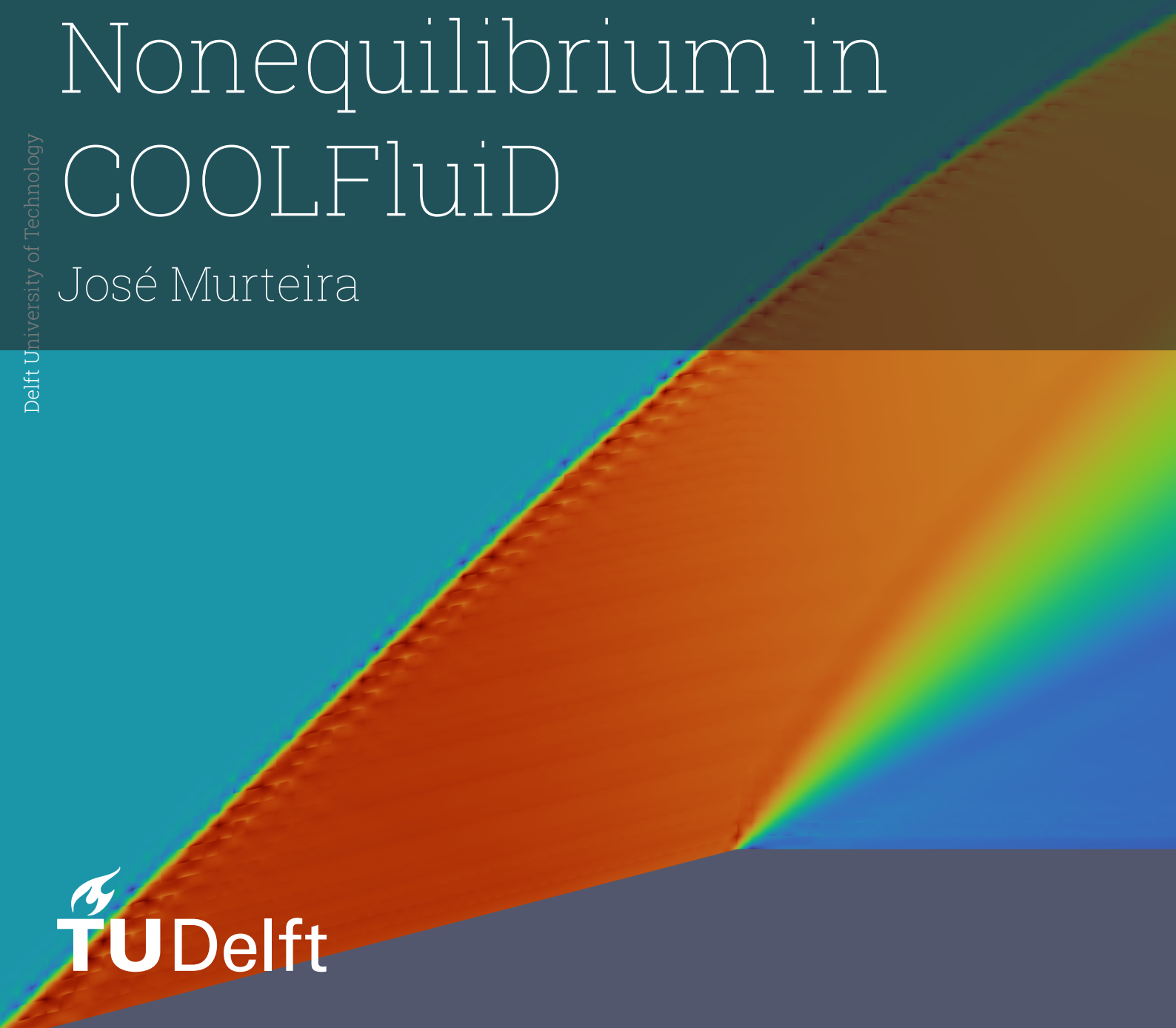
Positivity Preservation in Flux Reconstruction Methods with Thermochemical Nonequilibrium in COOLFluiD

José Murteira

Delft University of Technology



TU Delft

Positivity Preservation in Flux Reconstruction Methods with Thermochemical Nonequilibrium in COOLFluiD

by

José Murteira

to obtain the degree of Master of Science

at the Delft University of Technology,

Internal Supervisor:	Prof. Dr. ir. Stefan Hickel
External Supervisor:	Dr. Andrea Lani
Project Duration:	October, 2024 - June, 2025
Faculty:	Faculty of Aerospace Engineering, Delft

TU Delft

É ter fome, é ter sede de Infinito!
– Florbela Espanca, *in* "Ser Poeta"

"[...]
and deep below the earth
await answers to questions
yet unasked"
– *in* "Signalis"

Preface

This document concerns the thesis for the Masters in Aerospace Engineering at Technische Universiteit Delft, in the track Space Flight, in the profile of Space Engineering. Here stands the product of nine months of full time work in enlarging and testing specifics of a scientific code as well developing and implementing new methods for simulating reacting hypersonic flows.

The work was developed at KU Leuven, in the department of Mathematics, in the Centre for mathematical Plasma Astrophysics, in professor's Stefaan Poedts' group, under the external supervision of Dr. Andrea Lani, with the informal supervision contribution of Dr. Vatsalya Sharma and the Ph.D student Rayan Dhib, the latter of which gave large direct contributions in terms of guidance with the code and case running. The internal supervision was taken by Professor Stefan Hickel.

This thesis gave me the opportunity to truly explore, learn and expand upon the field higher order simulations for fluids. It is my sincere hope that the scientific contribution in this document may serve as basis for further developments in the field and in the on going research in the department.

It is not with lightness that I must affirm that this thesis also serves as a testament to the power of reading the code with a patient enough eye, namely in the face of lack of documentation. It was a learning process whose results and effects on me I hold with tight regard and motivation for the future. In contrast, it is with happiness and appreciation that I also learned to count on my research partners for guidance and discussion and to whom I am ever so much grateful, be it for understanding the code or the literature, to guide me in running, managing and testing cases, in analysing and structuring, in the standards and idiosyncrasies of research, and in further motivating me when roadblocks were found.

Given not only the interest in applicability of the work of this thesis and line of research, but also the pleasure of the research itself, I hope that any detailed future work may be taken upon and be fruitful.

José Murteira
Delft, June 2025

Acknowledgements

It is with much gratitude that I must acknowledge the contributions that made this thesis feasible and allowed it to come to fruition.

I would like to thank Professor Stefan Hickel for accepting to be my internal supervisor. I would also like to thank Dr. Andrea Lani for accepting my wish to participate in the project and conduct a thesis with him as well as to be my external supervisor.

To Professor Stefaan Poedts I must extend my gratitude accepting me into his research group not only for this thesis but also for the 3 months I spent in the spring of 2024, allowing me my first research experience. I value dearly how I was ever so warmly welcomed into this research group, and I truly hold dear this opportunity.

I must thank Rayan Dhib, currently a Ph.D student at KU Leuven, for guiding me closely in this thesis. His help with the code, the literature, and the overall research process was most crucial for the successfully advances in this thesis. I appreciate his kindness and lovely attitude, and how he took time from his ever so busy schedule to help me move forward, discuss my problems and questions, give me important insights, and reviewing my writing. I value greatly his help. I must also thank Dr. Vatsalya Sharma for his help during this thesis, namely with the literature, the discussion of the relevant physics and encountered problems, and in reviewing my writings. I must thank them both for meeting with me at ungodly times to help me, and for with true interest helping me in my efforts.

