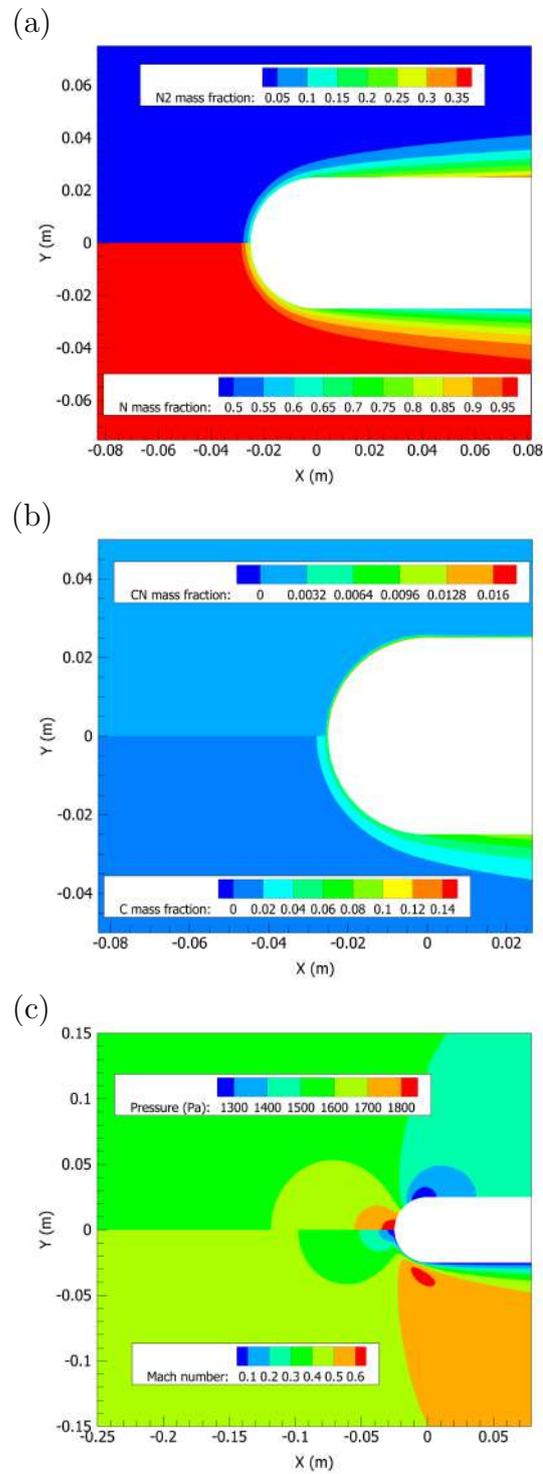


Figure 5.34: Contours of the mass fractions (a-b), pressure and Mach number (c).

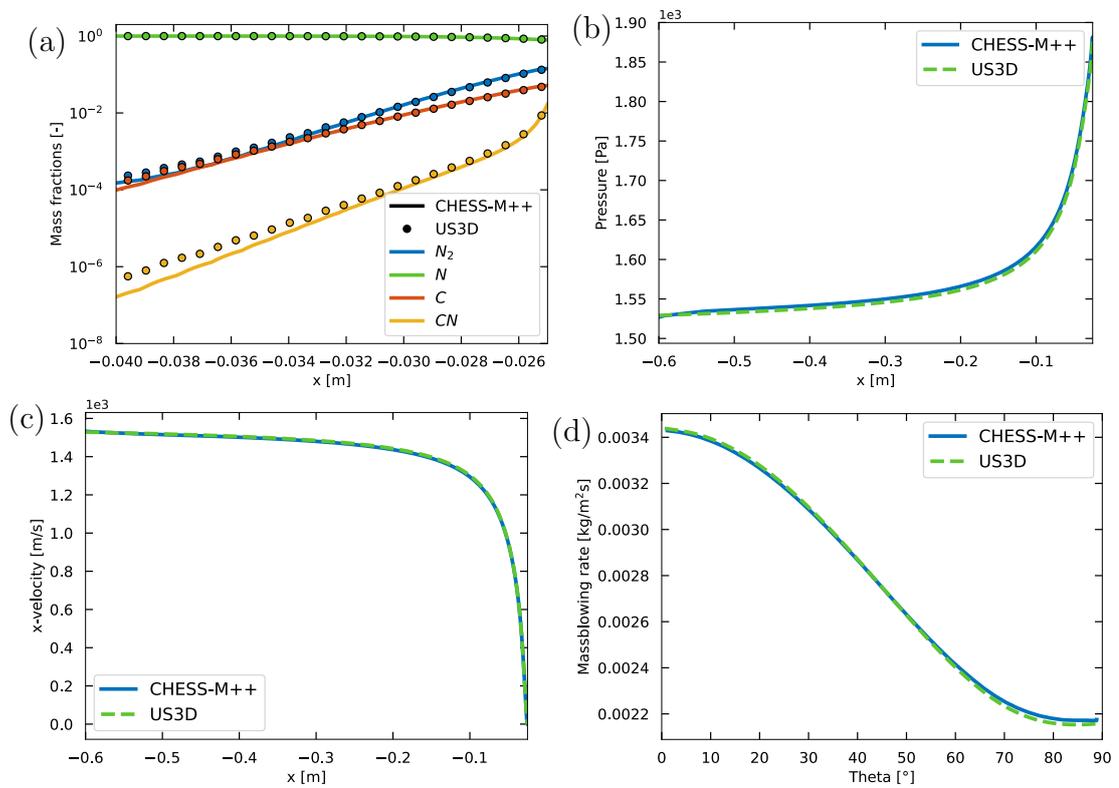


Figure 5.35: Profiles of the mass fractions (a), pressure (b) and stream-wise velocity (c) along the stagnation line. Massblowing rate on the wall (d).

Chapter 6

Conclusive remarks

The first part of this thesis presented the detailed characterization of the fundamental features of hypersonic flows. The models implemented in the solver deal with thermochemical non-equilibrium, including chemistry and internal energy modes excitation. The first model is based on a multitemperature approach, known to be cheap and versatile. On the other hand, the State-to-State kinetics model allows to handle strong non-equilibrium conditions, such as across shock waves and in the boundary layer. In order to reduce the computational time required by the numerical simulations of such challenging flow conditions, the software part of the code is designed in the framework of GPU programming. For this purpose, the code has been developed in a CUDA environment, that allows faster and more efficient calculations. The performance of the parallelization has been also presented, demonstrating a satisfactory scalability of the code.

Preliminary studies have been carried out to verify the correct implementation of each part of the solver. Comparisons with the well known US3D solver and with many reports in the literature showed a very good agreement with benchmark results. The main investigations concerned several configurations. The analysis of hypersonic shock wave/boundary layer interaction over a double-wedge and a double-cone has been presented for low and high enthalpy regimes, reporting interesting findings both from the fluid dynamics and from the non-equilibrium point of view. The mT and the StS models provided similar results, even if some difference appears in the prediction of the wall quantities. In addition, bluff body geometries were also subject to numerical investigations: for these cases, gas-surface interactions have been also analyzed. The hypersonic flow past a cylinder (Knight case) is extremely attractive in the hypersonics community and the results obtained in this work are in agreement with numerical reports and experimental measurements in the literature. Even more accurate results have been highlighted through the coupling with MUTATION⁺⁺ library, confirming the importance of an appropriate description of the transport phenomena. Finally, ablative simulations have been performed to deal with material deterioration. A good agreement has been observed with US3D solver for a subsonic test case.

Future perspectives will involve the extension of the StS model to ablative species (carbon). For this purpose, an effective strategy must be designed to improve the efficiency of the code. An implicit time integration would help in accelerating the simulations, but it would also make GPU environment not suitable given the complexity of the floating point operations required by such an approach. Moreover, a standalone module to deal with complex gas-surface interactions (ablation, partial catalysis) could be implemented to exploit GPU capabilities. Despite the versatility of MUTATION⁺⁺ library, its coupling with the present solver forces CPU executions, increasing the computational time.

Another interesting perspective concerns the extension of the present solver to turbulent flows. The *Reynolds Averaged Navier-Stokes* (RANS) approach is the cheaper one for this purpose, even if vortex structures would be under resolved. However, it represents the starting point for further developments.

Finally, efforts will be dedicated to to development of advanced models for ionized flows. The test cases proposed in this thesis concern flows characterized by temperature values not large enough to induce ionization. However, the formation of electrons is relevant in many atmospheric entry flow applications. In this view, an additional equation is needed for the transport of the electronic energy.

Appendix A

Analytical solution of the 1D diffusion problem with catalysis

Following the formulation proposed by Bariselli et al. [167], plugging Fick's law into the molar continuity equation and looking for the steady state solution, one obtains:

$$\nabla \cdot \left(n \frac{\mathcal{M}_N}{\mathcal{M}} \mathcal{D}_{N_2, N} \nabla (\chi_{N_2}) \right) = 0 \quad (\text{A.1})$$

being n the number density, \mathcal{M} the molar weight of the mixture and χ_{N_2} the molar fraction of molecular nitrogen. In this work, $\mathcal{D}_{N_2, N} = 1.64675 \text{ m}^2/\text{s}$ (obtained from MUTATION⁺⁺). Recalling that $\mathcal{M}_{N_2} = 2\mathcal{M}_N$ and $\mathcal{M} = \sum_s \chi_s \mathcal{M}_s$, the 1D ordinary differential equation to be solved is:

$$\frac{d}{dx} \left(\frac{1}{\chi_{N_2} + 1} \left(\frac{d}{dx} \chi_{N_2} \right) \right) = 0 \quad (\text{A.2})$$

being x the spatial coordinate. Solving for χ_{N_2} yields:

$$\chi_{N_2} = \frac{e^{C_1 \mathcal{M}_N x} e^{C_2 \mathcal{M}_N}}{\mathcal{M}_N} \quad (\text{A.3})$$

where C_1 and C_2 are the integration constants to be found from the boundary conditions. Since the injected mixture is pure atomic nitrogen, the boundary condition at the reservoir is trivial:

$$\chi_{N_2}|_{x=0} = 0 \rightarrow C_2 = \frac{\ln(\mathcal{M}_N)}{\mathcal{M}_N} \quad (\text{A.4})$$

whereas for the wall boundary condition, the diffusion flux must be equal to the chemical production rate:

$$(j_{N_2} = \dot{\omega}_{N_2})|_{x=L} \rightarrow \left(\frac{C_1 \mathcal{M}_N}{2 - e^{C_1 L \mathcal{M}_N}} = \frac{\gamma}{2\mathcal{D}_{N_2 N}} \sqrt{\frac{k_B T}{2\pi m_N}} \right)_{|x=L} \quad (\text{A.5})$$

where k_B is the Boltzmann constant. The last expression could be solved for C_1 iteratively by means of the Newton-Raphson method, as a function of the catalysis efficiency γ . Knowing the two constants C_1 and C_2 , the species distribution is described by Eq. (A.3) as a function of spatial variable x .

Appendix B

2D/3D codes verification

The present 3D solver CHESH is an extension of an older 2D version. In order to verify the correct implementation of the third dimension, the hypersonic flow past a sphere has been simulated and the results compared to the ones obtained through the 2D axial-symmetric simulation. The flow conditions are taken from the work by Nonaka et al. [33], whose experimental measurements were already compared with a numerical investigation to validate the 2D code [21].

In this test case, an air flow past a 7 mm radius spherical sample was considered at $u_\infty = 3490\text{ m/s}$. The other free stream conditions are $p_\infty = 4825\text{ Pa}$ and $T_\infty = 293\text{ K}$. Thermal equilibrium is assumed at free stream and at wall, considered isothermal ($T_w = 1000\text{ K}$). The initial composition is composed of N_2 ($Y_{\text{N}_2} = 0.767$) and O_2 ($Y_{\text{O}_2} = 0.233$). Thermochemical non-equilibrium is modeled employing the mT model, even if it has been shown the the StS model provides a more accurate result [21]. Nevertheless, this goes beyond the scope of the present analysis.

For the purpose of code-to-code verification, a coarse grid is employed for both cases, with a wall resolution of $8 \times 10^{-5}\text{ m}$. The results are illustrated in Fig. B.1, where the temperature and mass fraction profiles are shown along the stagnation streamline. The normalized shock stand-off distance is perfectly reproduced in the 3D configuration, thus assessing the correct implementation of the third dimension in the 3D solver.

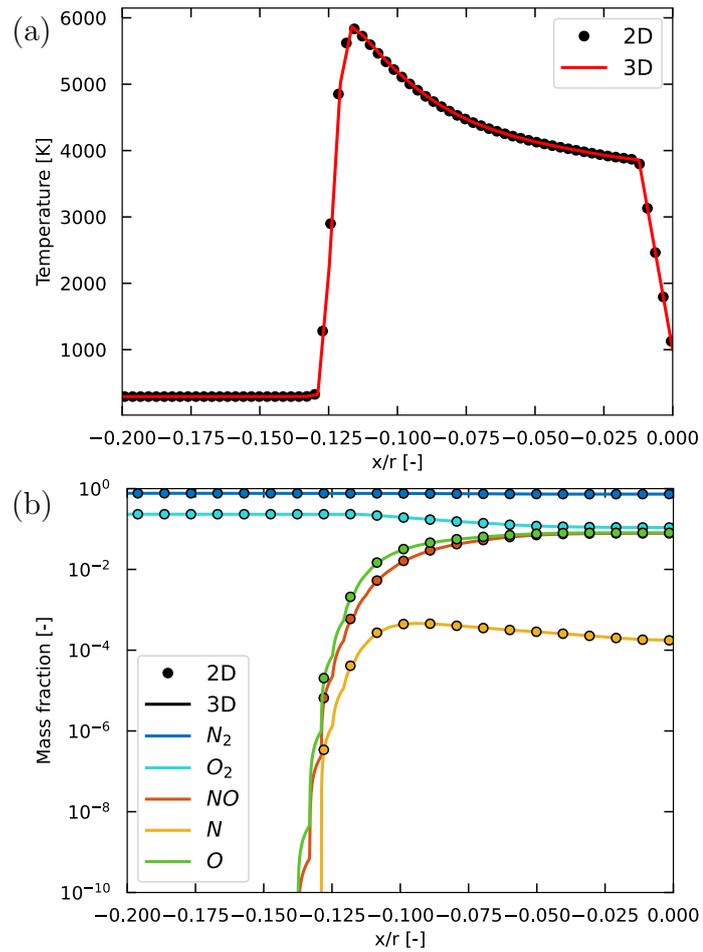


Figure B.1: Temperature (a) and species mass fraction (b) profiles along the stagnation streamline.

Appendix C

Grid independence studies

This appendix is devoted to the presentation of grid independence studies for the cases presented in sections 5.2, 5.3 and 5.4. In order to reduce the computational cost, these studies are carried out with the standalone version of the code CHESSE, except for the ablative test case which is performed with CHESSE-M⁺⁺, as it requires the GSI module of MUTATION⁺⁺.

C.1 SWBLI (sec. 5.2)

Since the HEDW and HEDC flows are stationary, high enthalpy conditions have been selected for the grid convergence analysis. Three grid refinement level were set, listed in table C.1. Following the work by Hao et al. [27], the wall normal spacing for the double-wedge is the same for all the three grids. A wall resolution of $5 \times 10^{-6} m$ was sufficient to solve the boundary layer. Basing on the double-wedge results, the same wall resolution was employed for the double-cone simulations.

Relying of the results shown in Fig. C.1, the medium grids were employed for the main simulations for both double-wedge and double-cone flows.

Table C.1: Number of nodes for the three grid refinement level for the SWBLI test cases.

Test	Coarse	Medium	Fine
Double-wedge	512×256	768×384	1024×512
Double-cone	300×200	500×300	766×384

C.2 Knight case (sec. 5.3)

The grid independence analysis for the *Knight case* was conducted for the non-catalytic case, assuming that catalysis does not affect the convergence of the numerical solution. Table C.2 reports the four refinement levels employed. It is

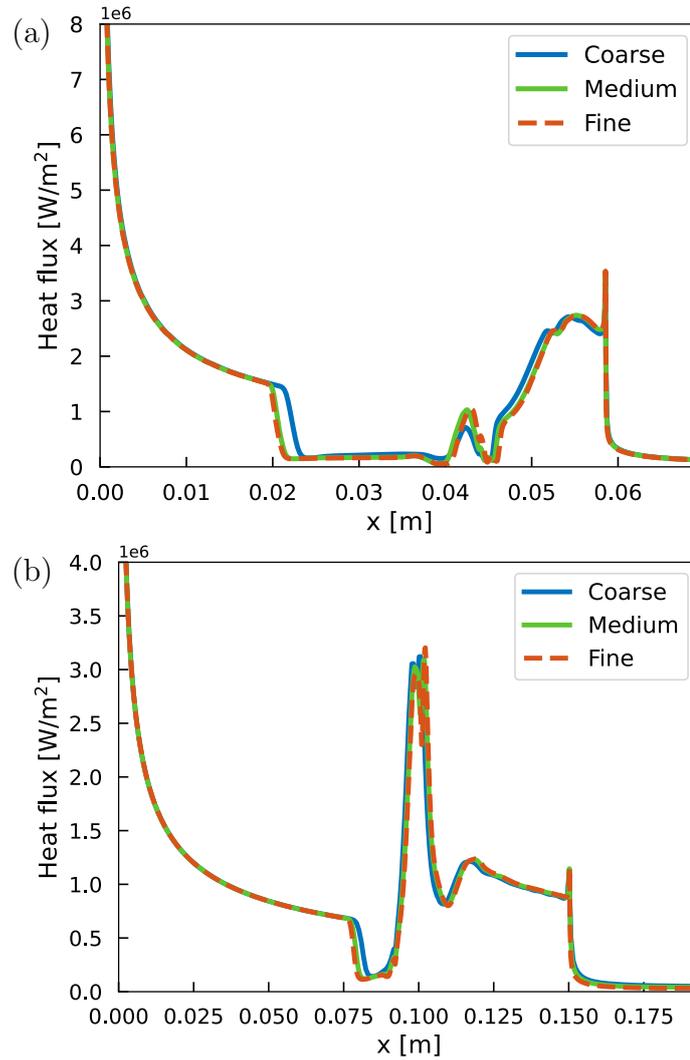


Figure C.1: Heat flux distributions at steady state: double-wedge (a) and double-cone (b).

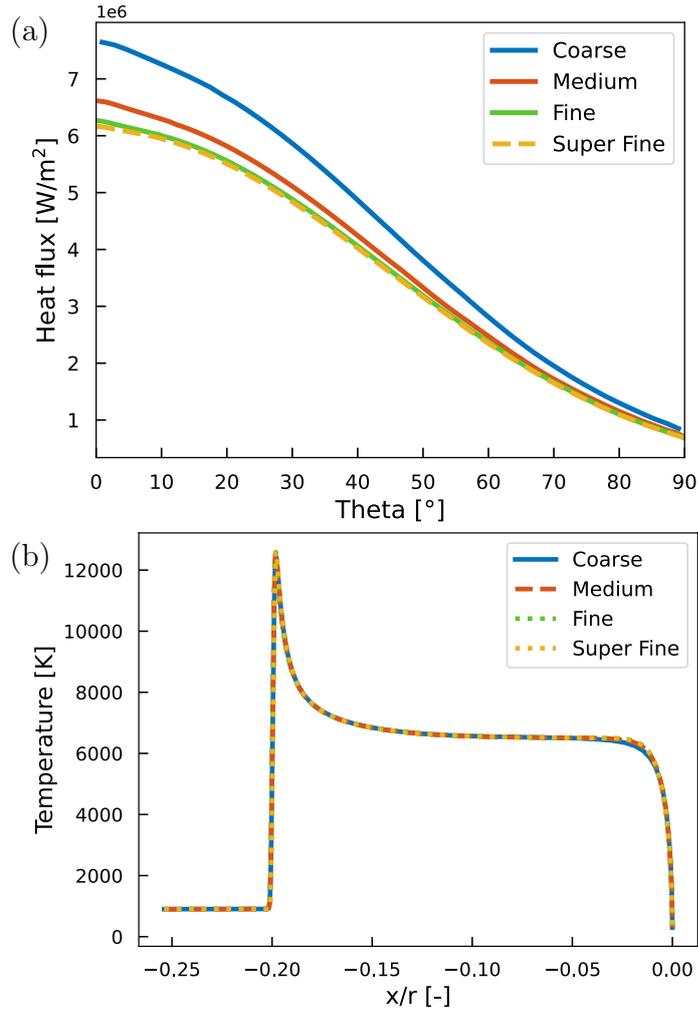
worth underlining that the number of nodes has been double-up in order to obtain geometrically similar grids. In such a case, it is easier to analyze the trend of the error on a specific quantity, such as the heat flux or the shock stand-off distance [168,169].

In Fig. C.2 the wall heat fluxes and the temperature profiles along the stagnation streamline are shown. The four grids demonstrate convergence in terms of shock standoff distance, with slight differences in the prediction of the temperature gradient in the boundary layer. As a consequence, the wall heat flux presents different profiles depending on the refinement level.

To complete this analysis, the order of accuracy has been computed by considering the values of the heat flux in the stagnation point q . Following the formulation given in [169], one has:

Table C.2: Characteristics of the three grid refinement level for the cylinder case.

Refinement level	Nodes	Shock resolution [m]	Wall resolution [m]
Coarse	100×50	1×10^{-5}	2×10^{-7}
Medium	200×100	5×10^{-6}	1×10^{-7}
Fine	400×200	2.5×10^{-6}	5×10^{-8}
Super Fine	800×400	1.25×10^{-6}	2.5×10^{-8}

Figure C.2: Wall heat fluxes (a) and temperature profiles along the stagnation line (b) for the *Knight* case.

$$\hat{p} = \frac{\ln \left(\frac{q_{\text{super fine}} - q_{\text{fine}}}{q_{\text{fine}} - q_{\text{medium}}} \right)}{\ln(2)} = 1.82$$

in agreement with the accuracy of the scheme. Indeed, Fig. C.3 illustrates the values of the heat flux in the stagnation point for the four grids, with the fifth

evaluated through the classical Richardson extrapolation. On the abscissa axis, *ratio* indicates the dimension of the wall resolution with respect to that of the first mesh. Hence, it can be assessed that in terms of shock stand-off distance the solution is grid converged. Concerning the wall heat flux, an even finer mesh would lead to a more accurate solution. However, the simulation would be too demanding and the difference would be less than 1%. For this reason, the fine mesh has been considered for the main investigations.