I must also unquestionably thank Dr. Michaela Brchnelová who not only welcomed me and guided me in my first time at KU Leuven, but also with whom I discussed the idea of doing a thesis here in the first place. She was most helpful in helping me in speaking with Dr. Andrea Lani for such opportunity. Her kindness and loveliness were undoubtedly part of why I ever so much enjoyed my time at KU Leuven.

I must thank my mother not only for her undying support in my endeavours, but also in my idiosyncratic ideas and pulls. I must thank my parents for their ever so present support and love and in nurturing a love for learning and this world. My family has ever been most supportive throughout all my years and hence more than deserve my gratitude and love. It is thanks to them that I am able to so pursue my studies and this thesis with sole focus.

I would like to ever so much thank Margarida for being not only one of the loveliest people I have had the pleasure to know, but for supporting me in unending days and nights throughout this thesis. Her incalculable patience and support marked these days.

To my good friends aeroamigos I extend my gratitude for all their friendship and patience, as well as helping me in most dire moments. They were there during my bachelors and masters, and I hold many memories with them dearly. I would have not gotten through my studies and thesis in the same way without them; nor my happiness would have been the same. Their patience, presence, friendship and kindness is something I will ever cherish, and which undoubtable enabled the path to and in this thesis.

I would like to thank Margot Winters for not only her friendship but also for introducing me to this department and the pushing forward the idea of contacting this research group. She was instrumental in me deciding for this path.

I must acknowledge the people at Centre for mathematical Plasma Astrophysics who received me ever so warmly and always contributed to great environment. They have made my stay during my thesis lovely and memorable, for which I am most grateful. May many more cakes come.

I must also thank my good friends at fencing for their continuous support and friendship for over a decade. I would also like to thank my friends at Airbus – namely Julia, Isa, Steven and Brendan – for their loveliness during my internship preceding this thesis. Their support during my masters was most appreciated.

It was beautiful to find that, despite what I was told, the sun in Belgium does shine.

José Murteira
Delft, June 2025

Abstract

Hypersonic flows regard a special class of flows where dire conditions for spacecraft may be found. Simulation of these flows requires taking chemistry and thermal nonequilibrium into account as to acquire the correct wall quantities for the vehicles. Flux reconstruction – a higher order method – enables fluid simulations with a coarser grid than finite volume, as well as with easier shock discretisation for the practitioner. Computational Object-Oriented Libraries for Fluid Dynamics (COOLFluiD) allows the simulation of hypersonic flows with flux reconstruction, but not yet with thermochemical nonequilibrium due to positivity issues encountered. A survey of the literature for positivity methods in higher order methods, together with a presentation of flux reconstruction, the chemical model used and COOLFluiD were given. An entropy based exponential filter and a Fejér filter were implemented in COOLFluiD based on a pre-existing framework of filter based positivity and tested against the previous positivity method in place. The exponential filter was found to be equivalent for a 2D simulation with the Euler equations to the previous method while the Féjer filter behaved in a more destabilising manner. The entropy based exponential filter was implemented for cases with multiple species and multiple temperatures, with modularity for species sets. The derivation of entropy for this case was given. A new positivity method based on least squares optimisation was developed to target density positivity issues where the filter fails. The case was tested against two cases from literature in P0, ensuring positivity and validated against literature. The cases were found not to be yet fully spatially resolved, as expected in P0 simulations. P1 simulations were tested for the same cases and found to have issues with the time integration method and the artificial viscosity. An inspection of the code base was executed, revealing that the artificial viscosity was not fully implemented for the cases tested and that the current base implementation of backwards Euler cannot be used for these cases, requiring being solved through the already implemented Newton iterations method. The artificial viscosity was fully implemented for the tested cases. The P1 simulations were retested. The artificial viscosity was found to be active and to diffuse the shock wave. No positivity failures of the method were found. A stringent CFL condition was found to exist, and possibly linked to the stiffness of the equations. The exponential filter was found to lead to a growth of the temperatures just upstream of the shock wave, where the filter is active. This behaviour was linked to a possible interaction with the highly nonlinear model or with the artificial viscosity. While positivity was found to work, further issues in testing it are still found and expected to be part of future work. Possible extensions or alterations to the code were found to possibly be required in order to allow further testing. Thus, positivity was found not to be the only issue with running thermochemical nonequilibrium simulations, being only part of the problem, requiring further research in order to fully test it in the context of COOLFluiD. Future work was defined to focus on studying the exponential filter interactions, on enabling future tests, on expanding on these testes based on presented literature, and on expanding on the model used to include more robust but complicated methods.

Contents

List of Figures

List of Tables

Nomenclature

Greek Symbols

ξ_{filter}	Exponential filter's strength
α	Blending coefficient
α_c	First coordinate component in the square element $[-\pi, \pi]$x$[-\pi, \pi]$
α_{damping}	Coefficient used for damping term in the interface flux for artificial viscosity.
α_{stoch}	A stoichiometric coefficient
β_c	Second coordinate component in the square element $[-\pi, \pi]$x$[-\pi, \pi]$
β_{stoch}	A stoichiometric coefficient
$\boldsymbol{\xi}$	Standard element coordinates
χ	Entropy constraint function
$\dot{\omega}$	Source term
ϵ_σ	Entropy functional's user defined tolerance
ϵ_{Jacob}	Perturbation for the numerical Jacobian
ϵ_l	Reference quantity for limiting based on average values on the element
ϵ_s	Small value added to a mass fraction
η	Standard element second coordinate
Γ	Functional used for constraints during filtering application
γ	Specific heat ratio
$\hat{\rho}$	Limited density
$\hat{\vartheta}$	Intermediate limited state
κ	Half width of the interval on the smoothness factor on which the AV strength varies
λ_s	Lagrange multiplier for equality constraints in the least squares positivity method
μ_s	Lagrange multiplier for inequality constraints in the least squares positivity method
Ω	Domain
Ω_n	Element
Ω_S	Standard element

ω_s Entropy variable for species s

ϕ Scalar function defined on the interval [-1,1] and used on on the Féjer filter derivation

ψ Modal basis function

ρ Density

σ Entropy functional used for constraints during filtering application

θ_2 Limiting coefficient

θ_l Limiting coefficient

Θ_n Mapping function from the standard element to the non standard element

Θ_v^s Characteristic vibrational temperature

ε_0 Base strength of the AV

ε_{AV} Viscosity coefficient of the Artificial Viscosity

ς Standard element third coordinate

ξ Standard element first coordinate

ξ_{filter} Exponential filter strength

t_ϵ Blending coefficient used during a limiting process

Roman Symbols

e_e Specific electron energy

e_t Specific internal energy

e_v Specific vibrational energy

s Specific entropy

s_α Intermediate coefficient for limiting process

$\boldsymbol{v}$ Species velocity

$\hat{F}_{n,i}$ Value of the polynomial representing the the Fluxes in the standard element at the solution point i

$\hat{u}_{n,i}$ Value of the polynomial representing the the conserved variables in the standard element at the solution point i

$\hat{w}$ When indexed and summed over that index, modal polynomial representation coefficient of the primitive variables saved in an element in a flux reconstruction scheme

$\mathbb{N}$ Natural numbers set

$\mathcal{H}$ Set of non electron species (denoted heavy species)

$\mathcal{I}$ Set of species which have roto-vibrational energy

$\mathcal{M}$ Molar mass

$\mathcal{V}$ Change of basis function from Lagrangian polynomials to a modal polynomial basis

$\mathscr{H}$ Filter

$\hbar$ Function defined on the interval [-1,1] and filtered by the Féjer filter

$\mathbf{F}$ Flux

$\mathbf{U}$ Conserved variables

$\mathbf{U}^{\delta}$ Function representing the conserved variables over the domain

$\mathbf{v}$ Velocity vector

T Temperature, assumed to be in equilibrium with the translational temperature

T_e Electron temperature

T_v Roto-vibrational temperature, possibly assumed to be in equilibrium with electron temperature

T_{tr} Translational temperature

$a_{\text{threshold}}$ Threshold related parameter for the order blending method

c_s Mass fraction of the species $_s$

$c_{\text{threshold}}$ Threshold related parameter for the order blending method

C_p Heat capacity at constant pressure

c_p Specific heat capacity at constant pressure

C_v Heat capacity at constant volume

c_v Specfici heat capacity at constant volume

E Total Energy

E_{max} Maximum energy in the order blending method

f Flux point f

F^{δ} Function part of the function set representing the fluxes over the domain

F_n Polynomial function representing the fluxes inside an element

f_{blended} Flux function in the blending order method

f_{strength} Fejér filter strength

g Function that maps $[-\pi, \pi]$ to [-1,1] used by the Féjer filter formulation

H Total Enthalpy

h Mesh size

h_f Flux reconstruction correction function

J_g Jacobian of the function that maps $[-\pi, \pi]$ to [-1,1] used by the Fejér filter formulation

J_n Jacobian of the function that maps an element to the standard element

k_B Boltzman constant

k_{mesh} Spatial order of a numerical method

k_b Backwards reaction rate

k_f Forward reaction rate

$l_{\text{characteristic}}$ Characteristic length used in the interface flux for artificial viscosity

$l_{d,i}$ the Lagrange polynomial in the d-direction, where d is a placeholder for the coordinate

m Mode

N_0 Number of densities set to 0 in the solution of the least squares positivity method

N_{pol} Number of polynomial representations inside each element

N_p Number of solution points

N_s Number of Species

N_s Number of species

$n_{d,i}$ the normal vector in the d-direction, where d is a placeholder for the coordinate, at the flux point f

N_d Number of directions in space, i.e the dimensionality of the physical space

P Polynomial order

p Pressure

p_k k^{th} primitive variable

Q Heat flux

q_d the gradient polynomial in the d-direction, where d is a placeholder for the coordinate

R Ideal gas constant

S Smoothness

S_0 User given reference smoothness

S_n Féjer Kernel of order n

s_ϵ Intermediate coefficient for limiting process

t Time

t_1 A blending parameter used in the current limiting method used by COOLFluiD

t_2 A blending parameter used in the current limiting method used by COOLFluiD

t_j	A blending parameter at a flux point j used in the current limiting method used by COOLFluiD
u	Conserved variables represented in a numerical method
u^δ	Function part of the function set representing the conserved variables over the domain
u_i	Value of a variable (e.g conserved variable) at a solution point i
u_n	Polynomial function representing the conserved variables inside an element
u_s	Diffusion velocity of the species
u_x	Velocity component in the x direction
V	Volume
v	Velocity component in the y direction
W	work per unit volume and per unit time
w	When indexed and summed over that index, primitive variable saved in a solution point in a flux reconstruction scheme
w_{quad}	Quadrature weight
w_{c_s}	Weight for a species $_s$ used in the least squares positivity method
X_s	Member s of the set of species in a given chemical reaction
Z	Primitive variables

Subscripts

$_A$	Related to species A
$_B$	Related to species B
$_D$	Related to the diffusive fluxes
$_E$	Related to the energy equation
$_e$	Related to electrons
$_F$	Related to the forces
$_f$	At the flux point f
$_h$	Related to heavy species
$_I$	Calculated at the interface, e.g interface flux
$_i$	In the absence of a summation over i, relates the variable to solution point i
$_j$	In the absence of a summation over j, relates the variable to flux point j
$_M$	Related to the momentum equation
$_m$	In the absence of a summation over m, it means the polynomial regards the monitored variables m and is in modal representation

$_S$ Related to entropy

$_S$ Standard

$_s$ Species index

$_{AV}$ Related to the Artificial Viscosity

Superscripts

$^{-}$ averaged quantity

$^{\sim}$ filtered or limited quantity

Acronyms

ARD	Atmospheric Reentry Demonstrator
AV	Artificial Viscosity
CE	Chapman-Enskog
CFD	Computational Fluid Dynamics
CFL	Courant–Friedrichs–Lewy
COOLFluiD	Computational Object-Oriented Libraries for Fluid Dynamics
CPU	Central Processing Unit
DG	Discontinuous Galerkin
EC	Entropy Conservative
ENO	Essentially Non-Oscillatory
ES	Entropy Stable
ESA	European Space Agency
FR	Flux Reconstruction
FV	Finite Volume
GPU	Graphics Processing Unit
HDG	Hybridised Discontinuous Galerkin
HEG	High Enthalpy shock tunnel Göttingen
HLL	Harten-Lax-van Leer
HLLC	Harten-Lax-van Leer-Contact
hMLP	high-order Multi-dimensional Limiting Process
LES	Large Eddy Simulations
LLAV	Localised Laplacian Artificial Viscosity
LS	Least Squares
MHD	Magneto Hydro Dynamics
NLTE	Non-local Thermal Equilibrium
PLATO	PLAsmas in Thermodynamic nOn-equilibrium
RANS	Reynolds Averaged Navier Stokes
RKDG	Runge-Kutta Discontinuous Galerkin
SD	Spectral Difference
SV	Spectral Volume
TCNEQ	Thermochemical Nonequilibrium
VCJH	Vincent–Castonguay–Jameson–Huynh
WENO	Weighted Essentially Non-Oscillatory

1

Introduction

Current research at Centre for mathematical Plasma Astrophysics at KU Leuven focuses, among other topics, in the use of higher order methods for hypersonic flow simulations. This has concerned the use of flux reconstruction (see the work of Huynh [1]) as the method of choice. The work developed allowed the use of flux reconstruction for hyperonic simulations with the Navier Stokes and with Reynolds averaged turbulence models [2].

Current efforts seek to extend this formulation to include thermal and chemical nonequilibrium. This extension were already target of research in works such as [2] and [3]. The simulation software used for this research is Computational Object-Oriented Libraries for Fluid Dynamics (COOLFluiD) [4], [5] which encompasses both finite volume and flux reconstruction formulations. The latest development in encompassing thermochemical nonequilibrium was the coupling of a new chemical source term model – PLAsmas in Thermodynamic nOn-equilibrium (PLATO) – for the finite volume formulation in COOLFluiD and its validation [6].