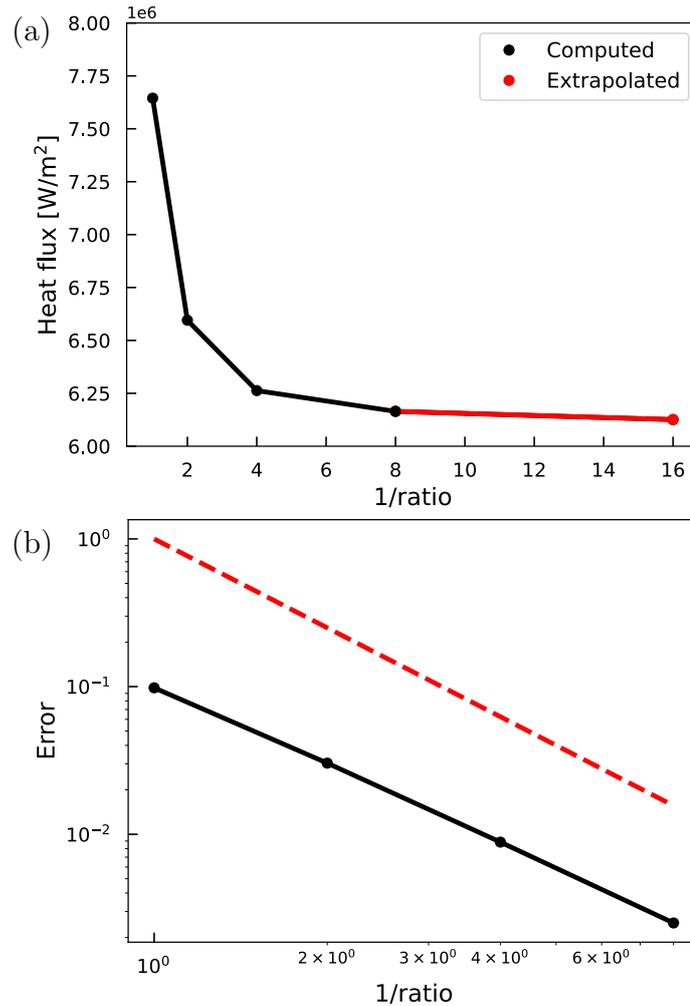


Figure C.3: Values of the heat flux in the stagnation point for the four grids (a) and trend of the error compared to the theoretical solution (b).

C.3 Ablative test case (sec. 5.4)

Finally, the results for the ablative test case have been also verified to be grid converged. Following the analysis carried out by Capriati et al. [165], two refinement level have been chosen, whose characteristics are summarized in table C.3.

In such a case, the convergence is demonstrated with no need of employing a third refinement level. Also, as mentioned in section 5.4, the aim of this test case was a code-to-code verification. Hence, the solution was compared to the reference (US3D solver), provided of the same computational grid. In Fig. C.4 the pressure profiles along the stagnation line and the massblowing rates at wall are illustrated. The results are not affected by the wall resolution nor by the number of control volumes used, confirming that the results are grid converged.

Table C.3: Computational grids employed for the ablative test case.

Mesh	Nodes	Wall resolution [m]
I	100×59	1×10^{-5}
II	200×109	5×10^{-6}

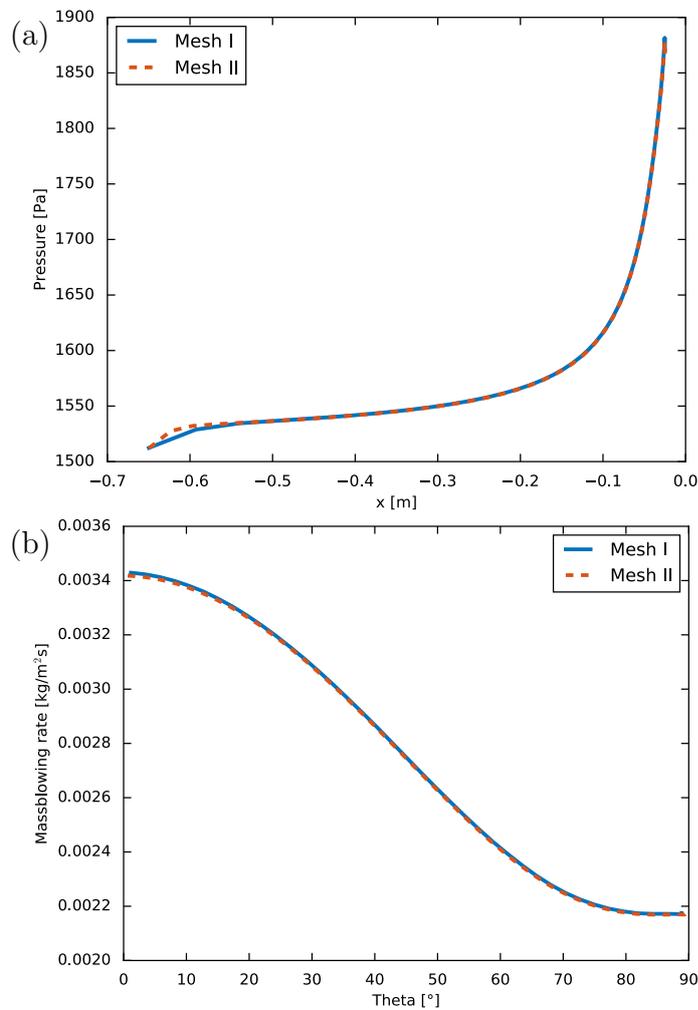


Figure C.4: Comparison of the two grids: pressure profiles along the stagnation line (a) and massblowing rates at wall (b).

Part II :

Immersed boundary approach for hypersonic flows

Chapter 7

Overview of the Immersed Boundary approach

The first part of this thesis is focused on the description of a classical finite volume approach applied in a body-fitted multi-block computational domain, where the grid topology is conformal to the geometry. Such an approach is the most common way to solve the governing equations of the flow. Specifically, the grids employed in this thesis are *structured*, namely each control volume is hexahedral and uniquely defined by three integers (i,j,k) . This strategy implicates several benefits. For instance, there is no ambiguity about the position of the body, where the wall boundary conditions can be directly imposed. Another advantage is the very high aspect ratio of the cells, resulting in very stretched grid portions in the most critical regions of the domain that increase the accuracy of the solution (shocks or walls). On the other hand, there are few drawbacks when dealing with body-fitted grids. For instance, the generation of the grid can be very complicated and costly if the geometry is complex. In such cases, one can think to employ an *unstructured* grid, where the control volumes present a generic geometry. An example is given in Fig. 7.1. Of course the discretization of the equation can not follow a *structured* topology since the cells are not ordered. This leads to the need of a generalization of the structure of the solver. Another disadvantage is the difficulty in handling moving geometries, involving both rigid bodies moving into the flow field as well as deformable bodies, as in the case of ablation with recession.

In this context, the Immersed Boundary Methods (IBM) come to rescue. The philosophy of the IBM is based on the overlap of two different grids, describing the geometry (*Lagrangian grid*) and the fluid domain (*Eulerian grid*). An example is given in Fig. 7.3: the *Lagrangian grid* is a discretization of the surface \mathcal{S} of the body by means of a finite amount of markers (black dot in Fig. 7.3). These are normally triangular surfaces. The higher is the number of triangles, the better is the accuracy of the surface geometry. The *Eulerian grid* discretizes the fluid region and presents a Cartesian mesh, with no need to be conformal to the body. In this way, the solver is very easy to implement given the Cartesian structure of the com-

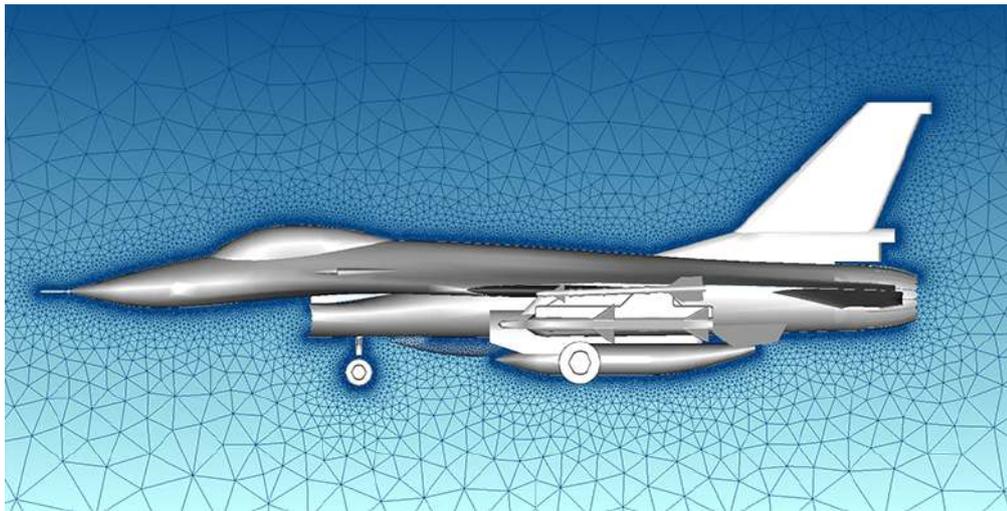


Figure 7.1: Example of an *unstructured* grid to simulate the flow past an aircraft. Credits to Pointwise website.

putational domain. Moreover, the grid is generated once in pre-processing, with no need for a re-meshing procedure if the body is moving. Nevertheless, the solver is not aware of the position of the body, and the wall boundary conditions can not be directly imposed on a certain border of the domain. Hence, a modification of the classical governing equations is needed in the vicinity of the body (yellow cells in Fig. 7.3). Such a modification is based on the introduction of a forcing term in the equations to make them obey to the wall boundary conditions [47, 58]. Of course this implies the necessity of knowing where this forcing has to be applied. For this purpose, a ray-tracing procedure is employed to classify each cell in the domain. The easiest classification, shown in Fig. 7.2, is the following:

- *solid cell*, each cell whose center lies into the body;
- *fluid cell*, each cell whose center lies outside the body;
- *interface cell*, each cell confining at least with one *solid cell*.

The first IB approach was proposed by Peskin [48] to perform numerical simulations of the blood in a heart valve. The first formulation was based on the calculation of the body forces exerted on the fluid, in turn provoking the motion of the boundary. This kind of technique is referred to as *continuous forcing* [47], since the forcing term is incorporated into the equations before the discretization. Peskin improved it in his next works [49, 50] to include several configurations, involving different kinds of valves. The main drawback of this approach was the difficulty in computing the response of the boundary to the forces (pressure and stress). Also, it was unsuitable for rigid bodies configurations: in principle, the deformability of the boundary could be decreased to make it rigid, but this results

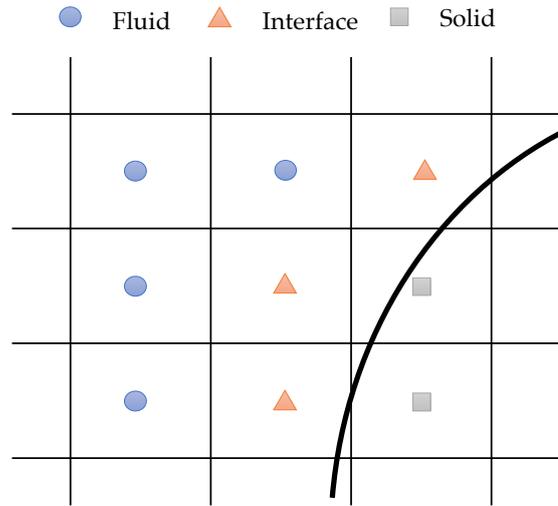


Figure 7.2: Example of cell classification.

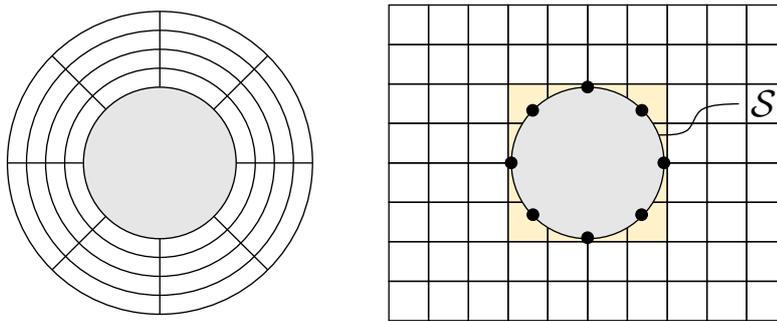


Figure 7.3: Comparison of the grids employed for a body-fitted approach (left) and an immersed boundary approach (right).

in a numerical stiffness that makes this method inefficient. The first rigid boundary applications were presented with by Basdevant and Sadourny [51] and then extended by Briscolini and Santangelo [52]. In their method, known as MASK method, the velocity field was directly imposed in the cells near the wall to satisfy the no-slip condition, even if a low accuracy of the solution was found in the vicinity of the body, showing spurious oscillations due to the sharp change in the values of the velocity. A similar formulation was proposed by Goldstein et al. [53], who exploited a central finite difference scheme to propagate the forcing in the cells near the surface smoothly. This approach avoided the appearance of oscillations, but it required the distribution of the forcing term along the surface. Also, the expression of the forcing contained two free parameters that made such an approach problem-dependent, namely computationally inefficient. A substantial improvement was introduced by *discrete forcing approaches* [47] proposed by Modh-Yusof [54], Fadlun [55] and Balaras [56]. Contrary to the *continuous forcing*, this approach extracts the forcing term by exploiting the numerical solution (flow

field), once the discretization of the equations is performed. Specifically, Modh-Yusof [54] proposed to apply the forcing not only in the vicinity of the surface, but also in the nearest interior cells, in order to obtain a smooth velocity field. Fadlun [55] followed this idea: the forcing can be applied by interpolating the solution along a certain Cartesian direction, as a function of the value on the wall and a value in a fluid cell. Nevertheless, this leads to an ambiguity in choosing the interpolation direction, especially when the curvature of the body is relevant. To overcome such an issue, Balaras [56] extended this formulation proposing an effective way to perform a reconstruction along the wall normal direction and exploiting the wall boundary condition. In this way, the forcing was suitable for both Dirichlet and Neumann boundary conditions and is second order accurate. Differently, Kim et al. [170,171] suggested to apply the forcing in the points inside the body. Such an approach found application also for turbulent flows [172] showing a good agreement with the expectations. Later works involved the extension of these methodologies to an hybrid approach, known as *moving least square* [173]: in this formulation, the solution in the forcing cell is reconstructed as a function of the surrounding points whether they are inside or outside the body. This results in an efficient way to treat also deformable or moving geometries and is nowadays employed for many applications [174–177].

It is noteworthy that all the aforementioned formulations are focused on incompressible flows. One of the first applications to a (weakly) compressible flow was proposed by De Tullio et al. [57], who simulated supersonic flows past a cylinder and an airfoil. The forcing was applied by means of a linear distance weighted interpolation, exploiting the neighbors of the interface cells. A local refinement strategy was adopted to increase the accuracy in the critical region of the computational domain. The results showed a very good agreement with references; nevertheless, only wall pressure distributions were analyzed, whereas the evaluation of the skin friction and the heat flux was not presented. They are known to be the most critical quantities to compute, as the velocity and temperature gradients at wall are very sharp. An alternative idea to increase accuracy and stability of the scheme was proposed by Brehm and Fasel [178] in the framework of incompressible flows, then extended to compressible [179] and moving boundary [180] configurations. In these works, the main idea is to perform an optimization analysis in order to derive a suitable set of coefficients for a finite-difference formulation on an irregular stencil. Results for the flow past a sphere or through an open-rotor were satisfactory.

Nevertheless, in the framework of hypersonic flows, a different approach was needed to correctly evaluate temperature gradients at wall. These are known to be extremely sharp during the re-entry and their calculation is mandatory for an opportune design of the heat shield. Classical Cartesian grids were employed as first attempt, with good results for friendly configurations such as flat plates or compression ramps [181]. In particular, these flows did not present extreme temperature regimes, and a good agreement was reported with both numerical

and experimental findings. On the contrary, simulations of hypersonic flows past bluff bodies did not provide good results in terms of heat flux. Specifically, Brahmachary et al. [182] performed numerical investigations of perfect gas flow past a cylinder and past a sphere-cone sample. In both cases, the pressure distribution was coherent with the expectations, but the heat flux was drastically underestimated in the stagnation point. A comparison with a body-conformal algorithm was also provided, highlighting that the issue was related to the reconstruction of the wall temperature gradients. The authors tried to address this inconvenience to both resolution and reconstruction errors, but no convincing reason was found and the problem of the heat flux remained still open. To overcome such an issue, a different wall treatment is needed.

Recently, in the work by McQuaid et al. [183, 184] a novel concept has been presented. To overcome the issue of the temperature gradient reconstruction, they overlap a body-conformal grid to the Cartesian one. The equations are solved on the first grid and the solution is then projected to the Cartesian nodes. Such an approach provides very good results also in the presence of strong bow shocks in bluff body configurations. Nevertheless, the issue of moving boundaries remains unfixed. Moreover, the main drawback of this method is that the formulation loses the main philosophy of the immersed boundary approach, namely the possibility to employ a purely Cartesian domain.

In the following sections, canonical configurations are object of numerical investigations, in order to address possible reasons of the main issues in hypersonic flows using an immersed boundary method.

Chapter 8

Present technique

8.1 Generation of the mesh

In the framework of the immersed boundary methods, one of the benefits is the generation of the computational grid. Specifically, when dealing with complex bodies, the definition of the boundaries of the domain and the topology of the blocks composing a classical body-fitted grid can be hard. Also, thinking about moving body applications, as in case of recessing surfaces, the grid should be updated to deal with the motion of the boundaries. This procedure should be applied run-time, thus implying a relevant increment of the computational cost required by the simulation. On the other hand, the employment of a Cartesian grid requires the minimal user intervention. The mesh generation can be performed once in pre-processing through external tools. Once the grid is exported, it can be read by the code to create the data structure useful for the numerical solver. The grid is refined according to the critical region of the flow field (boundaries, shocks, wakes) so that the accuracy is ensured also in case of moving bodies.

In the present algorithm, the mesh is created through an external tool called TOMMIE [185], which takes a geometry file as an input. STL formats are widely developed for such a purpose. An example of triangulation for a re-entry capsule is given in Fig. 8.1. The user can easily identify the regions to be refined, according to the position of the body within the domain. An example is given in Fig. 8.2, where the grid is refined for a flow past a cylinder. The strategy is applied near the wall and in the detached shock region.

Of course, the refinement strategy follows certain criteria that allow to obtain a coherent splitting of the cells in each Cartesian direction. The main constraints are based on the approach by Iaccarino and Ham [185–187] and are here listed:

1. the maximum coarsening level in each direction must be one, namely each cell can have at most 2 neighbors for 2D configurations and 4 for 3D configurations;
2. all the faces associated to a specific Cartesian direction must present the

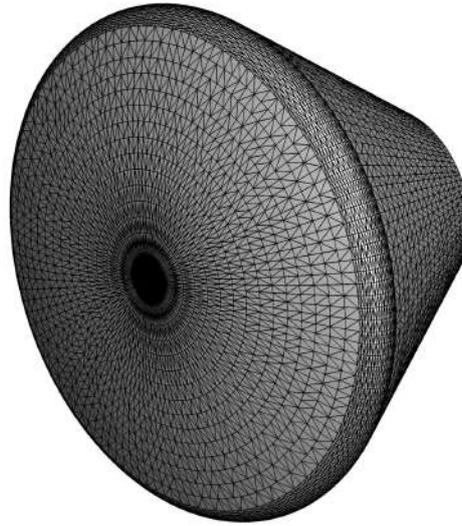


Figure 8.1: Example of triangulated surface for a re-entry capsule.

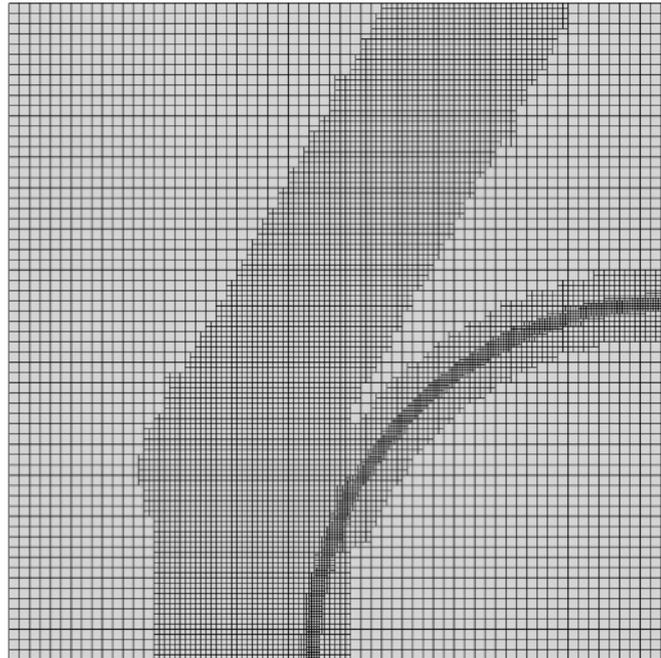


Figure 8.2: Example of a refined mesh for the flow past a cylinder.

same coarsening level, such a way it is impossible that a cell has 3 neighbors;

3. normal surface vectors must pass through the center of at least one neighbor cell.

Following these rules there is no ambiguity about the number of neighbors in case of a refinement, making easier the generation of the data structure. Fig. 8.3

shows examples of correct and incorrect configurations.