A high degree of interest is placed on thermochemical nonequilibrium models for hypersonic aerodynamics since the the flow, at these high speeds, often displays these conditions at and after the shock wave, affecting both the shock wave position, the thermodynamic quantities behind the shock wave and the wall quantities [7]. This concept is well established in literature, being present in textbooks such as the one by Anderson [7], in use chemical species and interaction models specifically for hypersonics, such as the report by Gupta, Yos, Thompson, *et al.* [8], and in current research in hypersonics at this research group [6]. Thus, the inclusion of thermochemical nonequilibrium is well motivated by the physics of the problem and literature, as well as already a target of ongoing research in the context of COOLFluiD.

Higher order models can display positivity problems, requiring a positivity scheme [2]. The current positivity scheme in use for COOLFluiD – which can be seen in works such as by Ameur, Vandenhoeck, and Lani [3] or Vandenhoeck [2] – was found internally to not be adequate for flux reconstruction with COOLFluiD when using PLATO for the chemistry, failing to correct negative densities, and leading to PLATO crashing the program by reporting an error. This positivity problem, thus, is unsolved in this

context, and is a clear research gap, and the target of this thesis.

Closing this research gap would possibly enable higher order simulations with thermochemical nonequilibrium to be used for hypersonic cases with COOLFluiD; thus enlarging the usability scope of COOLFluiD for research and design. These higher order simulations would enable the use of coarser grids for cases of interest, as well as reduce the required care for refining the mesh near the shock wave as much as needed for finite volume. The work of Vandenhoeck [2] shows several converged and validated supersonic cases where the shock wave is present and well defined, supporting this idea.

This research gap and motivation are tackled thoroughly in the literature review chapter (chapter 2). Further information regarding the working of FR, an overview and motivation for the new chemical source term model PLATO being used, as well as the reaffirmation of the positivity issue in higher order methods are given also in chapter 2. Given the pressing need for a new positivity method, while other problems with higher order methods are found and stated, they are not chosen as the target of research. Emphasis is given to the current available positivity methods in literature.

The research questions are given at the end of chapter 2, in section 2.4, after the motivation of the literature review. The research is then divided into three major sections. Each major section concerns four distinct parts: the identification of the problem being solved in specific, the required derivations, the implementation in COOLFluiD, and the testing.

The first major section concerns chapter 3. There, filtering techniques are tackled and implemented in COOLFluiD based on already in development code. It is focused on the Euler model case. Research questions regarding the effectiveness and effect of these filters are given. Testing is done against a well established test case present in the COOLFluiD code base, testing two different filtering techniques against the previous method for positivity. The analysis focused on possible negative effects of the filters and on whether they managed to keep positivity as well as the previous method. This first section opened a path for integrating further filters for more complicated models, such as ones with multiple species and multiple temperatures.

The second major section concerns chapters 4 and 5, and focuses on the positivity for thermochemical nonequilibrium cases. Chapter 4 derives an original new method for positivity of the densities for simulations of any order. Furthermore, it derives new entropy conditions for filtering for these new models. The implementation in COOLFluiD is motivated and given for both the new method and for a filtering method. Chapter 5 focuses on presenting the research questions regarding these positivity methods as testing them against two test cases which were motivated during chapter 2. These cases are run in order 1 (P0) and order 2 (P1) and validated. Given problems found for P1 cases, a third major section was motivated.

The third major section encompasses chapters 6 and 7. Chapter 6 focuses on analysing the source code for flux reconstruction based on motivations given in chapter 5. Problems found in the code base which were not known are presented and corrected. These problems concern specifically the cases with thermochemical nonequilibrium which were sought to be run in chapter 5. Chapter 7 focuses on redoing the P1 runs with the goal of answering the previously defined research questions.

The methods used for testing concern test cases from literature which can be validated against reported values. The setups of the cases, including initial values and boundary conditions, are justified from reported values from literature. Mesh refinement studies for the thermochemical nonequilibrium cases

were not used, opting instead for attempting to run the simulation at successively higher orders, and with validation done against reported values. Mesh refinement was used for the Euler model case as no reported values were given, and based on shock wave position and jump rather than wall quantities, given the nature of the case chosen (a wedge with shock reflection).

The last two chapters concern future work (chapter 8) and the conclusion (chapter 9). The future work is based on still present issues with the P1 simulations as well as future improvements and possible to be implemented methods based on the literature review.

2

Literature Review

2.1. Introduction

A hypersonic vehicle encounters an extreme environment characterised by high levels of heating, and strong shocks [7], [9], [10] , as well as possibly thermal and chemical nonequilibrium and ionisation [6]–[8], [10]. Space missions such as planet reentry include vehicles entering planets' atmospheres at hypersonic speeds [9], and are required for retrieving astronauts or equipment from orbit – such as crews from the international space station, as in the Dragon Capsule [11], or current technology tests such as the recent use of the Orion Capsule [12] – or for entering other planet's atmospheres for scientific missions. Of the latter, one might include entering Mars' or Jupiter's atmosphere [9]. Currently, interest in hypersonic reentry might be found from uses for Moon missions with the Artemis Mission, as shown by the Orion Spacecraft [12], or Starship developments by Space X, which has already survived reentry during its fourth test flight [13] – reentry in which the effect of radiation due to the high temperatures can be clearly seen [13]. Evidently, further missions that might require reentry at high speed into an atmosphere would benefit from research in hypersonic aerodynamics.

The aforementioned physics involved lead to high heat fluxes, which require attention during the design of a hypersonic vehicle [7], [9]. Furthermore, Bose, Brown, Prabhu, *et al.* [9] speak of degrees of ionisation of ~10% of the flow and of high radiative heating in addition to the heat transfer due to convection for the case of lunar trajectories. These conditions can lead to the requirement of ablative heat shields, which in themselves can lead to sooner transition to turbulence in the flow due to their rougher surface. Notwithstanding the focus on uncertainty in the modelling associated with these physics by Bose, Brown, Prabhu, *et al.* [9]'s comments, these comments also reveal the impact on design and the dire conditions for the vehicle. Bose, Brown, Prabhu, *et al.* [9] further comment on the hyperbolic trajectory used to reenter Earth, with entry speeds of ~12-16 km s^{-1}, and how it leads to highly coupled phenomena and high degrees of ionisation.

The above introductory remark on the physics involved shows the need for testing and modelling these phenomena in order to have successful missions with planetary reentry. Viviani, Aprovitola, Pezzella, *et al.* [14] comment on the importance for Computational Fluid Dynamics (CFD) on the

design of hypersonic vehicles, both in predicting physical conditions and in supporting experimental campaigns. They further advance how CFD integrates several stages of the design cycle, from backing up Engineering-Based Models in early stages, to predicting complex flow phenomena to which hypersonic vehicles are susceptible, to aiding in the testing phase. Furthermore, they argue that it has benefits in aiding the development of new designs as well as lowering the cost of design. Bose, Brown, Prabhu, *et al.* [9] emphasised the need of CFD for designers due to difficulty of obtaining high loads experimentally on ground facilities, and commented on the impossibility of ground facilities recreating all the phenomena found during the hypersonic flight simultaneously. Vandenhoeck [2] emphasises from the beginning that on-ground experiments and test flights cannot give all the required design information with a reasonable cost and time-frame, while counterposing accurate modelling and CFD as of high importance for the design of heat shields and aerodynamic control surfaces. As such, CFD comes as solution for design of hypersonic vehicles and for the study of hypersonic flows.

However, despite these advantages, given its numerical nature, CFD requires modelling of the above physics. Furthermore, it requires numerical modelling of the theoretical model, which might lead to numerical effects. Textbooks such as [15]–[17] include common methods used in numerically solving fluid related equations, and address the stability of the methods, as well as numerical errors, such as numerical viscosity, for example. While LeVeque [15] focused on presenting finite volume (FV) methods, Abgrall and Shu [16] and Marsden, Sirovich, and Antman [17] focus largely on higher order methods. The concept and definition of higher order methods is addressed and developed upon in section 2.3. These textbooks, thus, emphasise the foundational care in developing and applying numerical methods in order to extract their usefulness.

The aforedescribed physics require their inclusion in models in order to be taken into account. As such, several models can be used to try to simulate the same phenomena. Intuitively, incorporating more physics might lead to a more accurate result, but at a cost of a more complex and longer computations; assuming that the inclusion of these physics does not destabilise the model, and that they are sufficiently well modelled. Theoretical models used will be touched upon in section 2.2, with some emphasis on the chemical models.

Both theoretical and numerical modelling, hence, are required to achieve the required "accurate modelling" [2]. Several modelling choices and implementation can, then, lead to more accurate or faster results, or even enable previously impossible cases.

The current research is being done in the context of developing and testing the CFD solver Computational Object-Oriented Libraries for Fluid Dynamics (COOLFluiD) [4] for hypersonic flows with chemistry using higher order methods. While the text above, despite brief, has sufficiently contextualised and motivated the need for developing CFD software capable of accurate simulations in a reasonable time frame, while capturing the essential physics, the need for higher order methods will be addressed in section 2.3. There, not only will they be introduced and motivated, but their application be clear, and their concept explained in greater detailed. COOLFluiD is introduced in section 2.1.1, while the organisation of the literature review is given in 2.1.2.

2.1.1. COOLFluiD

Computational Object-Oriented Libraries for Fluid Dynamics (COOLFluiD) is an open source software designed for "scientific high-performance computing, CFD and multi-physics applications" [4]. It is modular in nature [4], [5], allowing for flexibility and reuse of modules. As such, it is possible, for

example, to develop new physical models, and incorporate them with numerical methods already present in the software, from time marching schemes, to discretisation techniques [5]. This modular nature can be further seen in [2], where a clearly separated structure is presented, showing separately the physics modules, the time marching schemes and the spatial discretisation method.

Regarding capabilities, COOLFluiD has been shown to simulate hypersonic flow without chemistry [2], and with chemistry [3], [6]. It currently includes finite volume discretisation [6] as well as the higher order method flux reconstruction [2], [3], including on simplex elements [18], allowing unstructured grids. Currently, COOLFluiD also allows to run the simulations in parallel, both on CPUs, and on GPUs [2], [4]. However, not all modules are available for GPUs due to limitations in the use of classes in C++ [2].

While the above shows how COOLFluiD can be and has been used for hypersonic flows, it should also be noted briefly for completeness that it can also be used to model plasma – both with magnetohydrodynamics (MHD) [2], [19] as with multiple species in low ionised plasma [6] – as well as several RANS turbulence models [2].

Given the current research in extending the COOLFluiD to include non-local thermodynamical equilibrium [6], and the presence of the higher order method flux reconstruction, there is a clear interest in have these physical effects working with this method. However, higher order methods can be more unstable and less robust than FV. A clear example, is the need for a artificial diffusion coupled with a shock detector and a positivity preservation method in [2] to allow the solver to converge to the steady state solution.

Regarding higher order methods, COOLFluiD has used flux reconstruction (FR) – further presented in section 2.3 – to simulate hypersonic cases [2], flows with chemistry [3], and plasma [2]. As such, the current research with COOLFluiD points towards a path of using FR with appropriate chemistry models to simulate hypersonic flows. While this is the focus of this review, the specific gap of the research found is only presented in section 2.4.

2.1.2. Organisation of the Literature Review

The focus of this review, as explained in section 2.1.1, is to review literature related to simulating hypersonic flows with chemistry with flux reconstruction, namely in the context of doing such simulations in COOLFluiD. As a consequence, the physics that require capturing are presented and reviewed in section 2.2. Higher order methods, including flux reconstruction, are reviewed in 2.3, with the goal of identifying the gaps in research in hypersonic flows with chemistry with higher order methods, with special interest in flux reconstruction. Given the similarities in problems between higher order methods, several are reviewed in order to grasp the current research. Solutions to these problems are also presented in this section, including ongoing efforts and trends. Such is done always in the context of what already is or can be implemented in COOLFluiD, and in the context of what physics require capturing. The problem found is clearly identified in section 2.4.

2.2. Reentry Physics and Chemistry

As aforementioned in the introduction, reentry vehicles can encounter drastic conditions during reentry, including high temperatures and pressures. A brief overlook of the typical conditions and physics to be captured is presented here for motivation and contextualisation of the research.

2.2.1. Blunt Body Reentry – a brief overlook

For hypersonic reentry, blunt bodies are usually used [7], such as the Apollo capsule [7] and the more recent the Orion capsule [20]. During reentry for blunt bodies, a detached shock wave appears in front of the body, as in figure 2.1. Behind the shock wave, velocity drops (in the direction perpendicular to the

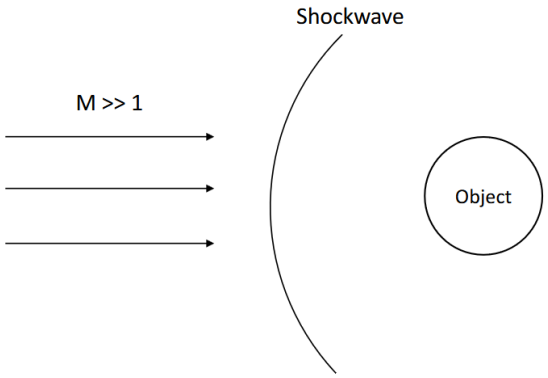


Figure 2.1: Detached shock wave appearing in front of a blunt body when faced with a hypersonic flow (based on information from [7]).

shock), while density, pressure and teperature rise [7]. Temperature can rise to high values, such as in the case of the reentry of the Apollo capsule, which registered temperatures of 11000 K [7]. The interest of these high values is further emphasised by works such as the review by Gupta, Yos, Thompson, *et al.* [8], where temperatures up to 30000 K are considered.

Behind the shock wave, there is a subsonic zone – associated with the normal part of the shock wave – and a supersonic one closer to the edges, more oblique part of the shock wave [7]. This qualitative description emphasises that a large range of velocities is to be expected even in inviscid simulations.

Boundary layer studies are of high importance for hypersonics, as they are closely connected with the study of heat transfer – one of the important design quantities of any reentry vehicle [7]. However, as it will be further seen, such is not the focus of this review and they will not be addressed.

An expansion fan is commonly found at the edge of the bowed part of the reentry vehicle [7]. Thus, this zone is expected to be captured in simulations, and will be impactful on the conditions behind the vehicle.