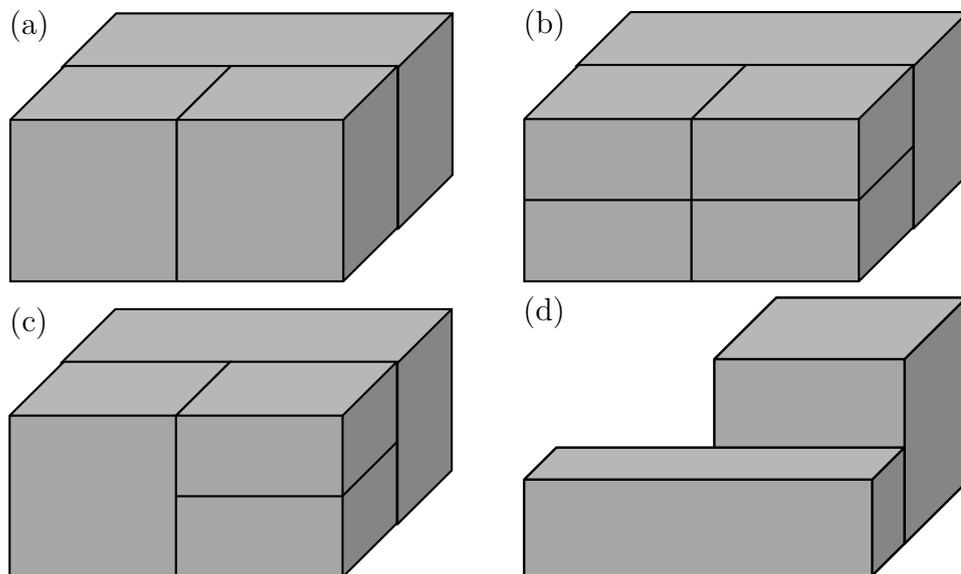


Figure 8.3: Example of refinement configurations: correct (a-b) and incorrect (c-d) for violation of rules 2 and 3 respectively.

The so generated grid follows a *semi-structured* topology. This means that all the information about the connectivity, faces, cells and nodes are collected in a `.cas` file, normally exported in ASCII format. This file must be read properly in order to create the data structure needed by the numerical solver. This procedure is explained in detail in section 8.2.

8.2 Data structure

8.2.1 Connectivity

The generation of the data structure is a fundamental aspect of the immersed boundary solver. Since the grid is locally refined, the control volumes can not be identified through the classical integer indices (i, j, k) as for the body-fitted structured grids. Hence, each cell will be uniquely defined by a single integer i , representing its identity. The same concept is applied for faces and nodes. Note that the internal cells will be defined by an integer $i \in [1, N_cells]$, being `N_cells` the number of internal cells, known from the mesh file. On the other hand, all the cells defined by an integer $i \in [N_cells + 1, TOT_CELLS]$ are the ghost cells of the domain. Note that `TOT_CELLS` represents the total number of cells, including both internal and ghost ones. Its definition is explained in section 8.2.3.

Once the number of cells, faces and nodes are all known, connectivity information must be stored as they are necessary for the numerical solver. For example, cell center or node coordinates are necessary to apply the spatial discretization,

along with information about the neighbors of each cell in all the Cartesian directions. Other examples are the cells sharing a specific node to evaluate vertex flow variables, or normal vector and surface extension of each face. For this reason, arrays of `struct` variables are allocated for cells, nodes and faces. All the information needed are reported in detail in appendix D.

Of course, dealing with an immersed boundary solver an efficient research algorithm must be implemented to classify the cells as *interface*, *solid* or *fluid*. The procedure is called *ray-tracing* and is explained in section 8.2.2.

8.2.2 Ray-tracing procedure

As already mentioned, an immersed boundary solver is not aware *a-priori* of the position of the body within the computational domain. Hence, a *ray-tracing* procedure is necessary to classify the cells where the forcing must be applied to satisfy a certain boundary condition. Assume that $\mathbf{C} = (x_C, y_C, z_C)$ is a control point in the far field (far from the body). The objective of the *ray-tracing* is to find out if a query point $\mathbf{Q} = (x_Q, y_Q, z_Q)$ lies inside the solid domain. For this purpose, a *bounding box* is created to limit the number of query points and speed-up the research. Its box can be easily set in the algorithm depending on the dimension of the body. For the case of the cylinder, for instance, the extension could be $1.5r \times 1.5r$ being r the radius of the cylinder.

If \mathbf{Q} lies into the body, then the segment $\mathbf{R} = (x_Q - x_C, y_Q - y_C, z_Q - z_C)$ intercepts the surface in (at least) one point. In general, if the number of intersections is odd, \mathbf{Q} is inside the body; contrarily, if the number of intersections is even, \mathbf{Q} is outside the body, as shown in Fig. 8.4.

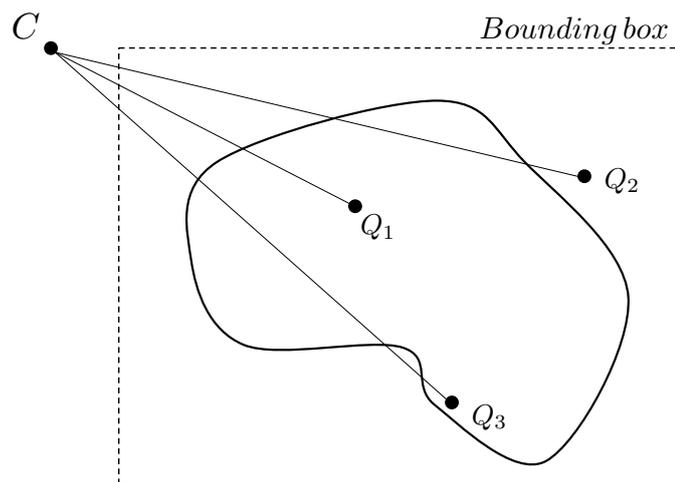


Figure 8.4: Example of points lying inside and outside the body.

Since the surface is discretized by triangles, the geometrical meaning of the *ray-tracing* is the intersection between the segment \mathbf{R} and a triangle. The algorithm is explained in detail in appendix E. All the cells whose center lies into the solid

are tagged with a flag -1 (*solid*), whereas the *interface* cells are tagged with a flag 1. All the other fluid cells are tagged with a flag 0.

Finally, it deserves to be pointed out that increasing the wall resolution implies increasing the number of triangles of the STL file. In general, the geometry is not continuous as it is discretized by a finite amount of triangles and this leads to the presence of holes on the surface. The *ray-tracing* procedure can fail if the quality of the surface is not adequate for a certain wall resolution, namely if a cell near the wall completely lies into a hole.

8.2.3 Definition of ghost cells

Ghost cells are used to ensure the same spatial reconstruction of the solution and the same accuracy also at the boundaries of the computational domain or between two adjacent processors. For the purpose of second order MUSCL reconstruction, the allocation of two ghost cells is required. Such a procedure is trivial when dealing with structured grid, either Cartesian or body-fitted, as the number of volumes along a certain direction is fixed. In case of unstructured or semi-structured grids, the number of cells along each Cartesian direction can vary depending on the refinement strategy, leading to an ambiguity in defining the ghost cells. As an example, one can consider the situation illustrated in Fig. 8.5(a). In case of the upper row of cells, the definition of $G1$ and $G2$ is straightforward. On the contrary, in the lower row the refinement makes this task harder as the cell $G2$ should be split. Second order spatial reconstruction requires to know the solution in both $G1$ and $G2$ or at least the variables' gradient at the interface f_{ghost} . In this sense, one can take advantage of the property of the boundary conditions. Dealing with supersonic outflow boundary conditions, the values in the ghost cells are copied from the fluid domain, as explained in section 3.5. Specifically, the values in the fluid cell $F1$ are copied to cells $G1$ and $G2$. In this way, the gradients between the two ghost cells are null. Only for symmetry boundary conditions, non-null gradients could appear on a certain velocity component since values from $F1$ and $F2$ are copied to $G1$ and $G2$ respectively. For these reasons, an effective strategy is implemented to define the domain ghost cells. First of all, only one ghost cell is allocated, as shown in Fig. 8.5(b), allowing also to save memory. This task is easily accomplished because it is immediate to identify the boundary cells, since they have (at least) one neighbor with index 0 (thus, no neighbor) in the initial data structure. Hence, an integer $i > N_cells$ is assigned to the ghost cell, following the order of definition. Secondly, the gradients at the ghost interface f_{ghost} are known *a-priori*, as they are always null except for symmetry boundary conditions. In such a case, there is no ambiguity in imposing $Q_{grad}(f_{fluid}) = Q_{grad}(f_{ghost})$ for a certain velocity component, as shown in section 3.5.

It should be pointed out that this procedure is valid only for the boundaries of the computational domain. On the contrary, in case of parallel executions the definition of the ghost cells at the boundaries of adjacent processors is not trivial.

In this case, the gradients need to be carefully evaluated as they are generally non-null. This will be explained in section 8.4.

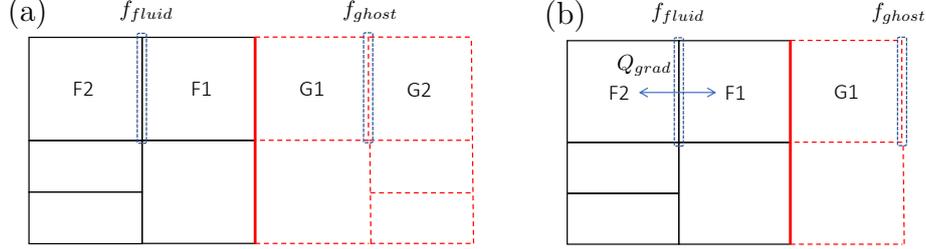


Figure 8.5: Example of ghost cells: configuration (a) is not applicable, configuration (b) is applicable.

8.2.4 Flux formulation

When solving the governing equations, the numerical fluxes at each interface of the control volumes must be evaluated. Dealing with conservation laws and considering a generic control volume, the ingoing flux must be balanced by the outgoing flux, as sketched in Fig. 3.1. As an example, one can consider the 1D configuration given in Fig. 8.6, where subscripts l and r stand for *left* and *right* respectively. Cell A presents one left neighbor (cell D) and two right neighbors (cells B and C). Hence, the ingoing flux is just F_1^l , whereas the outgoing flux is given by the sum of two contributions, namely F_1^r and F_2^r : these are the fluxes across the faces f_1^r (from cell A to cell B) and f_2^r (from cell A to cell C). Obviously the same approach is applied for the other directions.

Since the number of faces composing a cell is stored in the data structure, this formulation can be generalized. For a generic PDE:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0 \longrightarrow \frac{dU}{dt} = -\frac{\sum_{f=1}^{N_r} F_f^r - \sum_{f=1}^{N_l} F_f^l}{\Delta x} \quad (8.1)$$

being N_x the number of neighbors on the given x direction (left or right). A last comment is mandatory for the advection term. The MUSCL reconstruction presented in section 3.2 requires the knowledge of the difference of a given flow variable ϕ between two consecutive cells ($\delta\phi$ in Eq. (3.10)). In the example illustrated in Fig. 8.6, $\delta\phi$ is needed across face f_1^l and across the generic interface on the right side of cell A for a second order reconstruction of the right state. Since A presents two right neighbors, ϕ_B and ϕ_C are averaged to account for both of them. Note that this is valid only for a second order discretization. Indeed for a first order scheme, the left and right states always correspond to the solution in the left and right cells with respect to a certain face. For instance, the left state and right states on the face f_1^r are just ϕ_A and ϕ_B respectively.

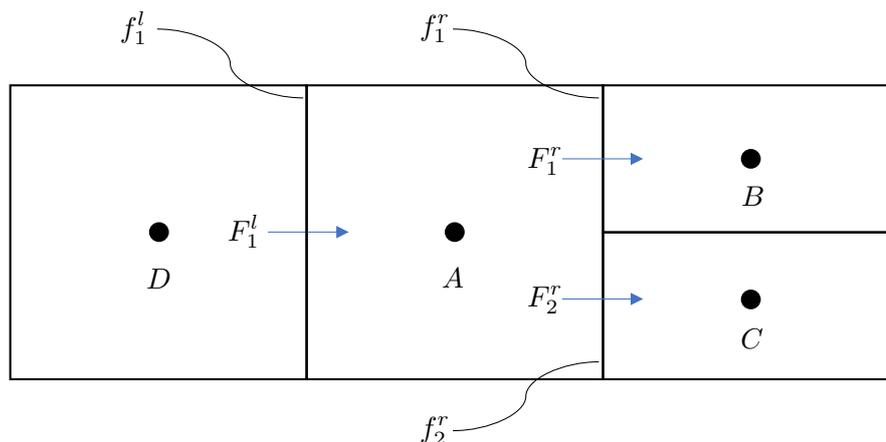


Figure 8.6: Example of flux formulation with local refinement.

8.3 Boundary reconstruction

In classical body-fitted approaches, at least one of the boundaries of the computational domain corresponds to the wall of the body invested by the flow. This allows to directly impose a wall boundary condition by fixing the values of the conservative in the ghost cells, as shown in 3.5. On the contrary, the philosophy of the immersed boundary methods is to keep the freedom of overlapping the *Lagrangian grid* (solid) to the *Eulerian grid* (fluid). In such a way, the wall of the geometry considered does not correspond to any of the boundaries of the domain. More in detail, the *ray-tracing* procedure presented in section 8.2.2 comes to rescue to identify the position of the body within the computational domain.

Once the *ray-tracing* procedure is completed, a forcing term is introduced in the equations to satisfy the boundary conditions on the surface. This is numerically achieved performing a reconstruction of the solution in the *interface* cells. Such a reconstruction depends on the type of boundary condition to be imposed at wall. In general, two kinds of boundary condition can be imposed, namely Dirichlet boundary condition and Neumann boundary condition. The first one allows to impose a specific value of the variable on the wall (e.g. velocity or temperature); the second one allows to impose a specific value of the gradient of a certain quantity, as in the case of the pressure. Depending on the kind of condition, the reconstruction of the solution change. In this sense, the strategy proposed by Fadlun et al. [55] is not always suitable as it is based on a linear interpolation along a certain Cartesian direction, defined by the higher wall normal component. Thinking about a cylindrical geometry, such a formulation leads to the ambiguity of choosing a specific direction. To overcome this issue, Balaras [56] proposed a novel formulation based on the bilinear interpolation (trilinear in 3D) along the wall normal direction, resulting in an effective way to impose both the Dirichlet and Neumann boundary condition types.

The present formulation follows the idea proposed by Balaras, as the geometries

under consideration often present a bluff shape. This technique is commonly used in many works available in the literature [58, 182]. In order to reconstruct the solution in the *interface* cells, the values of the specific variables in the nearest cells are used. A 2D example of the support domain is sketched in Fig. 8.7. Points A and B are taken as reference in the fluid domain, whereas W is the wall normal projection of I. The extension to a 3D formulation is straightforward. A note is here mandatory: points A and B should be purely *fluid* (not *interface*). Hence, a research algorithm should account for this condition in order to readapt the position of such points in case their tag is 1. Nevertheless, when one of the wall normal vector components is predominant with respect to the other (as for example in the vicinity of the stagnation point of a cylinder), the displacement of these support points would be too large, decreasing the accuracy of the interpolation and also leading to parallelization issues. Hence, points A and B will be always the nearest cells of I. Anyway, preliminary analysis showed no relevant differences in the results.

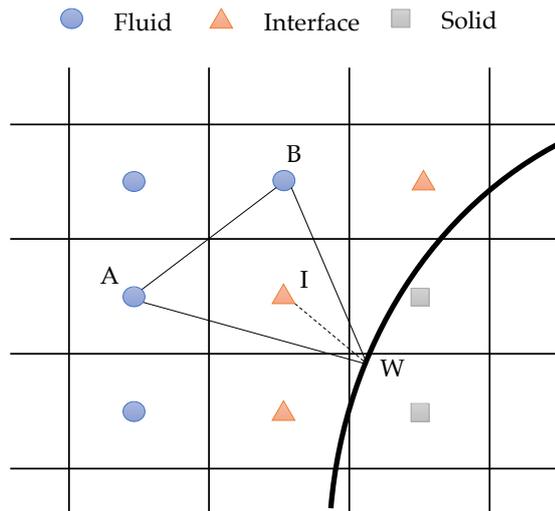


Figure 8.7: Sketch of the support domain for the solution reconstruction near the wall.

In the following, the 2D formulation is presented, being the extension to 3D trivial. Indicating with ϕ the generic flow variable, each quantity of interest is reconstructed in the cell I as:

$$\phi_I = C_1 x_I + C_2 y_I + C_3 \quad (8.2)$$

being (x_I, y_I) the spatial coordinates of cell I. Coefficients C_1 , C_2 and C_3 are evaluated applying the same formula in points A, B and W. As to do so, the following linear system of equations must be solved:

$$\begin{bmatrix} \phi_A \\ \phi_B \\ \phi_W \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix} \begin{bmatrix} x_A & y_A & 1 \\ x_B & y_B & 1 \\ x_W & y_W & 1 \end{bmatrix} \quad (8.3)$$

where ϕ_W is the desired value of the quantity on the wall. For the velocity, this is null if a no-slip boundary condition is imposed. For the temperature, this is the wall temperature value in case of isothermal wall. These are two simple cases of Dirichelet boundary conditions. On the other hand, the last equation of the system given in Eq. (8.3) must be modified in case of a Neumann boundary conditions, as for the pressure or the temperature for an adiabatic wall. This reads:

$$\frac{\partial \phi}{\partial n|_W} = \underbrace{\frac{\partial \phi}{\partial x|_W}}_{C_1} n_x + \underbrace{\frac{\partial \phi}{\partial y|_W}}_{C_2} n_y \quad (8.4)$$

where (n_x, n_y) are the outgoing wall normal components in the Cartesian frame and of course $C_3 = 0$. The complete list of boundary conditions applied is reported in table 8.1. The forcing is applied on the primitive variables, but once completed also the conservative ones are updated.

Table 8.1: Wall boundary conditions for IB.

Quantity	Case	BC
Velocity	No-slip	$u_W = v_W = w_W = 0$
Pressure	Boundary layer	$\partial p / \partial n _W = 0$
Temperature (transl. and vib.)	Isothermal wall	$T = T_W$
	Adiabatic wall	$\partial T / \partial n _W = 0$
Mass fractions	Inert wall	$\partial Y_s / \partial n _W = 0$
	GSI	$Y_s = Y_{s,W}$

On the other hand, the treatment of the *solid* cell is different with respect to that of the *interface* cells. All the cells whose center lies into the solid have not a real physical meaning, and there are many possibilities to proceed [188]. For example, a null value is imposed for the velocity, whereas the temperature is fixed to the wall value (T_w) [57, 58]. Nevertheless, there is lot of uncertainty about this treatment of the *solid* cells, since it was demonstrated to be suitable only for incompressible or weakly compressible flows [57, 186]. In some cases, part of the *solid* cells are used as ghost cells to extrapolate suitable values of the flow variables into the fluid [189, 190]. However, this did not provide different results with respect to the present technique, but for sure it deserves further investigations in the view of a correct flux reconstruction.

The present formulation presents pros and cons. The main benefit is its versatility: since the reconstruction is applied along the wall normal direction, both Dirichelet and Neumann boundary conditions can be performed. Moreover, the

identification of the support domain (cells A and B in Fig. 8.7) is trivial and cheap. In the view of recessing wall simulations, the generation of the data structure is very easy, as it just requires the call to the *ray-tracing* procedure to re-tag the cells. On the other hand, since the variables are interpolated in the *interface* cells, the equations are not actually conserved near the wall, as explained in chapter 9. As a consequence, this method requires very fine grids when the wall gradients are sharp, as in the case of high Mach number flows, even if it can fail also when employing very fine grids. More sophisticated algorithms exist, such as the *cut-cell* method [47,191,192] that ensures accuracy and conservation also at wall. Through this approach the numerical fluxes are computed across the real surface of the body, obtained through the interceptions of the wall with the Cartesian mesh. Such a method is very accurate also for coarse grids, but demands attention in the definition of the data structure, which is not trivial especially in case of moving or 3D boundaries.

A similar approach has been proposed by McQuaid et al. [184]: this technique allows to overlap a body-conformal grid to a Cartesian grid just close to the wall. In such a way, the solver preserves its Cartesian structure and is accurate when evaluating the wall gradients. This approach is similar to the use of the wall functions. Indeed, the main drawback is the need of solving the governing equations on an 'external' body-conformal grid, in order to project the solution on the Cartesian grid.

8.3.1 Wall quantity evaluation

In order to evaluate the wall quantities, such as pressure or heat flux distributions, the method employed to interpolate the solution in the *interface* cells is revisited to be adopted for this purpose. First of all, the points where the forces are calculated (referred to as *probes*) must be placed at the same distance from the surface. This implies that, in general, probe coordinates do not coincide with any cell centers. A sketch of the 2D formulation is given in Fig. 8.8, where M stands for *mother*. It is the cell where the probe point lies, needed to define points A and B. The mother cell can be any of the fluid cells, depending on the distance of the probe from the wall. It is defined opportunely: indeed, it is a common way to impose a distance from the wall about $1.5 \div 2$ times the local spatial resolution. This allows the probe to lie in the purely fluid region, where the solution is much more accurate with respect to the *interface* cells. Once the spatial location of P is known, it is projected on the wall and the generic variable is computed as $\phi_P = C_1 x_P + C_2 y_P + C_3$, where coefficients C_1 , C_2 and C_3 are calculated applying the same formula in points A, B and W, accordingly to the kind of the boundary condition (Dirichelet or Neumann). The extension of this formulation to a 3D configuration is straightforward.

Once ϕ_P is computed, the wall normal derivatives are discretized following a first order expression. For example, the heat flux is calculated as:

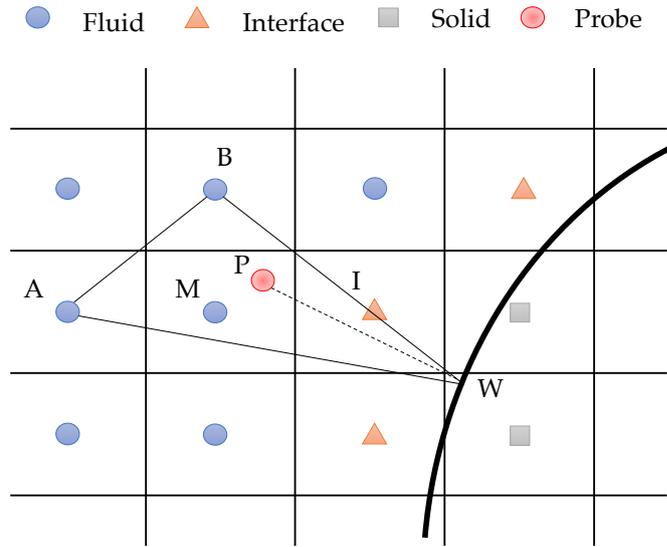


Figure 8.8: Sketch of the interpolation procedure in the *probe* point.

$$q = -\kappa \frac{\partial T}{\partial n|_W} = -\kappa_P \frac{T_P - T_W}{\Delta n_P} \quad (8.5)$$

being Δn_P the distance of P from the wall and κ_P the thermal conductivity in the probe. A similar expression is used to calculate the shear stress, substituting the thermal conductivity with the viscosity and the temperature with the tangential velocity. So the overall algorithm is here summarized:

- define Δn_P , allocate the *probes* and find their projection on the wall (point W);
- find where each probe point lies (*mother* cell M) to define points A and B;
- evaluate coefficients C_1 , C_2 and C_3 ;
- calculate the variables of interest in P;
- calculate wall gradients applying a first order derivative.