The above qualitative and brief description emphasises the importance of the shock wave for the flow conditions, as well as the need to accommodate a large range of values in models and simulations. In section 2.2.2, the consequence of the drastic jump in properties behind the shock wave is addressed, namely the reacting nature of the flow. The rest of the current section seeks to provide a context of the expected conditions of interest for hypersonic flows.

2.2.2. High Temperature and Chemistry Models for Reentry

Sharma, Giangaspero, Munafò, *et al.* [6] focused on validating a thermochemical nonequilibrium CFD solver by integrating the PLAsmas in Thermodynamic nOn-equilibrium (PLATO) library [21] with the FV code of COOLFluiD [5], [6]. In it, they motivate the importance of the research claiming that, due to a strong shock wave during reentry, the high temperatures give rise to finite rate phenomena related to chemical reactions – namely ionisation and dissociation – as well as how the energy is stored

and transmitted between particles. This finite rate, due to being near the same order as the time it takes the flow to go across the domain, leads to the gases not necessarily reaching equilibrium during their motion. As such, it is important to take into account how these states change in time, instead of assuming an equilibrium state. This need is further emphasised in reviews such by Gupta, Yos, Thompson, *et al.* [8], or by the fact that it is taken into account in introductory textbooks such as [7].

The nonequilibrium of the chemical species due to ongoing chemical reactions is dubbed chemical nonequilibrium. The nonequilibrium due to energy modes is dubbed thermal nonequilibrium. When both coexist, it is termed thermochemical nonequilibrium (TCNEQ), which is the case used by Sharma, Giangaspero, Munafò, *et al.* [6]. The term non local thermodynamic equilibrium (NLTE) is used in [21] to denote cases where the flow is not locally in thermodynamic equilibrium, as it is usually assumed in low temperature and low velocity aerodynamic applications. Instead, nonequilibrium states are found locally and must be taken into account in the formulation of the model. This is on par with the nonequilibrium states described. All four of these terms are found in literature for hypersonics.

Gupta, Yos, Thompson, *et al.* [8] presented a review on reaction rates and transport properties for air up to 30000 K, using an 11 chemical species model. They emphasise how air around at hypersonic vehicles, in certain regimes – including reentry – can reach temperatures up to this order of magnitude, bringing not only high heat loads, but also inducing chemical reactions such as dissociation and ionisation, in accordance with the aforementioned information. These chemical reactions, by removing energy from the flow, lower the temperature of the flow [7]. This can change heat and pressure loads on the flow, as well as the location of the shock wave [7].

Another change is the need for more than one temperature, since the internal energy depends not only on the temperature associated to the translational velocity of the particles, but also other energy modes such as the ones associated with rotations and vibration of the particles – modes which, at high temperatures, are no longer negligible [6], [8]. Consequentially, how these modes change and influence the internal energy becomes important. Also regarding temperature, different species might have different temperature. The most egregious example is that of the electron, after ionisation, as, due to it being light and highly mobile, might have a much higher temperature that heavier particles [8]. This changes make chemistry essential in modelling at high temperatures in order to capture the fundamental physics of the flow.

Besides chemistry, another high temperature effect is that the transport properties might not follow common relations that are valid at lower temperatures. Viscosity, at high temperatures, no longer follows Sutherland's law, and requires either curve fitting or careful modelling. For the thermal conductivity, the same applies. Given the dependence on temperature, it becomes more convoluted in the case of multiple temperatures.

As such, it is important to chose the correct model for the correct application. While low temperature cases can be modelled with single temperature and without these considerations, high temperature cases can be modelled differently in terms of these properties depending on the application.

In terms of chemical species, Gupta, Yos, Thompson, *et al.* [8] presents a model with 11 species for air, asserting that these are enough for the application such as reentry. This model is still used, and referred to in contemporary reviews [22], [23] in the context of blunt body reentry. Peyvan, Shukla, Chan, *et al.* [22] also makes note of 3 and 5 species models.

2.2.2.1. Basic Formulation

Regarding the formulation of the flow equations, they follow a model of a conservation equations with source terms, as such:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \dot{\boldsymbol{\omega}}, \tag{2.2.1}$$

with $\mathbf{U}$ being the vector of the conserved variables, $\mathbf{F}$ the matrix of fluxes, and $\dot{\boldsymbol{\omega}}$ the vector of source terms. The high temperature models focus not only in changing the primitive variables required (such as additional temperatures) [8], but include, as previously stated, models for the transport coefficients. These coefficients will appear in the formulation of $\mathbf{F}$, namely for diffusive and heat fluxes [8]. The chemistry models will mainly influence the $\dot{\boldsymbol{\omega}}$ term, as they will include source terms such that chemical reactions can transfer mass and energy between species [8]. They also impose additional mass conservation equations – one for each species – hence adding conserved variables [8].

The mass conservation equations – part of equation 2.2.1 – follow the following form:

$$\frac{\partial \rho_s}{\partial t} + \nabla \cdot (\rho_s \boldsymbol{v}) = \dot{\omega}_s, \quad \text{with} \sum_{s=0}^{N_s} \dot{\omega}_s = 0\,, \tag{2.2.2}$$

with ρ_s the density of the species s, $\boldsymbol{v}$ the velocity of the species, and $\dot{\omega}_s$ the source term for the species. Due to overall conservation of mass, the source terms must add to zero. Using ρ, it is possible to redefine these equations in terms of mass fractions instead of densities [21]. Using the molar mass of each species allows the reformulation of this equation in terms of number of moles, or in number fractions [21].

The velocity $\boldsymbol{v}$ is usually taken as $\mathbf{v} + u_s$, with u_s being the diffusion velocity of the species, and $\mathbf{v}$ the velocity of the fluid element.

The additional equations for fluid mechanics are usually a momentum conservation and an energy conservation equation. The momentum equation follows the following form:

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho_s \mathbf{v}\mathbf{v} + pI) = \nabla \cdot F_D + F_F + \dot{\omega}_M, \tag{2.2.3}$$

with pressure p, F_D the diffusive fluxes, F_F other forces per unit volume, and $\dot{\omega}_M$ the source terms for the momentum equation. The energy equation for a fluid element follows the following form:

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho_s H \mathbf{v}) = W + Q + \dot{\omega}_E, \tag{2.2.4}$$

with total energy E, total enthalpy H, work per unit volume and per unit time W, heat flux Q and source terms $\dot{\omega}_E$. The energy equation for a single species (required in certain cases, such as when decoupling electron energy [21]), has the following form:

$$\frac{\partial \rho_s e_s}{\partial t} + \nabla \cdot (\rho_s e_s \mathbf{v}) = W_s + Q_s + \dot{\omega}_{es}, \tag{2.2.5}$$

with internal energy e, the index $_s$ indicating that it is for the species s only, and source terms for this equation $\dot{\omega}_{es}$.

These equations differ from the Navier Stokes equations in the fact that they include several species and source terms. The exact terms on the RHS of the equations depends on the model and the required physics which must be captured, and it is the focus of this section. It will be seen that several models

are available for these terms, and that while some allow faster or simpler computations, for the case of hypersonic flows, complicated models based on reacting gas dynamics are required.