Lastly, it is noteworthy that this approach is effective provided that the solution in points A and B is accurate.

8.4 Parallelization procedure

8.4.1 Partitioning of the domain

Dealing with unstructured grids, the partitioning of the domain is not trivial. The number of control volumes assigned to each block has to be balanced according

to their density in a specific region of the domain. For this purpose, METIS library [193] is coupled with the code. Such a library allows for an optimal balance of a graph. In this case, the graph is represented by the computational grid, in which the control volumes have to be redistributed among the portions, namely the processors. An example of partitioned grid (4 processors) is given in Fig. 8.9. Of course, the partitioning does not follow a Cartesian topology because of the local refinement. As a matter of fact, a specific research algorithm must be devoted to identify the boundaries between the processors.

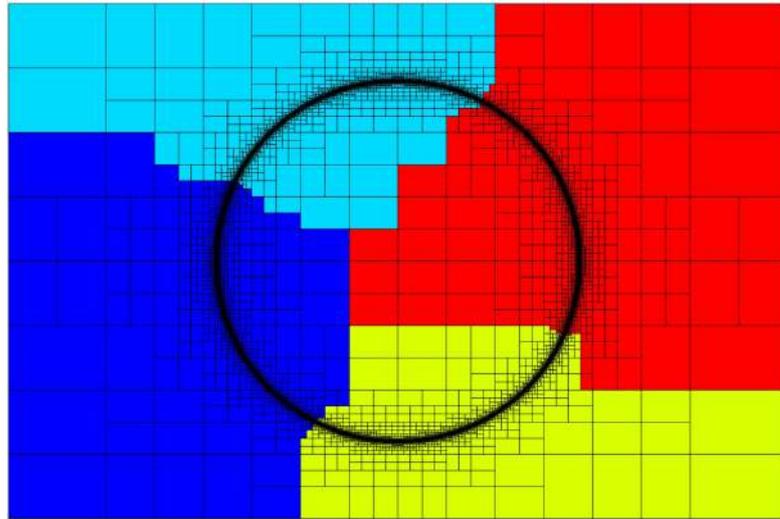


Figure 8.9: Example of METIS partitioning in 4 subdomains.

When partitioning the domain, two main issues must be handled.

- The index of each cell, node and face must be re-adapted to be suitable for the local portion of the domain. This allows to make easier the definition and allocation of the local data structure.
- One should identify the cells, nodes and faces at the boundary of the processors, in order to properly define the ghost cells. This task is harder than the definition of the domain ghost cells presented in section 8.2.3, as the gradients are in general non-null. Moreover, given the complexity of the partitioning, the allocation of the ghost cells is not trivial as each processor could communicate with more than one processor (see Fig. 8.9).

These two aspects are analyzed in section 8.4.2.

8.4.2 Data structure re-adaptation

The connectivity information collected in the mesh file exported by TOMMIE refer to the global domain, namely the domain used by one processor in case of a

serial execution. For the purpose of a parallel execution, different portions of the domain are created by METIS and the data structure must be re-adapted for a better definition and allocation of each cell, node or face.

As an example, consider a computational domain composed of 8 cells, as shown in Fig. 8.10. Each cell is uniquely defined in the global domain with the index i_g so that $i_g \in [1, 8]$ for the internal cells. Note that the cells are not ordered within the computational domain. If the code is executed with 2 processors, a local index i_l must be defined to identify each cell within a certain portion of the domain, so that $i_l \in [1, 4]$ for the internal cells of both the processors. For this purpose, the master rank (processor 0) reads the original global information from the mesh file and re-adapts the indices of its internal cells. Moreover, it scatters the information over all the other processors (in the case of Fig. 8.10, only processor 1). The latter stores the local indices following the receiving order. For example, the cell with global index $i_g = 2$ is the first cell that processor 0 sends to processor 1, so that it is defined with $i_l = 1$ within the portion of processor 1.

At this point, only the indices of the cells have been updated, but the information about the neighbors still follow the global indices and must be re-adapted. Consider the same cell ($i_g = 2$ and $i_l = 1$). Its left neighbor is the cell with global index $i_g = 6$, and this is how this information is stored at this stage (namely `CELLS(1).neigh_left[0]=6`). However, this index does not correspond to the local one. For this reason, each processor updates the information about the neighbors by exploiting the global indices. In the specific example, processor 1 looks for the cell defined by $i_g = 6$: its local index i_l will substitute the old information about the left neighbor, so that `CELLS(1).neigh_left[0]=3`.

$i_g = 7$ $i_l = 4$	$i_g = 5$ $i_l = 2$	$i_g = 1$ $i_l = 1$	$i_g = 8$ $i_l = 4$	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="display: flex; align-items: center; margin-bottom: 5px;"> Proc 0 </div> <div style="display: flex; align-items: center;"> Proc 1 </div> </div>
$i_g = 6$ $i_l = 3$	$i_g = 2$ $i_l = 1$	$i_g = 4$ $i_l = 3$	$i_g = 3$ $i_l = 2$	

Figure 8.10: Example of data structure re-adaptation with 2 processors.

Once all the information have been re-adapted, the ghost cells can be defined. For the domain boundaries, the procedure is the same of that illustrated in section 8.2.3, exploiting the fact that the boundary cells have (at least) one neighbor with index 0 (thus, no neighbor). For processor boundaries, this is no longer valid, as the cells have a non-null index neighbor.

As an example, consider the aforementioned cell ($i_g = 2$ and $i_l = 1$). Its right neighbor belongs to the other processor and its index is not 0. Hence, a different strategy must be applied to recognize the cells at the boundary between

adjacent processors. As reported in appendix D, each cell is characterized by the integer `rank`, representing the belonging processor. This integer is assigned by the master rank in the pre-processing stage. To identify a cell at the boundary between adjacent processors, its rank is compared to that of its neighbors. If the ranks are different, the cell is at the boundary between adjacent processors and a ghost cell must be allocated. As presented in section 8.2.3, the bottleneck of this procedure is the number of allocated ghost cells. A second order discretization needs two ghost cells, but this is a complicated task due to the local refinement. For this reason, only one ghost cell is allocated and an effective strategy is adopted to ensure second order accuracy. Indeed, a second order MUSCL reconstruction can be performed knowing the values of the solution in two consecutive cells or knowing the solution in one cell and the gradient on the face of interest of the same cell, as shown in Fig. 8.5(b). This second way to proceed is suitable for the present formulation. So, an array containing the gradients on the faces of each cell is allocated. In this way, the processors can send and/or receive the information about the ghost cells in terms of local solution (conservative variables in the cell center) and gradients on the faces.

Chapter 9

Results

9.1 Perfect gas hypersonic flow over a compression ramp

The first simulation performed with the immersed boundary approach is the hypersonic flow over a 15° compression ramp. This configuration is quite easy to simulate since the curvature of the body is not relevant as in the case of a bluff body. This does not mean that the reconstruction is easier, but for sure it will be less affected by spurious oscillations, typical of the IBM.

This test case is widely known in the literature [58, 181, 183] as it involves shock wave/boundary layer interaction at a high Mach number ($M_\infty = 11.63$). The low values of the free stream thermodynamic quantities ($p_\infty = 25$ Pa and $T_\infty = 67$ K) makes the assumption of perfect gas reasonable. Specifically, the mixture considered is composed of 76.7% of molecular nitrogen and 23.3% of molecular oxygen. The wall is isothermal ($T_w = 294$ K) and non-catalytic. From the experiments [194], the flow is expected to be fully laminar ($Re_\infty = 3.3 \times 10^5$). In order to provide an adequate comparison of the numerical solution with the one reported in [58], a wall resolution of 2×10^{-5} m was set for the simulations, retained sufficient to capture the boundary layer and the separation near the deflection. The total axial length of the geometry is 0.7 m, with 0.4394 m of flat plate. The contours of the Mach number and temperature are shown in Fig. 9.1, where it is evident that the temperature is not large enough to induce non-equilibrium phenomena. Also, the formation of a separation bubble near the ramp is observed. Such a separation can be identified also looking at the wall quantity profiles, shown in Fig. 9.2. Specifically, the pressure coefficient, skin friction coefficient and Stanton number have been extracted from the flow field to describe the distribution of pressure, friction and heat flux on the surface. These coefficients are computed as follows:

$$C_p = \frac{p - p_\infty}{0.5 \rho_\infty u_\infty^2} \quad (9.1a)$$

$$C_f = \frac{\tau_w}{0.5 \rho_\infty u_\infty^2} \quad (9.1b)$$

$$St = \frac{q_w}{0.5 \rho_\infty u_\infty c_p (T_0 - T_w)} \quad (9.1c)$$

where the subscript ∞ indicates free stream quantities, whereas τ_w is the wall shear stress computed as $\tau_w = \mu \partial U_t / \partial n$, being U_t the tangential component of the velocity. The expression of the Stanton number follows the definition given in [58]; c_p is the constant pressure specific heat, q_w is the wall heat flux, whereas T_0 and T_w are the total free stream temperature and the wall temperature, respectively.

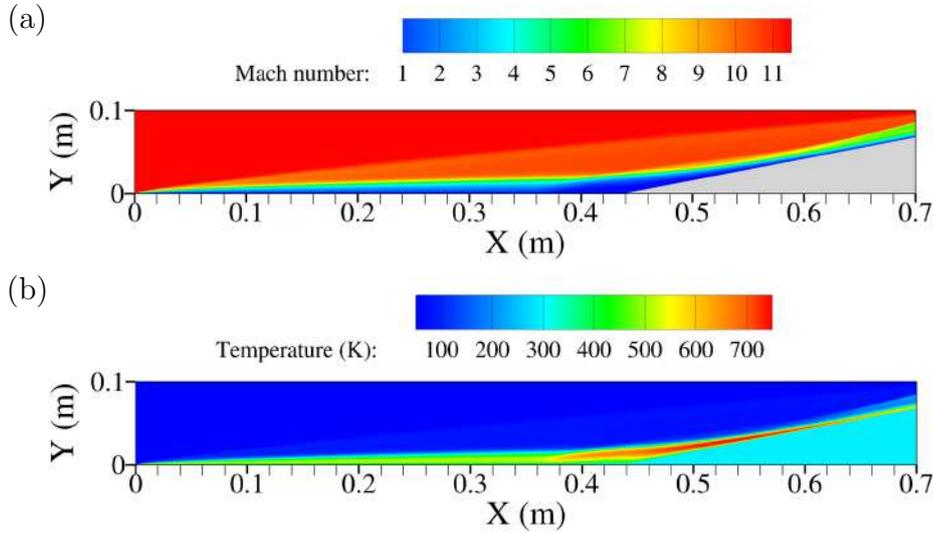


Figure 9.1: Contour of the Mach number (a) and temperature (b) for the compression ramp case.

The wall quantity distributions obtained through the present simulation are in a very good agreement with experiments [194]. In particular, analyzing the skin friction coefficient and the Stanton number, it can be assessed that the separation extent is well predicted. Slight deviations emerge in the calculation of the pressure coefficient. Nevertheless, it is hard to explain the presence of that single measurement point in the upstream region ($x \approx 0.28$ m), as it is a singular value. Although this, the pressure profile values along the ramp are underestimated by the numerical solvers if compared to experimental measurements, as also found by McQuaid et al. [183], but still in agreement.

9.2 Perfect gas hypersonic flow past a cylinder

Space vehicles normally present a bluff geometry. An example has been already illustrated in Fig. 8.1 where a re-entry capsule is shown. In order to verify and/or

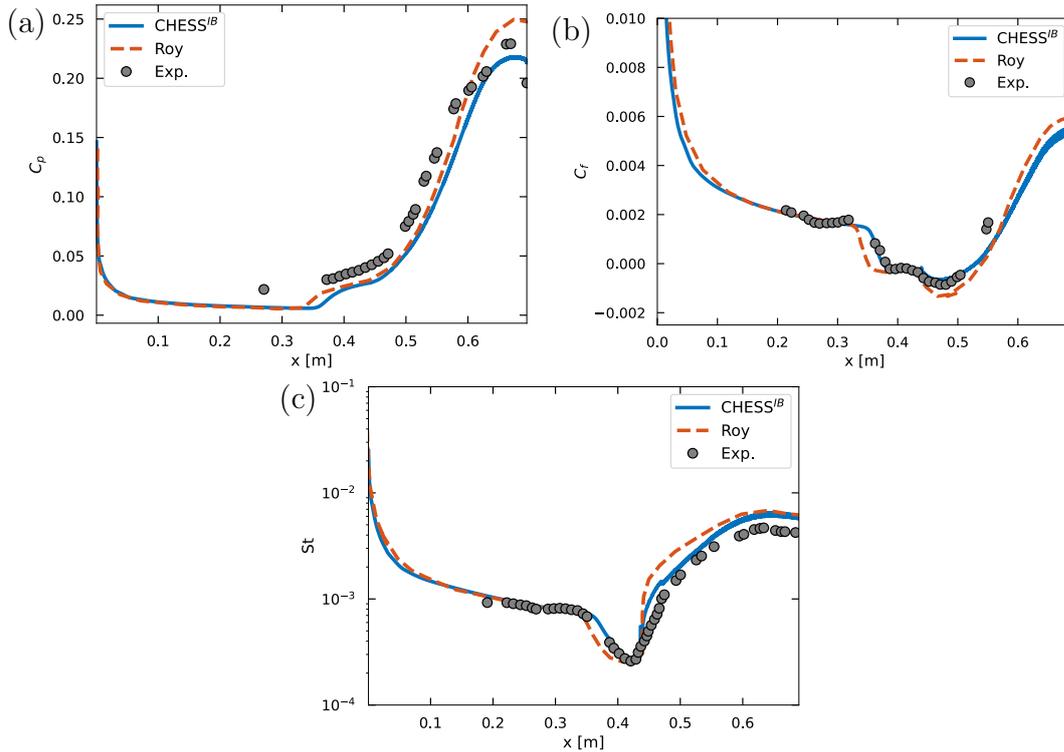


Figure 9.2: Wall quantity profiles for the ramp case: pressure coefficient (a), skin friction coefficient (b), Stanton number (c).

validate a CFD code, simplifications of such geometries can be considered. Cylinders or spheres are often object of numerical simulations since the flow properties are similar to those characterizing the flow around real shape such as a re-entry capsule. In this section, hypersonic flows around a cylinder are analyzed. Specifically, two different test cases are investigated, with corresponding characteristics summarized in table 9.1. The two cases are referred to as *Mach3-Cylinder* (M3C) and *Mach8-Cylinder* (M8C). In both cases, the flow is laminar [164] and the wall is assumed isothermal and non-catalytic. Moreover, the mixture is considered as perfect gas. It has to be stressed that the M3C case is just a numerical test [195], with the goal of a code-to-code verification between the body-fitted (BF) and immersed boundary (IB) solvers. On the other hand, the M8C test case is widely known in the literature [58, 182, 184] as it has been investigated experimentally [196] and can be used to validate the IB solver.

9.2.1 Mach 3 flow (M3C)

The main purpose of the present analysis is the comparison between the results obtained through the IB and BF solvers, provided of similar wall resolutions. To ensure a correct reconstruction of the wall gradients, wall resolutions of $4 \times 10^{-5} m$ and $3 \times 10^{-6} m$ were set for CHESSE^{BF}. The simulations provided the same result,

Table 9.1: Flow conditions for the M3C and M8C test cases.

Quantity	M3C	M8C
M_∞	3	8
Re_∞	246	1.72×10^5
h_0 [MJ/kg]	0.83	1.74
T_∞ [K]	293	124
p_∞ [Pa]	7.43	831.5
Y_{N_2}	0.767	0.767
Y_{O_2}	0.233	0.233
T_w [K]	293	294

as reported in the following. Hence, a resolution of $2 \times 10^{-5} m$ was chosen for CHES^{IB}, being a good compromise to reduce the computational cost and, at the same time, ensure accuracy near the wall. However, a coarser mesh was employed for the IB simulation to analyze the sensitivity of the spurious oscillations, typical of the immersed boundary methods [51–53, 182]. It turned out that wall quantity distributions are still affected by numerical oscillations even on the finer grid. The results in terms of temperature and stream-wise velocity profiles along the stagnation streamline are reported in Fig. 9.3. The normalized shock stand-off distance detected by the IB solver is in perfect agreement with that captured by the BF solver. Also, the gradients in the boundary layer are well reconstructed. As a consequence, the IB solver is able to well describe the wall quantity profiles, known to be the most critical issue of such a methodology. Distributions of pressure coefficient, skin friction coefficient and heat flux are shown in Fig. 9.4, where they are compared to those obtained from the BF simulations. The agreement is satisfactory, confirming the correct implementation of the methodology. Nevertheless, spurious oscillations appear also when employing a finer grid. It is complex to find an exclusive explanation to this behavior of the numerical solution, probably result of a combination of aspects. Wall resolution is not the only reason promoting these oscillations: the discretization of the Lagrangian grid could be an additional issue. Indeed, a higher quality of the STL file could improve the calculation of wall normal projection of the probes and, as a result, the estimation of the wall quantities.

9.2.2 Mach 8 flow (M8C)

Similarly to the M3C flow, the assumption of perfect gas is still valid, but the Mach number is higher, making the free stream total enthalpy twice higher than the M3C test. These conditions, summarized in table 9.1, make the M8C a more challenging test with respect to the M3C case.

This test is widely developed in literature, as many authors performed nu-

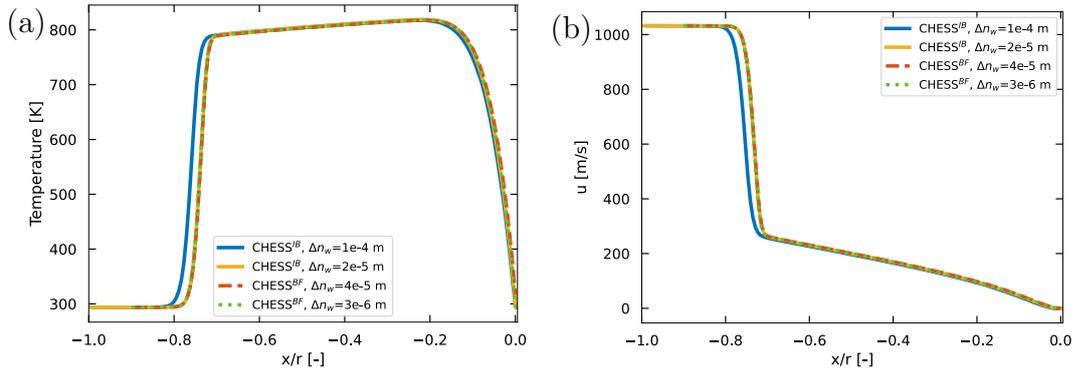


Figure 9.3: Profile of temperature (a) and stream-wise velocity (b) along the stagnation line for the M3C case.

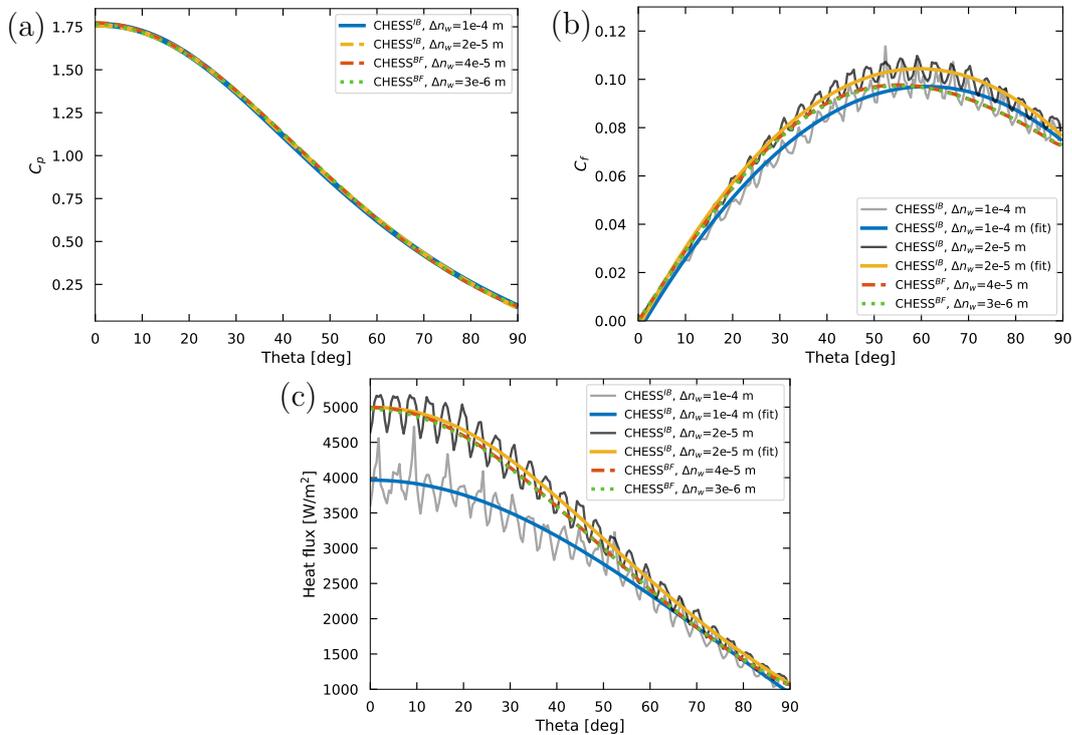


Figure 9.4: Wall quantities profiles for the M3C case: pressure coefficient (a), skin friction coefficient (b), heat flux (c).

merical investigations to assess the accuracy of their computational codes. The most critical issue is the calculation of the heat flux on the surface. It turned out that a classical IB approach was not suitable to well capture the thermal boundary layer at wall, thus leading to a relevant under estimation of the heat transfer [58,181,182]. Recent efforts have been devoted by McQuaid et al. [183] to overcome such an issue. Their aim was twofold: first of all, they demonstrated that an hybrid approach improves the reconstruction of the boundary layer. Indeed, the

employed computational grids presented a Cartesian topology in the far field, but they were conformal to the body near the wall. Secondly, a third order space discretization allowed to obtain satisfactory results also on coarse grids. Nevertheless, this methodology presents an important drawback. The use of a body conformal mesh near the wall could imply an increase of the computational cost. Moreover, in the presence of relevant ablative phenomena, the handling of the wall recession can be very challenging. For these reasons, the aim of the present analysis is to try to correctly calculate the heat transfer on the surface when employing classical IB methods. Furthermore, a possible motivation of its under estimation is proposed.

The results presented in [58] are not satisfactory, but it is noteworthy that, given the high Mach number and the presence of a bow shock, the wall gradients are very sharp and they need an opportune wall resolution. Hence, a preliminary analysis has been carried out by means of the BF code (CHESS^{BF}) to estimate a suitable wall resolution. The results are shown in Fig. 9.5. The shock stand-off distance is well predicted in all the cases, being in agreement with the expected one [197]. The distribution of the pressure on the surface is practically the same for the three resolutions and is in agreement with the experimental measurements [196], except for the stagnation point where it is slightly under estimated by the numerical solver. Concerning the heat flux, it is worth underlining that the results are not grid converged and this would deserve further investigations. Nevertheless, the goal of this analysis is to provide a comparison of the BF and IB results for the same wall resolution. A finer mesh would start to be prohibitive for an IB application, since the aspect ratio of the cells near the body should be around 1. For this reason, wall resolutions of $8 \times 10^{-6} m$ and $4 \times 10^{-6} m$ are employed for the IB simulations.

The pressure distribution over the surface of the cylinder is shown in Fig. 9.6. The results obtained with the IB simulations are in a very good agreement with those reported by the BF approach. Specifically, the coarser grid presents spurious oscillations around the maximum curvature of the wall (40 – 60 deg) due to the linear interpolation, but this follows the expectations. By the way, an higher wall resolution leads to a relevant improvement of the quality of the solution, that is smoother and with no oscillations. Despite this satisfactory result, it is known that, given the nature of the boundary condition on the pressure (null gradient), this calculation is straightforward.

On the other hand, the evaluation of the heat flux represents a bottleneck for these kind of applications. The mathematical expression of the heat flux is based on the temperature gradient, namely the slope of the temperature profile in the boundary layer. Hence, one can analyze the temperature profile along the stagnation line, focusing on the stagnation point region. These outcomes are illustrated in Fig. 9.7. The shock stand-off distance is well captured by the IB solver, even if the discontinuity is not as sharp as that provided by the BF solver, due to a coarsening of the mesh in the shock region. Nevertheless, the temperature peak in the shock layer is perfectly predicted by the IB solver and the error in

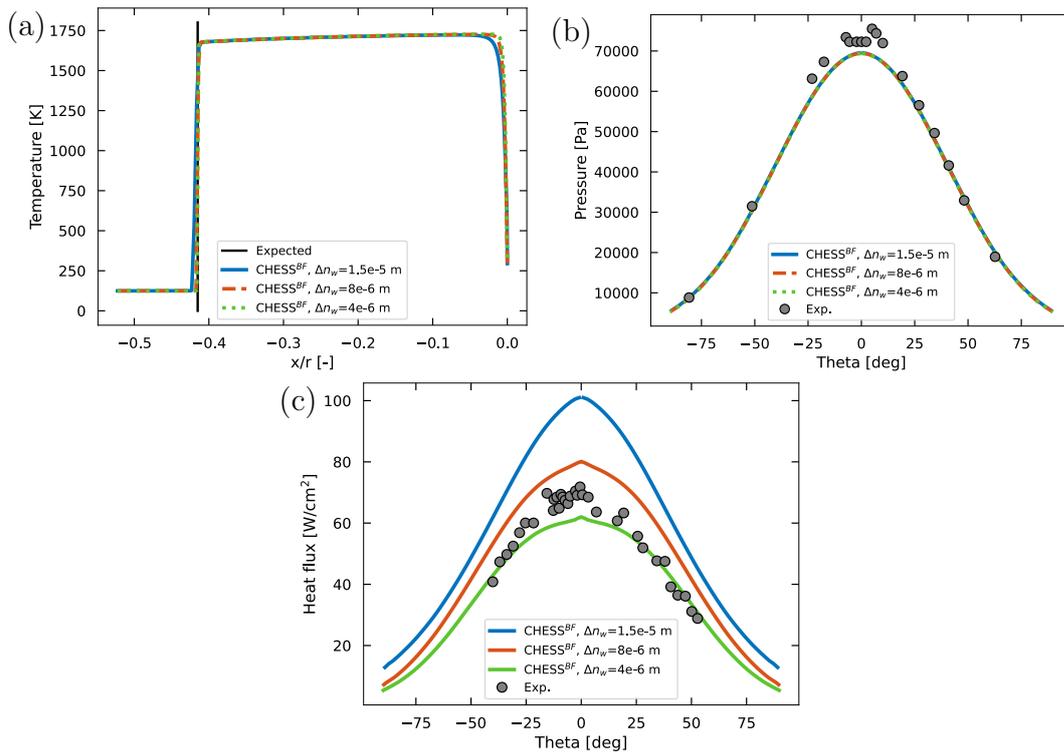


Figure 9.5: Temperature profile along the stagnation streamline (a), wall pressure (b) and wall heat flux (c) for the M8C case.

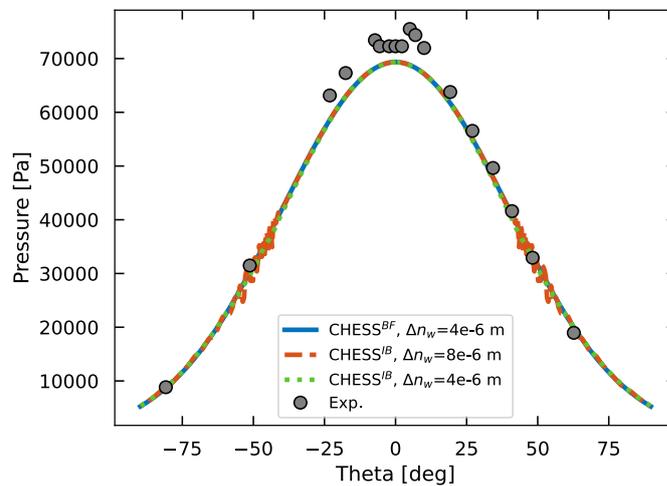


Figure 9.6: Pressure distribution over the surface for the IB simulations of the M8C case.

the evaluation of the shock stand-off distance is about 1%. On the other hand, the temperature gradient near the stagnation point strongly deviates from that provided by the BF approach. The non-conservative nature of the scheme leads to a relevant change in the slope of the curve near the wall, namely a different

temperature gradient. The corresponding heat fluxes are shown in Fig. 9.8.

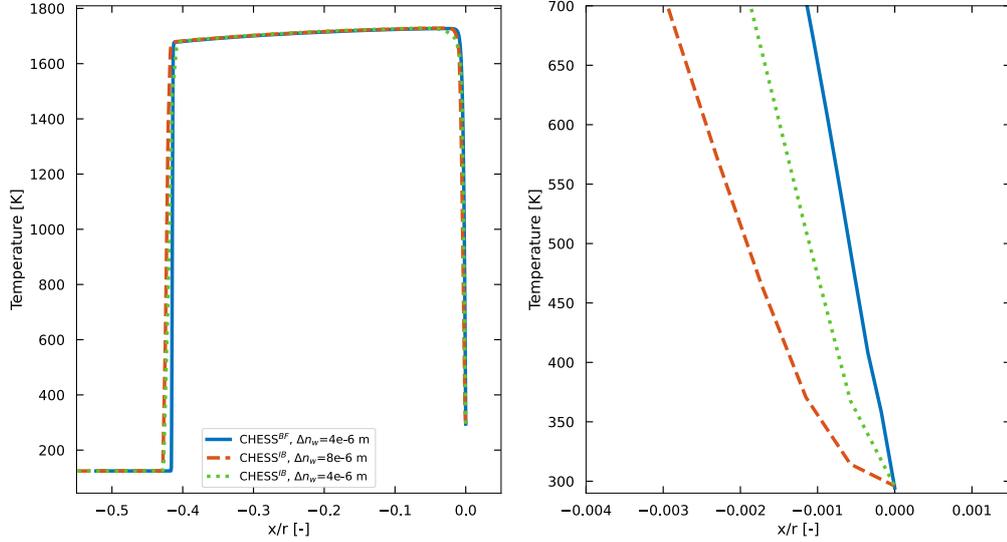


Figure 9.7: Temperature profile along the stagnation streamline for the IB simulations of the MSC case (left). Zoom near the stagnation point (right).

The orange and the black curves are the heat fluxes obtained from the IB simulations with a wall resolution of $8 \times 10^{-6} m$. In this case, `MUTATION++` helps in increasing the heat flux, but no important differences are found with respect to the standalone version of the code. The green and the yellow curves are the heat fluxes obtained with the finer grid (wall resolution of $4 \times 10^{-6} m$). It is noteworthy that the position of the probe points plays a key role in the evaluation of the wall gradients. Indeed, the heat flux decreases when the probe points are closer to the body. This is coherent with the temperature profiles shown in Fig. 9.7: the error introduced by the linear interpolation propagates in a certain range of cells in the vicinity of the wall, depending on the local resolution. Hence, an interpolation performed far from the wall (fluid region) can be much more accurate, as the solution does not depend on the reconstruction. Finally, one has to point out that the problem in such a case is not only linked to the value of the heat flux, but also to its profile. For a cylindrical geometry, the stagnation point is characterized by the maximum value of pressure and heat transfer, making the present results non-physical.

In conclusion, two main points can be addressed.

- The drawback in the calculation of the heat flux is not (only) linked to a grid convergence analysis in the present case, as the results are quite similar for different wall resolutions. The solution seems more sensitive to the position of the probe points.
- The wall resolution is not the only reason of the incorrect evaluation of the wall gradients, as for the BF and the IB solvers provide completely different

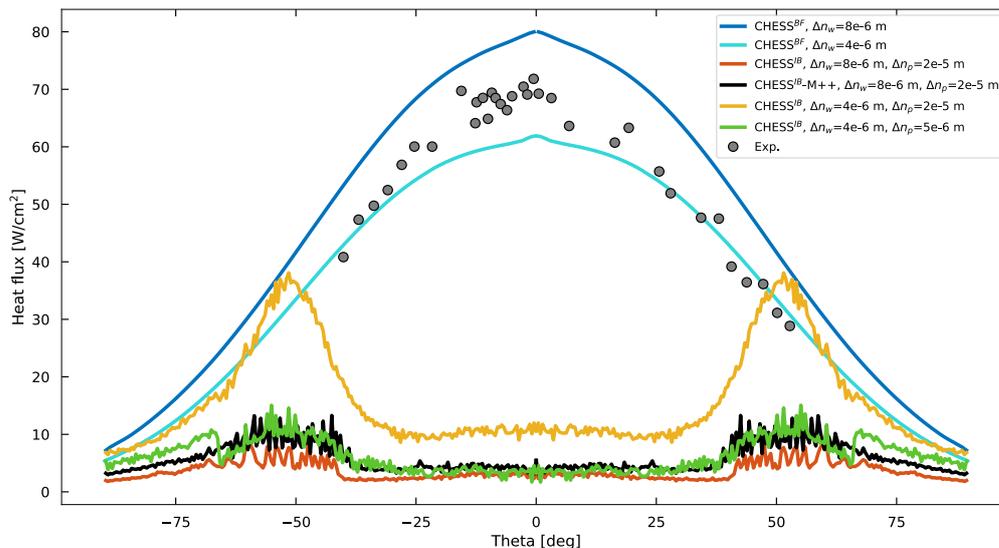


Figure 9.8: Surface heat flux for the IB simulations of the M8C case.

results for same wall resolutions.

9.3 Reactive hypersonic flow past a cylinder

The results shown in the previous sections underline important aspect of the IB method. Nevertheless, these analysis are limited to perfect gas mixtures. The aim of this section is to extend the IB approach to non-equilibrium flows, including gas surface interactions. For this purpose, the same test case proposed in section 5.3 is here investigated [30].

From the analysis of the M8C flow, it has been assessed that the most critical issue of the present IB method is the evaluation of the temperature gradients at wall. As shown in section 5.3, the Knight case presents an extreme heat load in the thermal boundary layer. The numerical error introduced by the linear interpolation in the interface cells can propagate in far field, affecting the shock stand-off distance. For this reason, three cases have been analyzed [166]: in the first one the wall is assumed adiabatic, an interesting configuration to verify the correct implementation of the IB algorithm, accounting for thermochemical non-equilibrium. Such an assumption implies a null temperature gradient at wall, thus making the shock stand-off distance independent from the thermal boundary layer. The other two cases are the same presented in section 5.3, with isothermal wall assumption, to assess the influence of catalysis on the heat flux. In all the cases, thermal equilibrium is assumed.

The results obtained with CHESSt^{IB} are compared to those obtained with the INCA solver [166], provided of similar wall resolutions. INCA is an accurate solver for hypersonic flows, that exploits a *cut-cell* strategy, thus being conservative also near the wall. Gas-surface interactions are accounted for by coupling the

solver with MUTATION⁺⁺. It is noteworthy that the results in the following are obtained through the standalone version of CHES^{IB}. Fig. 9.9 shows the profiles of temperature and species mass fractions along the stagnation streamline for the adiabatic case. The wall resolutions are $5 \times 10^{-8} m$ for US3D, $5 \times 10^{-6} m$ for INCA and $7 \times 10^{-6} m$ for CHES^{IB}. The three codes provide results in a perfect agreement, confirming both the correct implementation of the algorithm and the correct treatment of non-equilibrium.

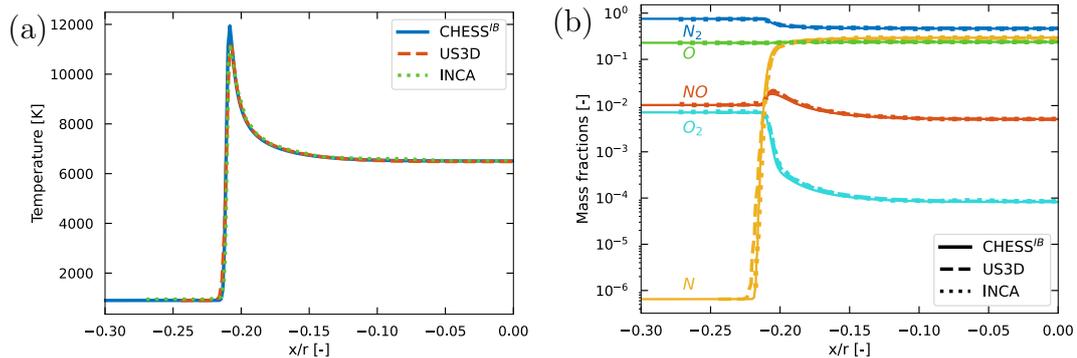


Figure 9.9: Temperature (a) and species mass fractions (b) profiles along the stagnation streamline for the adiabatic Knight case.

The results for the isothermal wall case are shown in Fig. 9.10 for the non-catalytic configuration. An even finer mesh has been employed to simulate such flow conditions. As expected, the present IB solver is not able to well describe the physics in the boundary layer. Differently from the M8C flow, the error introduced by the forcing involves a wider range of cells: as a consequence, the shock position is detected more in the upstream with respect to US3D and INCA (error about 12%). This is confirmed by the inset shown in Fig. 9.10(b), where it emerges that the temperature gradient is not well reconstructed when a Dirichlet boundary condition is imposed at wall. This induces two main effects: first of all, the heat flux prediction is not correct. Secondly, the behavior of the mass fraction profiles along the stagnation streamline is completely different than the expectations. This is remarkably evident in Fig. 9.10(c), where the mass fraction of N is drastically decreasing as soon as the fluid approaches the wall. This is a consequence of the fact that the temperature value in the boundary layer are extremely lower than the real ones, promoting a different chemical activity.

Finally, wall quantity distributions are reported in Fig. 9.11. As for the M8C case, the agreement in terms of pressure distribution is remarkable, even for the coarser mesh. The numerical simulations reproduce very well the experimental measurements. On the other hand, the agreement in terms of heat flux is not satisfactory. The incorrect reconstruction of wall gradients leads to a drastic under estimation of the heat load over the surface of the cylinder. Of course, catalysis contributes to an increment of the heat flux, but it is not the main issue (indeed, the catalytic simulation was not performed on the finer mesh). Contrarily to the

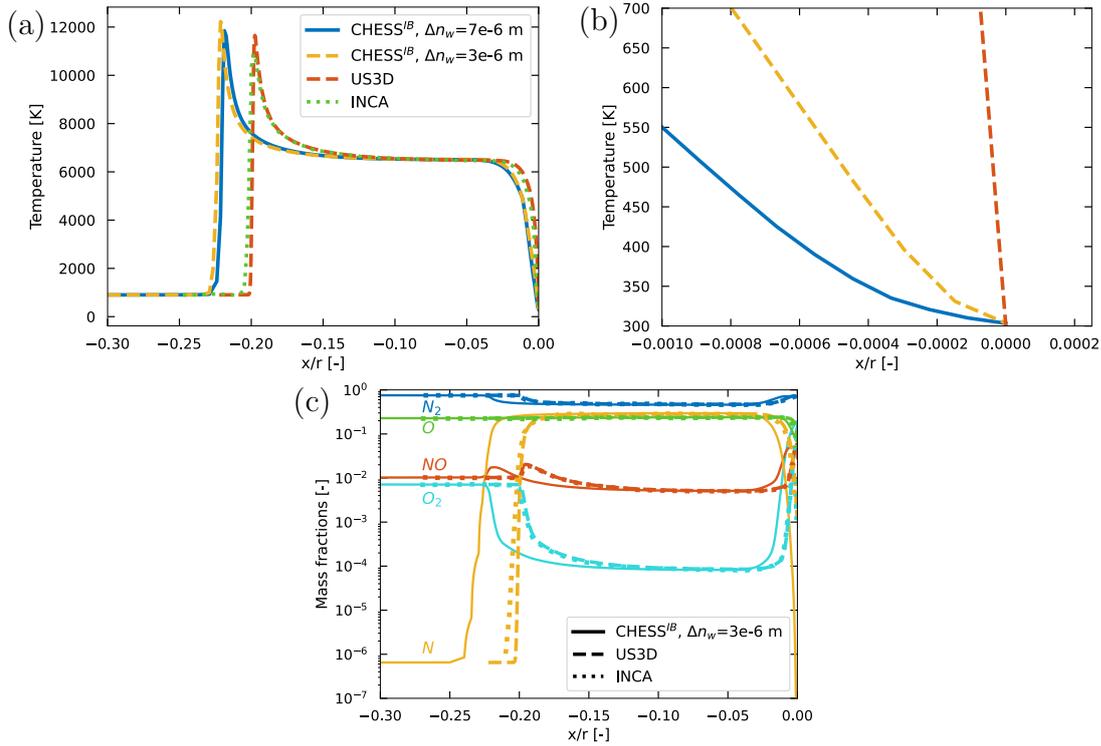


Figure 9.10: Temperature (a) and species mass fraction profiles (c) along the stagnation streamline for the isothermal wall Knight case (non-catalytic). Zoom of the temperature near the wall (b).

M8C case, the gauge temperature is about 20 from the shock layer (≈ 6000 K) to the wall (300 K) and this makes the interpolation error propagates far from the wall, affecting not only the heat flux but also the shock stand-off distance.

It is noteworthy that the aforementioned non-conservative nature of the scheme near the wall leads to a strange behavior of the velocity field. An unexpected recirculation zone forms near the stagnation point when an isothermal wall boundary condition is imposed. Indeed, this is not experienced for the adiabatic case. This aspect is analyzed in detail in appendix F.

9.4 Effects of ablation

In order to test the solver in the presence of ablation, the subsonic case presented in first part of this thesis is here investigated using the IB method. Detailed specifications about this test case are explained in section 5.4. The wall resolution is the same of the BF approach, namely $1 \times 10^{-5} m$.

The profiles of the stream-wise velocity, temperature and species mass fractions along the stagnation streamline are illustrated in Fig. 9.12, where the comparison with the solution from US3D is also provided. The agreement in terms of velocity and temperature is remarkable, as expected because of the subsonic regime.

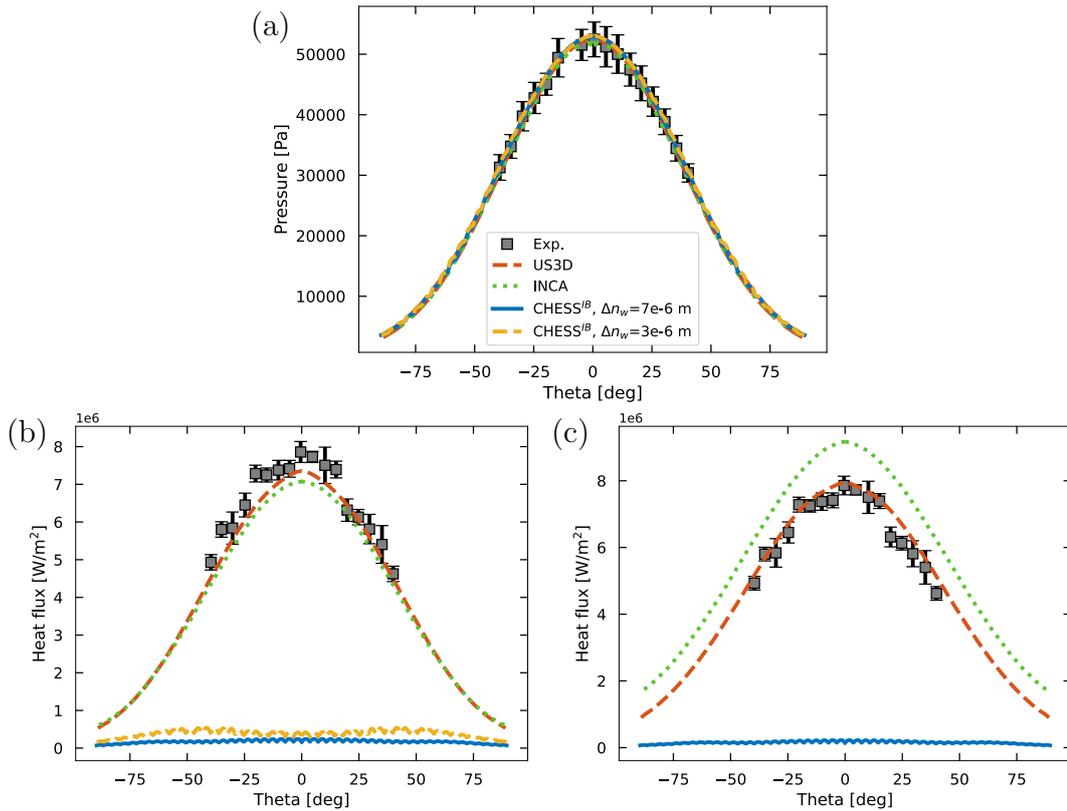


Figure 9.11: Wall pressure distributions over the surface for the non-catalytic case (a). Heat flux over the surface for the non-catalytic (b) and the fully-catalytic (c) cases.