2.2.2.2. Chemical Nonequilibrium

Chemical nonequilibrium is, as previously stated, essential to model high temperature flows in hypersonics [7], [8], [21]. Dissociation and ionisation take place above certain temperatures. Gupta, Yos, Thompson, *et al.* [8] presented a figure (figure 1 in [8]) for the different nonequilibrium regimes during reentry of a small sphere. This figure is adapted from a shuttle reentry related figure, and clearly shows how during a large portion of the descent chemical reaction are present, and presents at which combinations of altitude and speed each of the main reactions are present.

Anderson [7] presented an introductory text to chemical nonequilibrium in hypersonics, emphasising the role of both reaction rates and transport coefficients. With chemical reactions, the change of amount of a species is modelled based on reaction rates k and species concentrations, giving rise to the coefficient $\dot{\omega}_s$. In the case of elementary reactions, the reactions are proportional to the species concentrations to the power of their stoichiometric coefficient. While the amount of the species is directly taken into account in equation 2.2.2, the reaction rate requires modelling.

For clarity, an simple example is given. Given species A and B, and the elementary reaction $A \leftrightarrow 2B$, the reaction could be modelled by:

$$\dot{\omega}_A \equiv \mathcal{M}_A \frac{\mathrm{d}A}{\mathrm{d}t} = k_{AB}[A] - k_{BA}[B]^2, \tag{2.2.6}$$

with $\mathcal{M}_A$ being the molar mass of species A, k_{AB} the forward reaction rate, k_{BA} the backward reaction rate, and the square brackets denoting concentration in moles per unit volume [8]. In general, a chemical reaction is given by:

$$\sum_s^{N_s} \alpha_{\mathrm{stoch},s} X_s \leftrightarrow \sum_s^{N_s} \beta_{\mathrm{stoch},s} X_s, \tag{2.2.7}$$

where N_s is the number of species, X is a species, and α_{stoch} and β_{stoch} are the stoichiometric coefficients. For systems of reactions, all the contributions of each reaction must be taken into account as follows:

$$\dot{\omega}_s \equiv \mathcal{M}_s \sum_r^{N_r} \left(\frac{\mathrm{d}X_s}{\mathrm{d}t}\right)_r = \mathcal{M}_s \sum_r^{N_r} (\beta_{\mathrm{stoch},rs} - \alpha_{\mathrm{stoch},rs}) \left(k_{f,r} \prod_j^{N_s} \left(\frac{\rho_s}{\mathcal{M}_s}\right)^{\alpha_{\mathrm{stoch},j,r}} - k_{b,r} \prod_j^{N_s} \left(\frac{\rho_s}{\mathcal{M}_s}\right)^{\beta_{\mathrm{stoch},j,r}} \right), \tag{2.2.8}$$

where the index $_r$ indicates the reaction, and N_r is the number of reactions, $k_{f,r}$ and $k_{b,r}$ the forward and backward reaction rates for reaction r [8]. This formulas states that the changes due to each reaction must be summed over to get the total change. For further details, the reader is directed to [8].

The forward and backward reaction rate are usually computed with the modified Arrhenius equation [8]. This equation depends on the temperature and empirical factors, and is specific for each reaction. It also clarifies the dependency on temperature of the reaction rates. However, for multiple temperature formulations, the use of this relation is not straightforward. Gupta, Yos, Thompson, *et al.* [8] cited, at the time, different formulations of this expression to take into account for multiple temperatures, including using a geometric average of the translational and rotational temperatures, as done by Sharma, Huo, and Park [24].

Munafò and Panesi [21] circumvent this problem for their model when taking into account reactions

where a heavy particle and an electron collide. A rate can, then, be computed for the electron temperature, and the other rate can be acquired through a relation – through the principle of micro-reversibility – hence acquiring the necessary rates, without the use of the Arrhenius expression. In general, Munafò and Panesi [21] use the Chapman-Enskog (CE) method to acquire the source terms $\dot{\omega}_s$ (as well as $\dot{\omega}_E$). This method is derived from reacting gas kinetics [25], and already mentioned by Gupta, Yos, Thompson, *et al.* [8] in the context of transport coefficients. This treatment is more physical than that of the Arrhenius relation, which requires empirical coefficients.

However, Munafò and Panesi [21] also include the Arrhenius relation in their model, as seen in its use by Sharma, Giangaspero, Munafò, *et al.* [24], and the inclusion of the function in the code in [21].

The use of Arrhenius relations, however, has its advantages, such the possibility to derive analytical expressions which are more computationally efficient, assuming that the empirical factors can be written in an analytical expression too (such as a polynomial acquired through curve fitting) [26]. However, the CE method allows one to obtain what Munafò and Panesi [21] claims to be the "most rigorous theoretical framework" for deriving the equations for reacting plasmas, despite its complexity. Comparisons between these methods in terms of affecting simulation was not found, but it could be speculated that convergence and computation time could be affected by the more complicated method. The difference in physicality in validation simulations was also not found to be addressed in literature.

The Plato model described by Munafò and Panesi [21] has been successfully used in the context of COOLFluiD by [6] with an 11 species model [6], [8], for a FV scheme while simulating hypersonic regimes. Sharma, Giangaspero, Munafò, *et al.* [6] used the Arrhenius expression with a rate controlled temperature in order to acquire the rate coefficients directly from Plato. A rate controlled temperature is a geometric average of the translational and rotational temperature. From a theoretical standpoint, the description of chemical reactions in [6] is the same as in [8], emphasising the still importance and relevance of the latter.

2.2.2.3. Species Models

Species models take into account the number and type of species, as well as the reactions between them. PLATO considers the 11 species Parker model which comprises N2, O2, O, N, NO, NO+, O2+, N2+, O+, N+, e^- [21]. The modelling of the reaction between these species depends on the model itself. Plato contains a wide range of reactions which can be found in the work by Munafò and Panesi [21].

A 5 species model as also been used in the context of hypersonics for the high Earth atmosphere. Such was the case of the studies with the High Enthalpy Shock cylinder, where N2, N, O2, O and NO were considered (thus not taking into account ionisation) [27]. Depending on the case considered, ionisation might not be required to be modelled, thus making this model better suited. Reactions for different altitudes and velocities can be found in the appendix of the work of Gupta, Yos, Thompson, *et al.* [8].

Two species models, such as the N2 and N model, can also be of interest, namely in validation of numerical codes [27].

2.2.2.4. Non local thermal equilibrium

At high temperatures, thermal nonequilibrium can take place. It is characterised by the necessity of using several temperatures to model the flow. If several species are not in thermal equilibrium, each must be assigned its own temperature. Such is the case for the two temperature model, where electrons and heavy particles are assigned different temperatures due to the high mobility of the electrons [21].

For the same species, different temperatures might also be needed. Such is the case when the rotational and vibrational energy modes are not negligible for the internal energy and, as such, must be accounted for [8]. In these cases, a rotational and a vibrational energy are defined [8].

The use of these temperature models for the calculation of internal energy can be found in [8] for the non thermal equilibrium case. They make use of thermodynamic property tables to acquire the values for C_p for each temperature mode, and explain how they can be couple to calculate enthalpy. They also advice about how the method that they employ cannot be trivially extended to more than two temperature models. More information on using several temperatures for the calculation of the internal energy and enthalpy can be found in [25]. Munafò and Panesi [21] make use of the first order CE method to calculate the internal energy contributions without making use of C_v nor C_p, but rather analytical expressions based on the species temperatures. This method can be found more detailed in Nagnibeda and Kustova [25]. This method is based on the rigid rotor and harmonic oscillator models for diatomic particles, also described in [25]. These methods are also mentioned briefly in [8] in the context of the work of then contemporary authors.