The boundary layer is thick enough to make the wall resolution suitable to well describe the diffusion phenomena near the wall. This is an important outcome as it confirms that the mismatch observed in the cylinder test cases is linked to the small boundary layer thickness typical of high speed flows. However, the mass fraction profiles present a mismatch. Nitrogen recombination due to the low wall temperature ($T_w = 2407$ K) is underestimated by CHESSES^{IB}-M⁺⁺. This is probably provoked by the Dirichlet boundary condition, needed to impose the value of the mass fractions at wall. Indeed, the behavior of the mass fraction gradients is comparable to that observed for the temperature in the cases of the cylinder. The ablative layer is very thin and this leads to an incorrect reconstruction of the gradients at wall. Nevertheless, the estimation of the massblowing rate is in accordance with references. Wall quantity distributions are shown in Fig. 9.13. The wall pressure obtained with CHESSES^{IB}-M⁺⁺ is in agreement with that reported by CHESSES^{BF}-M⁺⁺, with slight differences with respect to US3D, probably attributable to the different thermodynamics model employed (NASA9 for US3D and RRHO for CHESSES). The massblowing rate is over estimated by the present IB method and this is in accordance with the expectations. The lower recombination predicted by CHESSES^{IB}-M⁺⁺ leads to a higher concentration of the atomic nitrogen. As shown

in section 2.5, the source terms arising from the gas-surface interaction is proportional to the number density of a certain species (N in this case). This explains the higher massblowing rate calculated by $\text{CHES}^{\text{IB}}\text{-M}^{++}$.

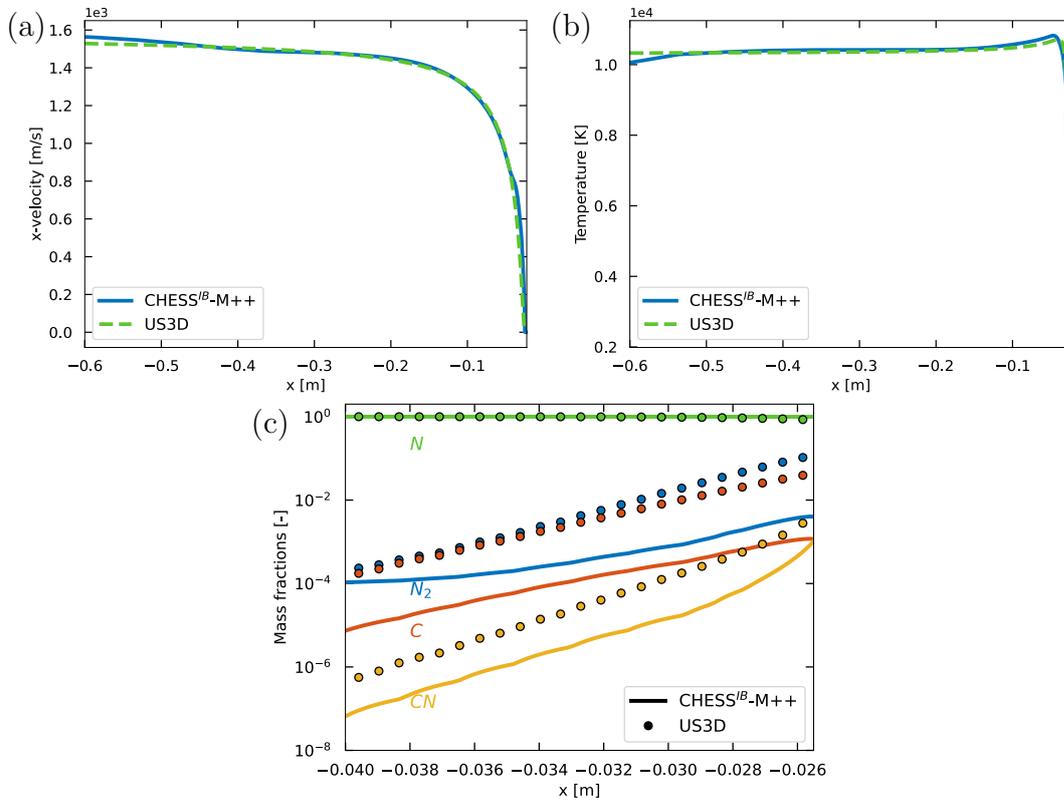


Figure 9.12: Profiles of stream-wise velocity (a), temperature (b) and species mass fractions (c) along the stagnation streamline for the subsonic ablative case.

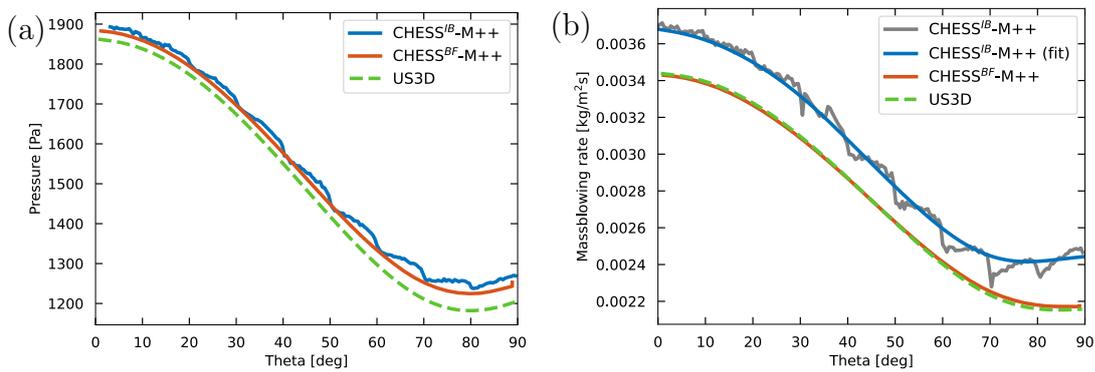


Figure 9.13: Pressure (a) and massblowing rate (b) distributions over the surface for the subsonic ablative case.

Chapter 10

Conclusions and perspectives

In the second part of this thesis, an immersed boundary approach has been applied to solve hypersonic flows in several configurations. The known problems of this kind of method came out, confirming the inaccuracy of the solution when dealing with sharp temperature gradients at wall. Indeed, simulations for the flow past a compression ramp reports very good results, even though the Mach number is very high ($M_\infty = 11$). The same was shown for the flow past a cylinder, for a Mach number of 3: heat flux, pressure and skin friction distributions at wall are satisfactory and in agreement with the expectations.

First problems arise when solving a higher Mach number flow past a cylinder ($M_\infty = 8$ or $M_\infty = 9$). In this case, the heat flux is extremely under predicted by the IB approach with respect to a BF approach and to experimental measurements. Wall resolution is surely an issue, but it can not be the unique responsible of this incorrect calculation. Indeed, the heat flux does not increase as expected when reducing the size of the first cell. Instead, it is more reasonable to think that it is provoked by a combination of the higher Mach number (which promotes sharper gradients near the wall) and the wall resolution, probably leading the an incorrect evaluation of the heat flux. The shock stand-off distance is also affected by the diffusion phenomena in the boundary layer: depending on the relevance of the thermal gradients, the shock can be detected more in the upstream region, underlining the limitations of the IB approach. On the other hand, it is noteworthy that the bow shock is well captured when a zero heat flux is imposed at wall (adiabatic flow). In this configuration, non-equilibrium phenomena are perfectly reproduced by the IB approach (also at wall), strongly inducing to argue that the reconstruction of the temperature is the most critical issue of the algorithm.

A possible solution to figure this out is an alternative formulation of the fluxes in the vicinity of the wall. Di Mascio and Zaghi [198] proposed a similar idea that is applicable to both finite-difference and finite-volume approaches. In their work, they present an *ad hoc* reconstruction of the solution so that the fluxes are reconstructed accordingly to a certain wall boundary condition. This can represent an important improvement, especially for the discretization of the diffusive terms,

responsible for the heat flux. Indeed, the present numerical scheme is second order cell-centered based. However, the nonphysical values of the flow variables in the solid cells can affect the accuracy of the scheme, leading to inaccurate evaluation of the gradients.

Another perspectives is linked to the simulations of flows past bodies with recessing surfaces. In such a case, the ray-tracing procedure must be performed run-time in order to classify the new cells (a cell that is solid at the iteration k can become fluid at the iteration $k + 1$).

Appendix D

Connectivity and data structure information

```
1 #define nof      8    // Nodes of face
2 #define foc     24   // Faces of cell
3 #define noc     54   // Nodes of cell
4 #define mxn     4    // Max neighbours
5
6 typedef struct cell_data_t{
7     int index_c_g, index_c_l;    // Index of the cell (global and
8     local)
9     REAL xc,yc,zc;              // Cell center
10    REAL volume,Sr,St,Sf,dt;     // Volume, surfaces (right, top and
11    front to compute dt) and dt
12    int nodes_of_cell[noc];      // Nodes of the cell
13    int f_left[mxn];             // Face(s) left
14    int f_right[mxn];           // Face(s) right
15    int f_bottom[mxn];          // Face(s) bottom
16    int f_top[mxn];             // Face(s) top
17    int f_back[mxn];            // Face(s) back
18    int f_front[mxn];           // Face(s) front
19    int neigh_left[mxn];        // Neigh(s) left
20    int neigh_right[mxn];       // Neigh(s) right
21    int neigh_bottom[mxn];      // Neigh(s) bottom
22    int neigh_top[mxn];         // Neigh(s) top
23    int neigh_back[mxn];        // Neigh(s) back
24    int neigh_front[mxn];       // Neigh(s) front
25    int left_count, right_count, bottom_count, top_count, back_count,
26    front_count;               // Number of neighbours of the cell
27    REAL min_x,max_x;           // Min and max value of x
28    REAL min_y,max_y;           // Min and max value of y
29    REAL min_z,max_z;           // Min and max value of z
30    int n_brick[8];             // Nodes of the brick (plotting
31    purposes)
32    int flag;                    // Flag ( -1 solid, 1 interface, 0
33    fluid)
34    REAL norm_proj[3];          // Wall normal projection
```

```

30     REAL wall_norm[3];           // Wall normal vector
31     int is_boundary[6];         // 0 LEFT 1 RIGHT 2 BOTTOM 3 TOP 4
    BACK 5 FRONT (0 = NO BOUNDARY, 1 = BOUNDARY) – DOMAIN
32     int rank;                   // Useful to find cells at
    processors boundaries
33 } cell_data;
34
35 typedef struct face_data_t{
36     int index_f_g, index_f_l;   // Index of the face (global and
    local)
37     int contiguous_cells[2];    // Cells sharing the face
38     int nodes_of_face[nof];    // Nodes of the face
39     REAL nx,ny,nz,S;           // Normal components and surface
40     int nodes_gauss[4];        // Nodes for Gauss theorem (viscous
    fluxes)
41     int is_boundary_proc;      // Determine if it is on a boundary
    between processors (0 NO 1 YES)
42 } face_data;
43
44 typedef struct node_data_t{
45     int index_n_g, index_n_l;   // Index of the node (global and
    local)
46     REAL xn,yn,zn;             // Coordinates of the node
47     int shared_by_cells[8];    // Cells sharing the node
48     int cells_count;          // Dummy for computations
49 } node_data;

```

Listing D.1: Struct variables for the data structure

Appendix E

Segment/triangle intersection

The *ray-tracing* procedure explained in section 8.2.2 is based on the intersection between a segment and a surface, specifically a triangle. Indicating with $\mathbf{C} = (x_C, y_C, z_C)$ the control point (far from the body) and with $\mathbf{Q} = (x_Q, y_Q, z_Q)$ the query point, the goal is to find if a point $\mathbf{P} = (x_P, y_P, z_P)$ simultaneously belongs to the ray (segment) $\mathbf{R} = (x_Q - x_C, y_Q - y_C, z_Q - z_C)$ and to a given triangle given by three vertices $\mathbf{V}_1, \mathbf{V}_2$, and \mathbf{V}_3 .

A generic point \mathbf{P} belonging to \mathbf{R} can be defined through a parameter $t \in [0, 1]$ such that $\mathbf{P} = \mathbf{C} + t\mathbf{R}$, thus $\mathbf{P} \equiv \mathbf{C}$ for $t = 0$ and $\mathbf{P} \equiv \mathbf{Q}$ for $t = 1$. In order to check if the ray \mathbf{R} intercepts a triangle, it is convenient to check for the intersection with the plane containing the triangle and given by the equation $ax + by + cz = d$, where $\mathbf{n} = (a, b, c)$ is the plane normal vector (this is not the unit vector) and (x, y, z) are the spatial coordinates of a point belonging to the triangle. The cross product between two edges of the triangle provides \mathbf{n} , namely the coefficients a, b and c . The remaining coefficient d can be directly computed by substituting the coordinates of a generic vertex of the triangle in the equation of the plane. Indeed, considering for instance the vertex \mathbf{V}_1 , the equation of the plane is recovered by imposing the condition $\mathbf{V}_1 \cdot \mathbf{n} = d$. Once all the coefficients a, b, c and d are known, the same condition can be applied to a generic point \mathbf{P} to check for the ray/plane intersection. Substituting the definition of \mathbf{P} yields:

$$(\mathbf{C} + t\mathbf{R}) \cdot \mathbf{n} = d \Rightarrow t = \frac{d - \mathbf{P} \cdot \mathbf{n}}{\mathbf{R} \cdot \mathbf{n}} \quad (\text{E.1})$$

The last task is to check if the so found point \mathbf{P} lies inside the triangle. For this purpose, one can consider Fig. E.1. Recalling that \mathbf{n} is the normal vector of the triangle $V_1V_2V_3$, if the point belongs to it the dot product between \mathbf{n} and the normal vector of each inner triangle is positive, provided that they are computed coherently. In practice, if they all points to the same direction \mathbf{P} is inside the triangle, as in Fig. E.1a. This is not the case if \mathbf{P} is outside the triangle (see Fig. E.1b). In this situation, at least one normal vector among the inner triangles points in the opposite direction with respect to \mathbf{n} .

Singularities might occur when \mathbf{P} lies on a certain edge of the triangle or it

corresponds to a certain vertex. In such cases, at least one normal vector is null and the algorithm must be able to handle these conditions.

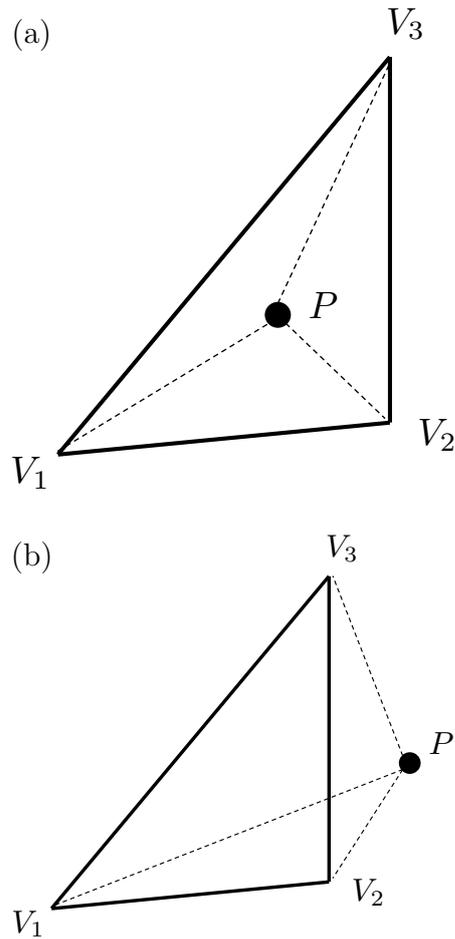


Figure E.1: Different configurations for the ray/plane intersection.

Appendix F

Influence of temperature gradients on the velocity field

When dealing with flows past a cylinder as the cases in sections 9.2 and 9.3, an interesting aspect is observed in the velocity field when the wall is assumed isothermal. A recirculation region forms in the vicinity of the stagnation point, as it is observed in Fig. F.1 (for the *Knight case*, the non-catalytic test is considered). When approaching the wall, the stream-wise velocity component assumes a negative sign, that is unexpected given the symmetry of the body. Moreover, it can appear over the surface for low wall resolutions.

Such a behavior occurs only when a Dirichlet boundary condition is imposed on the temperature. Indeed, the velocity never becomes negative in the adiabatic case. Fig. F.2 reports the profiles of u/u_∞ for the adiabatic *Knight case*. The stream-wise velocity component becomes null on the wall, preserving the correct direction along the stagnation streamline. It deserves to be stressed that, in such a case, the normalized shock stand-off distance detected by CHES^{IB} is correct, as already observed in section 9.3.

This observation induces to think that the temperature and the velocity could be linked in the presence of temperature gradients, implicating an incorrect prediction of the heat flux. Nevertheless, as one can notice in Fig. F.1(b), the values of the negative velocities are just small percentages of the free stream value and they should not strongly affect the flow field. As a matter of fact, in the M3C case the heat flux is well reconstructed (see 9.2.1) even in presence of a recirculation region. This aspect deserves further investigations.

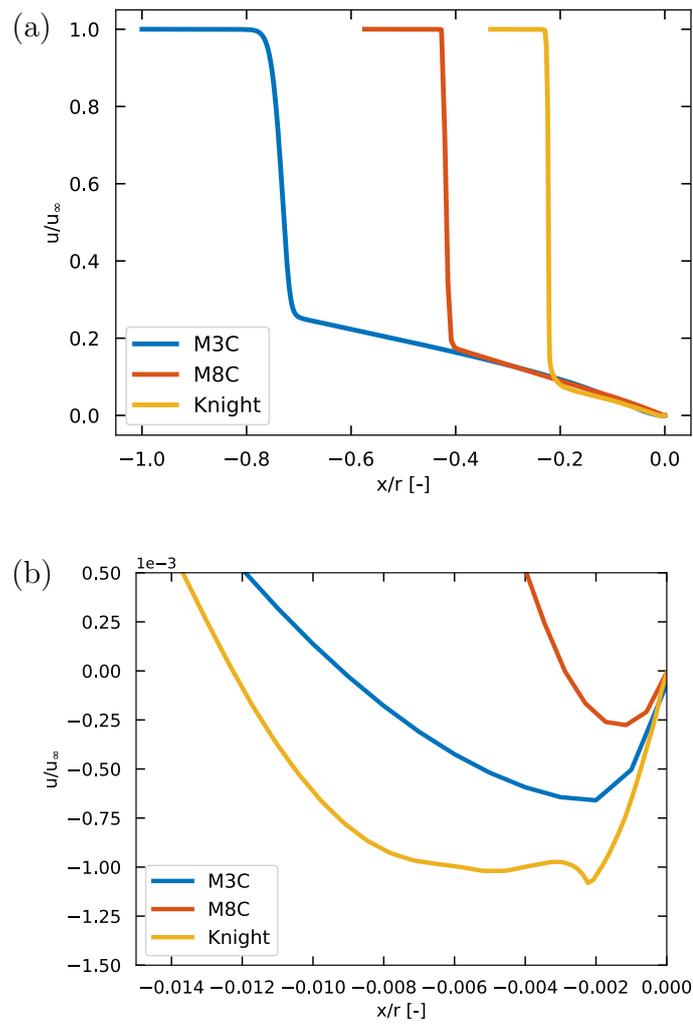


Figure F.1: Profiles of u/u_∞ along the stagnation streamline for the three isothermal cylinder cases (a). Zoom near the stagnation point (b).

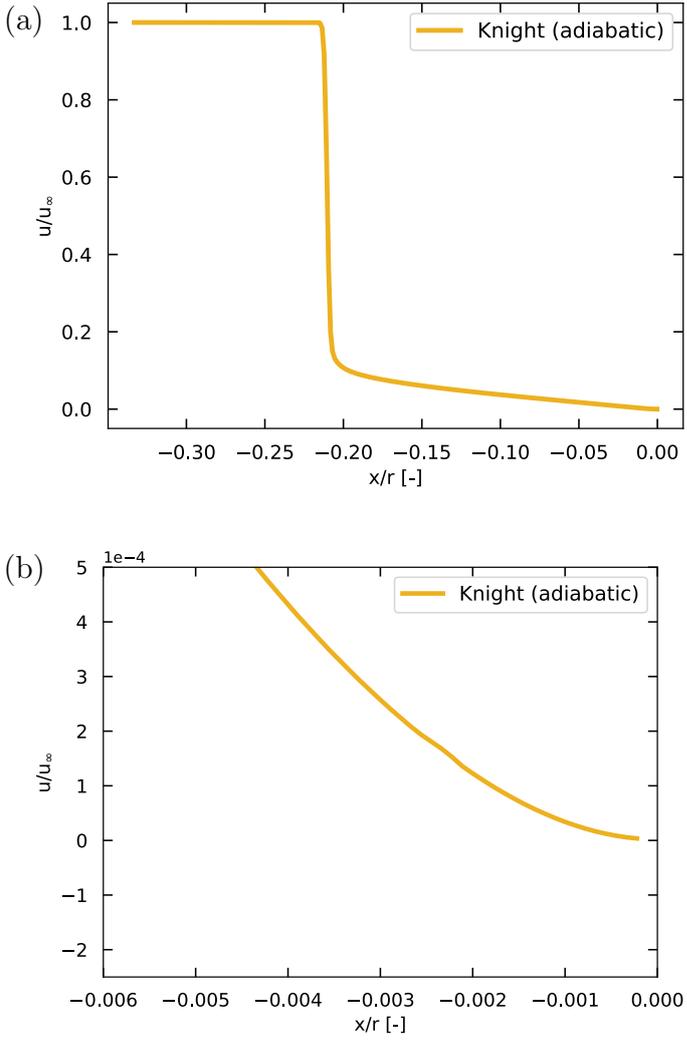


Figure F.2: Profiles of u/u_∞ along the stagnation streamline for adiabatic *knight* case (a). Zoom near the stagnation point (b).

Appendix G

Gantt chart of the project

The project started in November 2019 and the first year was mainly based on a preliminary phase devoted to the literature review and to the training on C/CUDA programming and high-performance computing (HPC). This allowed to fully understand the problem and the to establish the main goals of the project. Preliminary simulations were performed by the end of the first year to frame the main issues of the hypersonic flows and of the numerical method employed.

During the second year, a numerical investigation on double-wedge flows were carried out. It led to the publication of a journal paper [23] at the end of the second academic year. Moreover, the 3D version of the original body-fitted code was developed and verification studies with the 2D formulation were performed. This represented the starting point of the development of the Immersed Boundary version, terminated by the first half of the second academic year.

The first part of the third year took place at the von Karman Institute for Fluid Dynamics (VKI) in Belgium. During the collaboration, the code was extended to advanced algorithms to deal with gas-surface interactions. This visiting period allowed also to publish a conference paper for the American Institute of Aeronautics and Astronautics (AIAA AVIATION, 2022) conference [166], where first Immersed Boundary results were presented. Nevertheless, they still need improvement, as shown in this manuscript. Finally, a numerical investigation of double-cone flows was also carried out, representing an extension of the work carried out during the second year. Results of the investigation were published by the end of the third year [156].

During the PhD, academic courses about programming, modeling and turbulence were also attended, along with the participation to many conferences. The complete Gantt chart is shown in Fig. G.1.

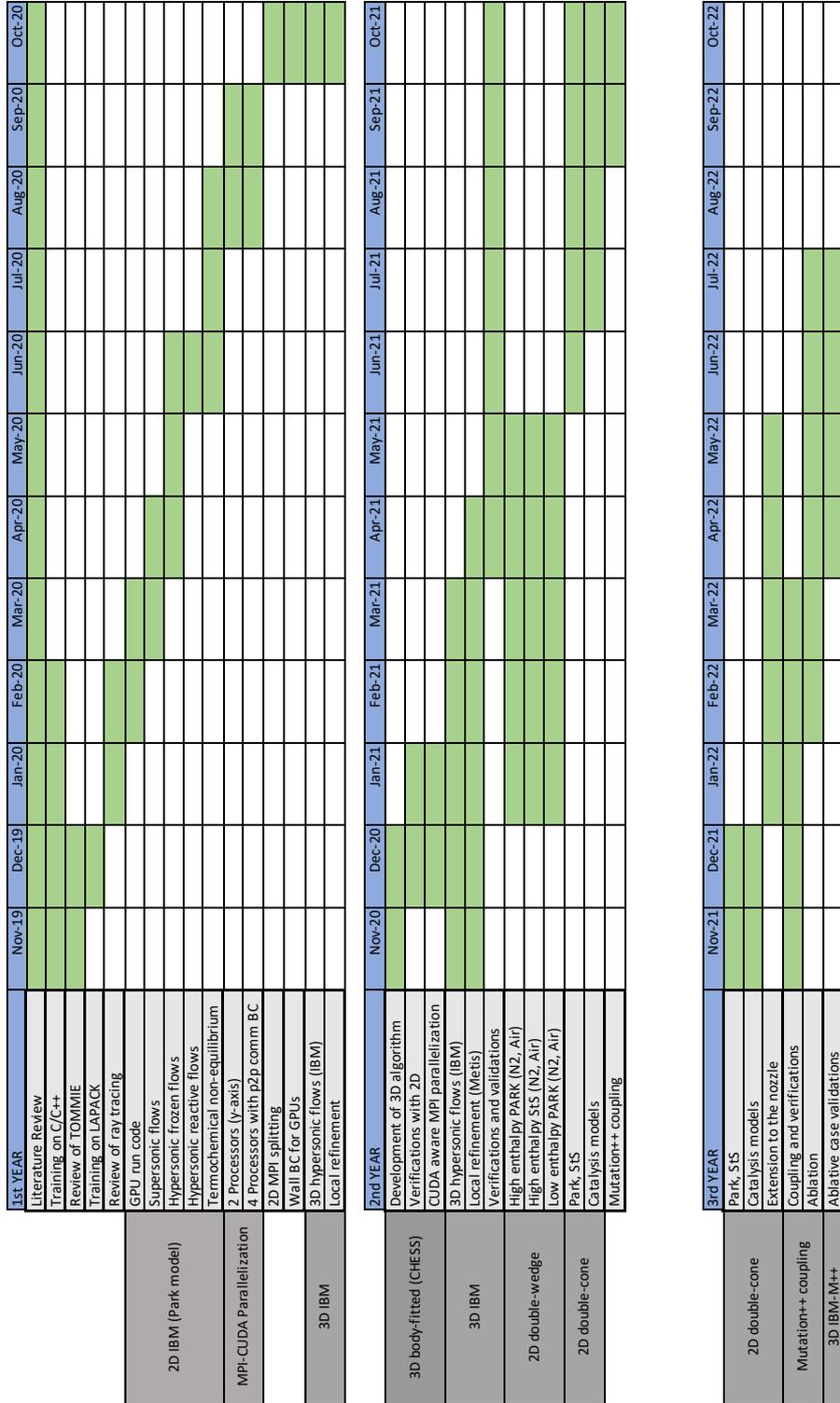


Figure G.1: Gantt chart of the project.

References

- [1] G. V. Candler, Rate effects in hypersonic flows, *Annual Review of Fluid Mechanics* 51 (2019) 379–402.
- [2] P. A. Gnoffo, Planetary-entry gas dynamics, *Annual Review of Fluid Mechanics* 31 (1) (1999) 459–494.
- [3] J. J. Bertin, R. M. Cummings, Fifty years of hypersonics: where we’ve been, where we’re going, *Progress in Aerospace Sciences* 39 (6-7) (2003) 511–536.
- [4] C. Park, Stagnation-point ablation of carbonaceous flat disks. I Theory, *AIAA journal* 21 (11) (1983) 1588–1594.
- [5] J. J. Bertin, R. M. Cummings, Critical hypersonic aerothermodynamic phenomena, *Annual Review of Fluid Mechanics* 38 (2006) 129.
- [6] B. Bottin, O. Chazot, M. Carbonaro, V. Van Der Haegen, S. Paris, The VKI plasmatron characteristics and performance, Tech. rep., VON KARMAN INST FOR FLUID DYNAMICS RHODE-SAINT-GENESE (BELGIUM) (2000).
- [7] C. Purpura, F. De Filippis, P. Barrera, D. Mandanici, Experimental characterisation of the CIRA plasma wind tunnel SCIROCCO test section, *Acta Astronautica* 62 (6-7) (2008) 410–421.
- [8] C. Park, *Nonequilibrium hypersonic aerothermodynamics*, John Wiley & Sons, 1989.
- [9] J. D. Anderson Jr, *Hypersonic and high-temperature gas dynamics*, American Institute of Aeronautics and Astronautics, 2006.
- [10] C. Park, Review of chemical-kinetic problems of future NASA missions. I-Earth entries, *Journal of Thermophysics and Heat transfer* 7 (3) (1993) 385–398.
- [11] C. Park, J. T. Howe, R. L. Jaffe, G. V. Candler, Review of chemical-kinetic problems of future NASA missions. II-Mars entries, *Journal of Thermophysics and Heat transfer* 8 (1) (1994) 9–23.

- [12] C. Park, Two-temperature interpretation of dissociation rate data for N₂ and O₂ (1988) 458.
- [13] M. Capitelli, I. Armenise, C. Gorse, State-to-state approach in the kinetics of air components under re-entry conditions, *Journal of thermophysics and heat transfer* 11 (4) (1997) 570–578.
- [14] G. Colonna, M. Tuttafesta, M. Capitelli, D. Giordano, Non-Arrhenius NO formation rate in one-dimensional nozzle airflow, *Journal of thermophysics and heat transfer* 13 (3) (1999) 372–375.
- [15] M. Capitelli, R. Celiberto, G. Colonna, F. Esposito, C. Gorse, K. Hassouni, A. Laricchiuta, S. Longo, *Fundamentals Aspects of Plasma Chemical Physics: Kinetics*, Springer, New York, 2016.
- [16] I. Armenise, M. Capitelli, G. Colonna, N. Koudriavtsev, V. Smetanin, Nonequilibrium vibrational kinetics during hypersonic flow of a solid body in nitrogen and its influence on the surface heat-flux, *Plasma Chemistry and Plasma Processing* 15 (3) (1995) 501–528.
- [17] G. Colonna, M. Capitelli, Electron and vibrational kinetics in the boundary layer of hypersonic flow, *Journal of thermophysics and heat transfer* 10 (3) (1996) 406–412.
- [18] G. Colonna, M. Capitelli, The influence of atomic and molecular metastable states in high-enthalpy nozzle expansion nitrogen flows, *Journal of Physics D: Applied Physics* 34 (12) (2001) 1812.
- [19] G. Colonna, L. D. Pietanza, A. Laricchiuta, Ionization kinetic model for Hydrogen-Helium atmospheres in hypersonic shock tubes, *International Journal of Heat and Mass Transfer* 156 (2020) 119916.
- [20] D. Giordano, V. Bellucci, G. Colonna, M. Capitelli, I. Armenise, C. Bruno, Vibrationally relaxing flow of N past an infinite cylinder, *Journal of thermophysics and heat transfer* 11 (1) (1997) 27–35.
- [21] G. Colonna, F. Bonelli, G. Pascazio, Impact of fundamental molecular kinetics on macroscopic properties of high-enthalpy flows: The case of hypersonic atmospheric entry, *Physical Review Fluids* 4 (3) (2019) 033404.
- [22] F. Bonelli, G. Pascazio, G. Colonna, Effect of finite-rate catalysis on wall heat flux prediction in hypersonic flow, *Phys. Rev. Fluids* 6 (2021) 033201.
- [23] D. Ninni, F. Bonelli, G. Colonna, G. Pascazio, Unsteady behavior and thermochemical non equilibrium effects in hypersonic double-wedge flows, *Acta Astronautica* 191 (2022) 178–192.

- [24] L. Cutrone, M. Tuttafesta, M. Capitelli, A. Schettino, G. Pascazio, G. Colonna, 3D nozzle flow simulations including state-to-state kinetics calculation, in: AIP Conference Proceedings, Vol. 1628, American Institute of Physics, 2014, pp. 1154–1161.
- [25] D. Knight, O. Chazot, J. Austin, M. A. Badr, G. Candler, B. Celik, D. de Rosa, R. Donelli, J. Komives, A. Lani, et al., Assessment of predictive capabilities for aerodynamic heating in hypersonic flow, *Progress in Aerospace Sciences* 90 (2017) 39–53.
- [26] A. Swantek, J. Austin, Flowfield establishment in hypervelocity shock-wave/boundary-layer interactions, *AIAA Journal* 53 (2) (2015) 311–320.
- [27] J. Hao, C.-Y. Wen, J. Wang, Numerical investigation of hypervelocity shock-wave/boundary-layer interactions over a double-wedge configuration, *International Journal of Heat and Mass Transfer* 138 (2019) 277–292.
- [28] J. D. Reinert, G. V. Candler, J. R. Komives, Simulations of Unsteady Three-Dimensional Hypersonic Double-Wedge Flow Experiments, *AIAA journal* 58 (9) (2020) 4055–4067.
- [29] A. Durna, B. Celik, Time-periodic shock interaction mechanisms over double wedges at Mach 7, *Shock Waves* 29 (3) (2019) 381–399.
- [30] D. Knight, J. Longo, D. Drikakis, D. Gaitonde, A. Lani, I. Nompelis, B. Reimann, L. Walpot, Assessment of CFD capability for prediction of hypersonic shock interactions, *Progress in Aerospace Sciences* 48 (2012) 8–26.
- [31] M. S. Holden, Experimental research and analysis in supersonic and hypervelocity flows in the LENS shock tunnels and expansion tunnel, in: 20th AIAA international space planes and hypersonic systems and technologies conference, 2015, p. 3660.
- [32] I. Nompelis, G. Candler, Numerical investigation of double-cone flow experiments with high-enthalpy effects, in: 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2010, p. 1283.
- [33] S. Nonaka, H. Mizuno, K. Takayama, C. Park, Measurement of shock stand-off distance for sphere in ballistic range, *Journal of thermophysics and heat transfer* 14 (2) (2000) 225–229.
- [34] M. Tuttafesta, G. Pascazio, G. Colonna, Multi-GPU unsteady 2D flow simulation coupled with a state-to-state chemical kinetics, *Computer Physics Communications* 207 (2016) 243–257.

- [35] F. Bonelli, M. Tuttafesta, G. Colonna, L. Cutrone, G. Pascazio, Numerical investigation of high enthalpy flows, *Energy Procedia* 126 (2017) 99–106.
- [36] D. Olynick, Y.-K. Chen, M. E. Tauber, Aerothermodynamics of the Stardust sample return capsule, *Journal of Spacecraft and Rockets* 36 (3) (1999) 442–462.
- [37] C. Park, Calculation of stagnation-point heating rates associated with stardust vehicle, *Journal of Spacecraft and Rockets* 44 (1) (2007) 24–32.
- [38] Y.-K. Chen, F. S. Milos, Navier-Stokes solutions with finite rate ablation for planetary mission earth reentries, *Journal of Spacecraft and Rockets* 42 (6) (2005) 961–970.
- [39] Y. Dobrov, V. Gimadiev, A. Karpenko, K. Volkov, Numerical simulation of hypersonic flow with non-equilibrium chemical reactions around sphere, *Acta Astronautica* (2021).
- [40] H. Jiang, J. Liu, S. Luo, W. Huang, J. Wang, M. Liu, Thermochemical non-equilibrium effects on hypersonic shock wave/turbulent boundary-layer interaction, *Acta Astronautica* 192 (2022) 1–14.
- [41] T. T. Aiken, I. D. Boyd, L. Duan, J. Huang, Assessment of Reynolds Averaged Navier-Stokes Models for a Hypersonic Cold-Wall Turbulent Boundary Layer, in: *AIAA SCITECH 2022 Forum*, 2022, p. 0586.
- [42] J. Urzay, Supersonic combustion in air-breathing propulsion systems for hypersonic flight, *Annual Review of Fluid Mechanics* 50 (2018) 593–627.
- [43] L. Sciacovelli, D. Passiatore, P. Cinnella, G. Pascazio, Assessment of a high-order shock-capturing central-difference scheme for hypersonic turbulent flow simulations, *Computers & Fluids* 230 (2021) 105134.
- [44] C. Stemmer, N. Mansour, DNS of transition in hypersonic boundary-layer flows including high-temperature gas effects, *Annual Research Briefs* 2001 (2001) 143–150.
- [45] D. Passiatore, L. Sciacovelli, P. Cinnella, G. Pascazio, Finite-rate chemistry effects in turbulent hypersonic boundary layers: A direct numerical simulation study, *Physical Review Fluids* 6 (5) (2021) 054604.
- [46] D. Passiatore, L. Sciacovelli, P. Cinnella, G. Pascazio, Thermochemical non-equilibrium effects in turbulent hypersonic boundary layers, *Journal of Fluid Mechanics* 941 (2022).
- [47] R. Mittal, G. Iaccarino, Immersed boundary methods, *Annu. Rev. Fluid Mech.* 37 (2005) 239–261.

- [48] C. S. Peskin, Flow patterns around heart valves: a numerical method, *Journal of computational physics* 10 (2) (1972) 252–271.
- [49] C. S. Peskin, Numerical analysis of blood flow in the heart, *Journal of computational physics* 25 (3) (1977) 220–252.
- [50] C. S. Peskin, D. M. McQueen, Modeling prosthetic heart valves for numerical analysis of blood flow in the heart, *Journal of Computational Physics* 37 (1) (1980) 113–132.
- [51] C. Basdevant, R. Sadourny, Numerical simulation of incompressible flow: the Mask method, *Laboratoire de Meteorologie Dynamique Ecole Normale Supérieure, Paris* (Unpublished) (1984).
- [52] M. Briscolini, P. Santangelo, Development of the Mask method for incompressible unsteady flows, *Journal of Computational Physics* 84 (1) (1989) 57–75.
- [53] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, *Journal of computational physics* 105 (2) (1993) 354–366.
- [54] J. Mohd-Yusof, Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries, *Center for turbulence research annual research briefs* 161 (1) (1997) 317–327.
- [55] E. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *Journal of computational physics* 161 (1) (2000) 35–60.
- [56] E. Balaras, Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations, *Computers & Fluids* 33 (3) (2004) 375–404.
- [57] M. D. de Tullio, P. De Palma, G. Iaccarino, G. Pascazio, M. Napolitano, An immersed boundary method for compressible flows using local grid refinement, *Journal of Computational Physics* 225 (2) (2007) 2098–2117.
- [58] S. Roy, A. De, E. Balaras, *Immersed Boundary Method* (2020).
- [59] P. A. Gnoffo, Conservation equations and physical models for hypersonic air flows in thermal and chemical nonequilibrium, Vol. 2867, *National Aeronautics and Space Administration, Office of Management . . .*, 1989.
- [60] T. Poinso, D. Veynante, *Theoretical and numerical combustion*, RT Edwards, Inc., 2005.

- [61] J. O. Hirschfelder, C. F. Curtiss, R. B. Bird, *Molecular theory of gases and liquids*, 1964.
- [62] K. K.-y. Kuo, R. Acharya, *Fundamentals of turbulent and multiphase combustion*, John Wiley & Sons, 2012.
- [63] G. Sarma, Physico-chemical modelling in hypersonic flow simulation, *Progress in Aerospace Sciences* 36 (3-4) (2000) 281–349.
- [64] T. E. Magin, G. Degrez, Transport algorithms for partially ionized and unmagnetized plasmas, *Journal of computational Physics* 198 (2) (2004) 424–449.
- [65] I. Armenise, E. Kustova, On different contributions to the heat flux and diffusion in non-equilibrium flows, *Chemical Physics* 428 (2014) 90–104.
- [66] M. Capitelli, G. Colonna, A. D’Angola, *Fundamental Aspects of Plasma Chemical Physics: Thermodynamics*, 1st Edition, Vol. 66 of *Atomic, Optical, and Plasma Physics*, Springer, New York, 2011.
- [67] M. Tuttafesta, *Theoretical aspects of reactive gas dynamics aided by massive parallel computing*, Ph.D. thesis (2014).
- [68] C. Wilke, A viscosity equation for gas mixtures, *The journal of chemical physics* 18 (4) (1950) 517–519.
- [69] E. Mason, S. Saxena, Approximate formula for the thermal conductivity of gas mixtures, *The Physics of fluids* 1 (5) (1958) 361–369.
- [70] R. B. Bird, W. E. Stewart, E. N. Lightfoot, *Transport Phenomena* revised 2nd Edition, John Wiley & Sons, Inc New York, 2006.
- [71] R. Design, *ANSYS Chemkin Theory Manual 17.0* (2015).
- [72] R. N. Gupta, J. M. Yos, R. A. Thompson, K.-P. Lee, 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 90* (1990) 27064.
- [73] M. Capitelli, G. Colonna, C. Gorse, A. d’Angola, Transport properties of high temperature air in local thermodynamic equilibrium, *The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics* 11 (2) (2000) 279–289.
- [74] E. Kustova, On the simplified state-to-state transport coefficients, *Chemical Physics* 270 (1) (2001) 177–195.

- [75] V. Istomin, E. Kustova, M. Mekhonoshina, Eucken correction in high-temperature gases with electronic excitation, *The Journal of Chemical Physics* 140 (18) (2014) 184311.
- [76] B. Srivastava, R. Srivastava, Thermal conductivity and eucken correction for diatomic gases and binary gas mixtures, *The Journal of Chemical Physics* 30 (5) (1959) 1200–1205.
- [77] J. O. Hirschfelder, Heat conductivity in polyatomic or electronically excited gases. II, *The Journal of Chemical Physics* 26 (2) (1957) 282–285.
- [78] C. Park, A review of reaction rates in high temperature air, in: 24th thermophysics conference, 1989, p. 1740.
- [79] R. C. Millikan, D. R. White, Systematics of vibrational relaxation, *The Journal of chemical physics* 39 (12) (1963) 3209–3213.
- [80] M. Capitelli, G. Colonna, A. D'Angola, *Fundamental aspects of plasma chemical physics*, Vol. 66, Springer, 2012.
- [81] O. Kunova, E. Kustova, M. Mekhonoshina, E. Nagnibeda, Non-equilibrium kinetics, diffusion and heat transfer in shock heated flows of N₂/N and O₂/O mixtures, *Chemical Physics* 463 (2015) 70–81.
- [82] L. Campoli, O. Kunova, E. Kustova, M. Melnik, Models validation and code profiling in state-to-state simulations of shock heated air flows, *Acta Astronautica* 175 (2020) 493–509.
- [83] I. Armenise, M. Capitelli, G. Colonna, G. Gorse, Nonequilibrium vibrational kinetics in the boundary layer of re-entering bodies, *Journal of thermophysics and heat transfer* 10 (3) (1996) 397–405.
- [84] I. Armenise, M. Capitelli, State to state vibrational kinetics in the boundary layer of an entering body in earth atmosphere: particle distributions and chemical kinetics, *Plasma Sources Science and Technology* 14 (2) (2005) S9.
- [85] I. Armenise, F. Esposito, N₂, O₂, NO state-to-state vibrational kinetics in hypersonic boundary layers: The problem of rescaling rate coefficients to uniform vibrational ladders, *Chemical Physics* 446 (2015) 30–46.
- [86] I. Armenise, M. Barbato, M. Capitelli, E. Kustova, State-to-state catalytic models, kinetics, and transport in hypersonic boundary layers, *Journal of thermophysics and heat transfer* 20 (3) (2006) 465–476.
- [87] G. Colonna, I. Armenise, D. Bruno, M. Capitelli, Reduction of state-to-state kinetics to macroscopic models in hypersonic flows, *Journal of thermophysics and heat transfer* 20 (3) (2006) 477–486.

- [88] G. Colonna, L. D. Pietanza, M. Capitelli, Recombination-assisted nitrogen dissociation rates under nonequilibrium conditions, *Journal of Thermophysics and Heat Transfer* 22 (3) (2008) 399–406.
- [89] G. Colonna, F. Esposito, M. Capitelli, The role of state-selected recombination rates in supersonic nozzle expansion, in: 36th AIAA Thermophysics Conference, 2003, p. 3645.
- [90] G. Herzberg, *Molecular spectra and molecular structure: I. Spectra of Diatomic Molecules*, Vol. 1, New York: Van Nostrand, 1963.
- [91] I. Armenise, F. Esposito, M. Capitelli, Dissociation–recombination models in hypersonic boundary layer flows, *Chemical Physics* 336 (1) (2007) 83–90.
- [92] F. Esposito, I. Armenise, M. Capitelli, N–N₂ state to state vibrational-relaxation and dissociation rates based on quasiclassical calculations, *Chemical Physics* 331 (1) (2006) 1–8.
- [93] G. D. Billing, E. Fisher, VV and VT rate coefficients in N₂ by a quantum-classical model, *Chemical Physics* 43 (3) (1979) 395–401.
- [94] G. Colonna, L. Pietanza, M. Capitelli, Macroscopic rates with vibrational non equilibrium for N₂ dissociation, in: 39th AIAA Thermophysics Conference, 2007, p. 4554.
- [95] F. Esposito, M. Capitelli, QCT calculations for the process N₂ (v)+ N → N₂ (v′)+ N in the whole vibrational range, *Chemical Physics Letters* 418 (4-6) (2006) 581–585.
- [96] O. Shatalov, Recommended data on rate constants of physical and chemical processes in N–O atoms system, Moscow State University–Russia, Institut of Mechanics AVOGADRO Center (1987).
- [97] G. D. Billing, R. Kolesnick, Vibrational relaxation of oxygen. State to state rate constants, *Chemical Physics Letters* 200 (4) (1992) 382–386.
- [98] F. Esposito, I. Armenise, G. Capitta, M. Capitelli, O–O₂ state-to-state vibrational relaxation and dissociation rates based on quasiclassical calculations, *Chemical Physics* 351 (1-3) (2008) 91–98.
- [99] M. Capitelli, C. M. Ferreira, B. F. Gordiets, A. I. Osipov, *Plasma kinetics in atmospheric gases*, Vol. 31, Springer Science & Business Media, 2013.
- [100] B. Potapkin, S. Macheret, *Physical and Chemical Processes in Gas Dynamics: Cross sections and rate constants for physical and chemical processes*, Vol. 1, AIAA (American Institute of Aeronautics & Astronautics), 2002.

- [101] M. Barbato, D. Giordano, J. Muylaert, C. Bruno, Comparison of catalytic wall conditions for hypersonic flow, *Journal of spacecraft and rockets* 33 (5) (1996) 620–627.
- [102] R. Goulard, On catalytic recombination rates in hypersonic stagnation heat transfer, *Journal of Jet Propulsion* 28 (11) (1958) 737–745.
- [103] G. Bellas-Chatzigeorgis, P. F. Barbante, T. E. Magin, Energy accommodation coefficient calculation methodology using state-to-state catalysis applied to hypersonic flows, *AIAA Journal* 58 (1) (2020) 278–290.
- [104] G. Bellas-Chatzigeorgis, P. F. Barbante, T. Magin, et al., Development of detailed chemistry models for boundary layer catalytic recombination, in: *8th European Symposium on Aerothermodynamics for Space Vehicle*, 2015, pp. 1–8.
- [105] I. Armenise, M. Rutigliano, M. Cacciatore, M. Capitelli, Hypersonic boundary layers: oxygen recombination on SiO₂ starting from ab initio coefficients, *Journal of thermophysics and heat transfer* 25 (4) (2011) 627–632.
- [106] P. F. Barbante, Accurate and efficient modelling of high temperature nonequilibrium air flows, Ph.D. thesis (2001).
- [107] B. Helber, A. Turchi, T. E. Magin, Determination of active nitridation reaction efficiency of graphite in inductively coupled plasma flows, *Carbon* 125 (2017) 582–594.
- [108] C. Park, Effects of atomic oxygen on graphite ablation, *AIAA journal* 14 (11) (1976) 1640–1642.
- [109] G. Bellas Chatzigeorgis, A. Turchi, A. Viladegut, O. Chazot, P. F. Barbante, T. Magin, Development of catalytic and ablative gas-surface interaction models for the simulation of reacting gas mixtures, in: *23rd AIAA Computational Fluid Dynamics Conference*, 2017, p. 4499.
- [110] C. H. Mortensen, X. Zhong, Simulation of second-mode instability in a real-gas hypersonic flow with graphite ablation, *AIAA journal* 52 (8) (2014) 1632–1652.
- [111] C. Park, R. L. Jaffe, H. Partridge, Chemical-kinetic parameters of hyperbolic earth entry, *Journal of Thermophysics and Heat transfer* 15 (1) (2001) 76–90.
- [112] J. B. Scoggins, V. Leroy, G. Bellas-Chatzigeorgis, B. Dias, T. E. Magin, Mutation++: Multicomponent thermodynamic and transport properties for ionized gases in C++, *SoftwareX* 12 (2020) 100575.
- [113] R. H. Pletcher, J. C. Tannehill, D. Anderson, *Computational fluid mechanics and heat transfer*, CRC press, 2012.

- [114] C. Hirsch, Numerical computation of internal and external flows. Vol. 2-Computational methods for inviscid and viscous flows(Book), 1990.
- [115] J. L. Steger, R. Warming, Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods, *Journal of computational physics* 40 (2) (1981) 263–293.
- [116] B. Van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method, *Journal of computational Physics* 32 (1) (1979) 101–136.
- [117] A. Harten, High resolution schemes for hyperbolic conservation laws, *Journal of computational physics* 135 (2) (1997) 260–278.
- [118] J. G. Verwer, Gauss–Seidel iteration for stiff ODEs from chemical kinetics, *SIAM Journal on Scientific Computing* 15 (5) (1994) 1243–1250.
- [119] C. Hirsch, Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics, Elsevier, 2007.
- [120] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, *Journal of computational physics* 126 (1) (1996) 202–228.
- [121] A. Coussement, O. Gicquel, B. Fiorina, G. Degrez, N. Darabiha, Multi-component real gas 3-D-NSCBC for direct numerical simulation of reactive compressible viscous flows, *Journal of Computational Physics* 245 (2013) 259–280.
- [122] A. Coussement, O. Gicquel, J. Caudal, B. Fiorina, G. Degrez, Three-dimensional boundary conditions for numerical simulations of reactive compressible flows with complex thermochemistry, *Journal of computational physics* 231 (17) (2012) 5571–5611.
- [123] T. J. Poinso, S. Lelef, Boundary conditions for direct simulations of compressible viscous flows, *Journal of computational physics* 101 (1) (1992) 104–129.
- [124] N. Okong’o, J. Bellan, Consistent boundary conditions for multicomponent real gas mixtures based on characteristic waves, *Journal of Computational Physics* 176 (2) (2002) 330–344.
- [125] J. Sanders, E. Kandrot, CUDA by example: an introduction to general-purpose GPU programming, Addison-Wesley Professional, 2010.
- [126] D. Guide, CUDA C programming guide, NVIDIA, July 29 (2013) 31.
- [127] S. Terrana, C. Nguyen, J. Peraire, GPU-accelerated Large Eddy Simulation of Hypersonic Flows, in: *AIAA Scitech 2020 Forum*, 2020, p. 1062.

- [128] M. Di Renzo, L. Fu, J. Urzay, HTR solver: An open-source exascale-oriented task-based multi-GPU high-order code for hypersonic aerothermodynamics, *Computer Physics Communications* 255 (2020) 107262.
- [129] M. Tuttafesta, G. Colonna, G. Pascazio, Computing unsteady compressible flows using Roe's flux-difference splitting scheme on GPUs, *Computer Physics Communications* 184 (6) (2013) 1497–1510.
- [130] F. Bonelli, M. Tuttafesta, G. Colonna, L. Cutrone, G. Pascazio, An MPI-CUDA approach for hypersonic flows with detailed state-to-state air kinetics using a GPU cluster, *Computer Physics Communications* 219 (2017) 178–195.
- [131] P. S. Pacheco, *An Introduction to Parallel Programming*, Morgan Kaufmann, 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA, 2011.
- [132] <https://wiki.u-gov.it/confluence/display/SCAIUS/UG3.2%3A+MARCONI100+UserGuide>.
- [133] I. Nompelis, T. Drayna, G. Candler, Development of a hybrid unstructured implicit solver for the simulation of reacting flows over complex geometries, in: *34th AIAA Fluid Dynamics Conference and Exhibit*, 2004, p. 2227.
- [134] I. Nompelis, T. W. Drayna, G. V. Candler, A parallel unstructured implicit solver for hypersonic reacting flow simulation, in: *Parallel Computational Fluid Dynamics 2005*, Elsevier, 2006, pp. 389–395.
- [135] G. V. Candler, H. B. Johnson, I. Nompelis, V. M. Gidzak, P. K. Subbareddy, M. Barnhardt, Development of the US3D code for advanced compressible and reacting flow simulations, in: *53rd AIAA Aerospace Sciences Meeting*, 2015, p. 1893.
- [136] J. Iannelli, Direct computation of thermodynamic properties of chemically reacting air with consideration to CFD, *International journal for numerical methods in fluids* 43 (4) (2003) 369–406.
- [137] B. Grossman, P. Cinnella, Flux-split algorithms for flows with non-equilibrium chemistry and vibrational relaxation, *Journal of Computational Physics* 88 (1) (1990) 131–168.
- [138] D. Vatansever, B. Celik, Unsteady shock interaction mechanisms of high enthalpy reacting flows over double wedges at Mach 7, *Physics of Fluids* 33 (5) (2021) 056110.
- [139] D. de Rosa, G. Pezzella, R. S. Donelli, A. Viviani, Flap effectiveness appraisal for winged re-entry vehicles, *Acta Astronautica* 122 (2016) 175–184.

- [140] F.-Y. Zuo, S.-L. Hu, Thermochemical non-equilibrium effects on aerothermodynamic prediction of laminar double-cone flow, *Acta Astronautica* 182 (2021) 179–188.
- [141] G. Shoen, G. Oblapenko, O. Kunova, M. Mekhonoshina, E. Kustova, Validation of vibration-dissociation coupling models in hypersonic non-equilibrium separated flows, *Acta Astronautica* 144 (2018) 147–159.
- [142] M. MacLean, G. Candler, M. Holden, Numerical evaluation of flow conditions in the LENS reflected shock-tunnel facilities, in: 43rd AIAA Aerospace Sciences Meeting and Exhibit, 2005, p. 903.
- [143] I. Nompelis, G. Candler, M. MacLean, T. Wadhams, M. Holden, Numerical Investigation of High Enthalpy Chemistry on Hypersonic Double-Cone Experiments, in: 43rd AIAA Aerospace Sciences Meeting and Exhibit, 2005, p. 584.
- [144] G. Candler, Hypersonic nozzle analysis using an excluded volume equation of state, in: 38th AIAA Thermophysics Conference, 2005, p. 5202.
- [145] J. Hao, J. Wang, C. Lee, Numerical simulation of high-enthalpy double-cone flows, *AIAA Journal* 55 (7) (2017) 2471–2475.
- [146] J. Hao, C.-Y. Wen, Numerical investigation of oxygen thermochemical nonequilibrium on high-enthalpy double-cone flows, *International Journal of Heat and Mass Transfer* 127 (2018) 892–902.
- [147] M. E. Holloway, K. M. Hanquist, I. D. Boyd, Assessment of Thermochemistry Modeling for Hypersonic Flow over a Double Cone, *Journal of Thermophysics and Heat Transfer* 34 (3) (2020) 538–547.
- [148] M. E. Holloway, I. D. Boyd, Assessment of thermochemical kinetics modeling on hypersonic flow over a double cone, in: AIAA Scitech 2020 Forum, 2020, p. 1228.
- [149] J. Hao, C.-Y. Wen, Effects of vibrational nonequilibrium on hypersonic shock-wave/laminar boundary-layer interactions, *International Communications in Heat and Mass Transfer* 97 (2018) 136–142.
- [150] N. Kianvashrad, D. D. Knight, Nonequilibrium effects on prediction of aerothermodynamic loading for a double cone, *AIAA journal* 57 (7) (2019) 2946–2963.
- [151] N. Kianvashrad, D. D. Knight, The effect of thermochemistry on prediction of aerothermodynamic loading over a double cone in a laminar hypersonic flow, in: 2018 AIAA Aerospace Sciences Meeting, 2018, p. 1812.

- [152] M. S. Holden, T. P. Wadhams, M. G. MacLean, A. T. Dufrene, Measurements of real gas effects on regions of laminar shock wave/boundary layer interaction in hypervelocity flows for “blind” code validation studies, in: 21st AIAA Computational Fluid Dynamics Conference, 2013, p. 2837.
- [153] A. Swantek, J. Austin, Heat transfer on a double wedge geometry in hypervelocity air and nitrogen flows (2012) 284.
- [154] C. Garbacz, F. Morgado, M. Fossati, Effect of thermal nonequilibrium on the shock interaction mechanism for carbon dioxide mixtures on double-wedge geometries, *Physics of Fluids* 34 (2) (Feb. 2022).
- [155] J. Hao, J. Fan, S. Cao, C.-Y. Wen, Three-dimensionality of hypersonic laminar flow over a double cone, *Journal of Fluid Mechanics* 935 (2022).
- [156] D. Ninni, F. Bonelli, G. Colonna, G. Pascazio, On the influence of non equilibrium in the free stream conditions of high enthalpy oxygen flows around a double-cone, *Acta Astronautica* 201 (2022) 247–258.
- [157] O. Tumuklu, D. A. Levin, V. Theofilis, Investigation of unsteady, hypersonic, laminar separated flows over a double cone geometry using a kinetic approach, *Physics of Fluids* 30 (4) (2018) 046103.
- [158] O. Tumuklu, V. Theofilis, D. A. Levin, On the unsteadiness of shock–laminar boundary layer interactions of hypersonic flows over a double cone, *Physics of Fluids* 30 (10) (2018) 106111.
- [159] A. A. Pasha, K. A. Juhany, Effect of wall temperature on separation bubble size in laminar hypersonic shock/boundary layer interaction flows, *Advances in Mechanical Engineering* 11 (11) (2019) 1687814019885556.
- [160] J. G. Kim, S. H. Kang, S. H. Park, Thermochemical nonequilibrium modeling of oxygen in hypersonic air flows, *International Journal of Heat and Mass Transfer* 148 (2020) 119059.
- [161] K. Neitzel, D. Andrienko, I. D. Boyd, Thermochemical nonequilibrium modeling for hypersonic flows containing oxygen, in: 46th AIAA Thermophysics Conference, 2016, p. 4023.
- [162] J. Ray, S. Kieweg, D. Dinzl, B. Carnes, V. G. Weirs, B. Freno, M. Howard, T. Smith, I. Nompelis, G. V. Candler, Estimation of inflow uncertainties in laminar hypersonic double-cone experiments, *AIAA journal* 58 (10) (2020) 4461–4474.
- [163] S. L. Kieweg, J. Ray, V. G. Weirs, B. Carnes, D. Dinzl, B. Freno, M. Howard, E. Phipps, W. Rider, T. Smith, Validation assessment of hypersonic double-cone flow simulations using uncertainty quantification, sensitivity analysis, and validation metrics, in: AIAA SciTech 2019 Forum, 2019, p. 2278.

- [164] Y. Cengel, J. Cimbala, *Fluid Mechanics Fundamentals and Applications*, McGraw Hill, 2013.
- [165] M. Capriati, K. Prata, T. Schwartzentruber, G. Candler, T. Magin, Development of a nitridation gas-surface boundary condition for high-fidelity hypersonic simulations, *WCCM-ECCOMAS (2021)*.
- [166] A. O. Baskaya, M. Capriati, D. Ninni, F. Bonelli, G. Pascazio, A. Turchi, T. Magin, S. Hickel, Verification and Validation of Immersed Boundary Solvers for Hypersonic Flows with Gas-Surface Interactions, in: *AIAA AVIATION 2022 Forum*, 2022, p. 3276.
- [167] F. Bariselli, E. Torres, T. Magin, State-specific catalytic recombination boundary condition for DSMC methods in aerospace applications, in: *AIP Conference Proceedings*, Vol. 1786, AIP Publishing LLC, 2016, p. 190009.
- [168] L. Eça, C. M. Klaij, G. Vaz, M. Hoekstra, F. S. Pereira, On code verification of RANS solvers, *Journal of Computational Physics* 310 (2016) 418–439.
- [169] T. S. Phillips, C. J. Roy, Richardson extrapolation-based discretization uncertainty estimation for computational fluid dynamics, *Journal of Fluids Engineering* 136 (12) (2014).
- [170] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, *Journal of computational physics* 171 (1) (2001) 132–150.
- [171] J. Kim, H. Choi, An immersed-boundary finite-volume method for simulation of heat transfer in complex geometries, *KSME international journal* 18 (6) (2004) 1026–1035.
- [172] S. Majumdar, G. Iaccarino, P. Durbin, et al., RANS solvers with adaptive structured boundary non-conforming grids, *Annual Research Briefs* 1 (2001).
- [173] M. Vanella, E. Balaras, A moving-least-squares reconstruction for embedded-boundary formulations, *Journal of computational physics (Print)* 228 (18) (2009) 6617–6628.
- [174] A. Coclite, M. D. de Tullio, G. Pascazio, P. Decuzzi, A combined Lattice Boltzmann and Immersed boundary approach for predicting the vascular transport of differently shaped particles, *Computers & Fluids* 136 (2016) 260–271.
- [175] A. Coclite, S. Ranaldo, M. De Tullio, P. Decuzzi, G. Pascazio, Kinematic and dynamic forcing strategies for predicting the transport of inertial capsules via a combined lattice Boltzmann–immersed boundary method, *Computers & Fluids* 180 (2019) 41–53.

- [176] M. Haji Mohammadi, F. Sotiropoulos, J. Brinkerhoff, Moving least squares reconstruction for sharp interface immersed boundary methods, *International Journal for Numerical Methods in Fluids* 90 (2) (2019) 57–80.
- [177] S. Wang, M. Vanella, E. Balaras, A hydrodynamic stress model for simulating turbulence/particle interactions with immersed boundary methods, *Journal of Computational Physics* 382 (2019) 240–263.
- [178] C. Brehm, H. F. Fasel, A novel concept for the design of immersed interface methods, *Journal of Computational Physics* 242 (2013) 234–267.
- [179] C. Brehm, C. Hader, H. F. Fasel, A locally stabilized immersed boundary method for the compressible Navier–Stokes equations, *Journal of Computational Physics* 295 (2015) 475–504.
- [180] C. Brehm, M. F. Barad, C. C. Kiris, Development of immersed boundary computational aeroacoustic prediction capabilities for open-rotor noise, *Journal of Computational Physics* 388 (2019) 690–716.
- [181] S. Brahmachary, Finite volume/immersed boundary solvers for compressible flows: development and applications, Ph.D. thesis, Ph. D. thesis dissertation, Indian Institute of Technology Guwahati (2019).
- [182] S. Brahmachary, G. Natarajan, V. Kulkarni, N. Sahoo, V. Ashok, V. Kumar, Role of solution reconstruction in hypersonic viscous computations using a sharp interface immersed boundary method, *Physical Review E* 103 (4) (2021) 043302.
- [183] J. A. McQuaid, A. L. Zibitsker, B. Saikia, A. Martin, C. Brehm, An Immersed Boundary Method for Hypersonic Viscous Flows, in: *AIAA Scitech 2021 Forum*, 2021, p. 0926.
- [184] J. A. McQuaid, A. Zibitsker, A. Martin, C. Brehm, Heat Flux Predictions for High Speed Flows with an Immersed Boundary Method, in: *AIAA AVIATION 2021 FORUM*, 2021, p. 3145.
- [185] G. Iaccarino, F. Ham, Automatic mesh generation for LES in complex geometries, *Annual Research Briefs* (2005) 460–490.
- [186] M. D. De Tullio, Development of an Immersed Boundary Method for the Solution of the Preconditioned Navier-Stokers Equations, Ph.D. thesis (2006).
- [187] D. De Marinis, An immersed boundary method for coupled multi-physics simulations, Ph.D. thesis (2015).
- [188] G. Iaccarino, R. Verzicco, Immersed boundary technique for turbulent flow simulations, *Appl. Mech. Rev.* 56 (3) (2003) 331–347.

- [189] C. Chi, B. J. Lee, H. G. Im, An improved ghost-cell immersed boundary method for compressible flow simulations, *International Journal for Numerical Methods in Fluids* 83 (2) (2017) 132–148.
- [190] S. Das, A. Panda, N. Deen, J. Kuipers, A sharp-interface immersed boundary method to simulate convective and conjugate heat transfer through highly complex periodic porous structures, *Chemical Engineering Science* 191 (2018) 1–18.
- [191] M. Aftosmis, M. Berger, G. Adomavicius, A parallel multilevel method for adaptively refined Cartesian grids with embedded boundaries, in: *38th Aerospace Sciences Meeting and Exhibit*, 2000, p. 808.
- [192] H. Forrer, Boundary treatments for a Cartesian grid method, Ph.D. thesis (1997).
- [193] <http://glaros.dtc.umn.edu/gkhome/metis/metis/overview>.
- [194] M. Holden, A study of flow separation in regions of shock wave-boundary layer interaction in hypersonic flow, in: *11th Fluid and PlasmaDynamics Conference*, 1978, p. 1169.
- [195] R. Arslanbekov, V. Kolobov, A. Frolova, Analysis of compressible viscous flow solvers with adaptive Cartesian mesh, in: *20th AIAA Computational Fluid Dynamics Conference*, 2011, p. 3381.
- [196] A. Wieting, Experimental study of shock wave interference heating on a cylindrical leading edge. (1989).
- [197] F. S. Billig, Shock-wave shapes around spherical-and cylindrical-nosed bodies., *Journal of Spacecraft and Rockets* 4 (6) (1967) 822–823.
- [198] A. Di Mascio, S. Zaghi, An immersed boundary approach for high order weighted essentially non-oscillatory schemes, *Computers & Fluids* 222 (2021) 104931.