The transport coefficients for diffusion, heat flux and viscosity are also contemplated differently in non local thermal equilibrium. In single temperature models at low temperature for local thermal equilibrium flows, more typical in aerodynamics, these can be related to the gas and temperature, or even taken as constants, in certain conditions. Curve fitting these can also be done. The viscosity is often, in these cases, related to temperature through the Sutherland formula. Formulas derived from gas kinetics, namely through CE methods, can also be used.

However, the simpler approaches breaks down at high temperatures for chemically reacting flows, and requires special treatment [8]. Sutherland's formula, for instance, is no longer valid. As such, these transport properties are obtained through other means, such as solving a linear system of equations dependent on the temperatures, the species taken into account in the flow, and gas kinetics related expressions such as collision integrals when applied in the work of Munafò and Panesi [21]. They can also be obtained directly in certain multiple temperature models, such as in [8], where they depend on temperature, the species taken into account and tabulated values for the C_p in function of temperature.

Tabulated values are essential for the use of the more physical approximations based on gas kinetics for the transport coefficients, be it C_p in [8] or the collision cross-sections in [21]. Regardless of these limitations, it is important to note that these models manage to better the results of high temperature flows present in hypersonics, when chemical reactions and thermal nonequilibrium are present. It is also important to note their utmost need due to the conditions for the simpler models breaking down with thermal non equilibrium.

2.2.3. Models

Depending on the exact physics that must be taken into account, different forces and phenomena need to be taken into account. For the cases described above other terms beyond the typical terms included in the Navier-Stokes must be taken into account.

2.2.3.1. Diffusion

Diffusion, for instance, must be taken into account directly, as in equation 2.2.2, where the diffusion velocity is u_s, with diffusion being modelled in the typical way with Fick's Law. The coefficient for this law might be calculated with procedures mentioned in section 2.2.2.4.

2.2.3.2. Heat Transfer

Heat transfer can be modelled differently. While a typical formulation might follow Fourier's law, further heating terms are required for the heat transfer. Munafò and Panesi [21], for example, include in the heat transfer terms the formation enthalpies associated to the different temperatures. Thus, chemical reactions, by changing the amount of a species, can be taken into consideration in their impact on enthalpy. Formation enthalpy could possibly also be taken into account in the enthalpy term directly in the energy equation, instead of grouping it with the heat transfer term.

These heat transfer terms are also different for each species, as they depends of several temperatures in multiple temperature formulations. Thus, since atomic species do not have rotational temperature, for instance, such term would not appear in their expression.

The heat flux coefficient can be calculated with procedures already mentioned in section 2.2.2.4.

2.2.3.3. Viscous Terms

The viscous terms are included by Munafò and Panesi [21] in the usual way, through Newton's Law. Other approaches were not found in literature. What can be changed, however, is the way the viscosity is calculated. These methods were already discussed in section 2.2.2.4, and can range from taking constant values, to interpolating from tables, to functions of temperature – such as Sutherland's Law – to using CE methods. For non local thermodynamical equilibrium, the CE method would be the most physical way of doing so, and most appropriate due to this nonequilibrium nature.

2.2.3.4. Additional Forces

The Navier-Stokes include pressure and viscous forces (the latter already mentioned in this chapter). In the case of NLTE plasmas (with free electrons and high enough degrees of ionisation), it is important to take into account the Lorentz force, which must be included both the momentum and energy equations (the latter in the form of work performed). This force implies the computation of the currents (movement of charged particles) in the flow, and of electric and magnetic fields. Furthermore, the generalised Ohm's law must also be included in the energy equation to take into account heat dissipation.

The formulation of these electromagnetic forces can be found in the description of the conservations equations in the works of Munafò and Panesi [21].

Body forces such as gravity are usually not included.

2.2.3.5. Source terms

The source terms for the mass conservation equation represent the exchange in mass during a chemical reaction, and were already tackled in section 2.2.2.2. For the momentum equation, these source terms are meant to take into account how the momentum changes with interaction between particles. Given that the momentum is usually written for the fluid element, and not for each species, the global change in momentum is what is sought in these terms. While Munafò and Panesi [21] makes use of them, they are not given. In the use of PLATO by Sharma, Giangaspero, Munafò, *et al.* [6], these terms are dropped all together, retrieving the Navier-Stokes formulation for the momentum equation.

For the energy equation, the source terms include both exchanges in energy during particle interactions, as well as the change in time of energy modes, such as the vibrational energy mode. The change in energy for electrons due to interactions is given explicitly in [21], but no other terms are given clearly. Sharma, Giangaspero, Munafò, *et al.* [6] made use of a model for the change in time of the vibrational energy. A model with exponential decay for this energy can also be found in [7].

2.2.4. Concluding Remarks on High Temperature and Chemical Models for Reentry and Possible Future Research

Chemical and thermal nonequilibrium are present largely in hypersonics flows, and must be taken into account in order to simulate these flows, namely in the case of reentry. It was found in literature that these phenomena alter both the overall flow – such as by altering shock wave position – as well as the thermodynamic variables of importance – such as pressure and temperature – as well as the loads which might be important to calculate – such as thermal and pressure loads. Furthermore, it was found that this nonequilibrium changes the relations and dependencies of the transport variables, such as the ones for diffusivity, viscosity and heat transfer, implying new more complex ways to calculate them grounded on the theory of kinetic gas dynamics. It was also found that, in the case of ionisation, electric fields can be of importance, as well as treating electrons separately.

The above remarks imply that the choice of model is of most importance, in order to capture the essential physics. PLATO [21] – which has already been used with COOLFluiD in a FV formulation – can be found as a good model to use for simulation of these flows, given that it has already been integrated. It has, however, not been used with higher-order methods, for chemistry has only been modelled in the context of FR methods in COOLFluiD with the Mutation library [3], which does not include such a refined method for the calculation of the transport coefficients, nor the inclusion of electric fields.

This seems to suggest that it would be beneficial to use this model with higher-order methods, such as FR, in COOLFluiD. While other models could be used, or a model could be built from the ground up, the already present coupling of PLATO with COOLFluiD [6] facilitates the work of simply treating the physical model with FR instead of FV; which is not only easier than building and validating a new model, but also makes more sense logically from the standpoint of building on previous work.

The basis and use of higher-order methods is presented in section 2.3, with emphasis on their struggles, possible solutions and possible use for reacting hypersonic flows.

2.3. Higher Order Methods

The spatial order of a numerical method is k_{mesh} if the error is proportional to $h^{k_{\text{mesh}}}$, with h being the mesh size [28]. Higher order methods are understood as methods with spatial order of accuracy of 3 or more [28].

Higher order numerical methods allow for more accurate solutions for the same computational cost as lower order methods, at the cost of the robustness and lower complexity of the latter [29]. This lack of robustness makes it harder for them to be as well established as lower order methods, such as FV methods. Peyvan, Shukla, Chan, *et al.* [22] also backs these claims, and further comments how, in the case of hypersonics, low order methods can encounter the carbuncle effect on the shock wave, as well as wrong predictions of the surface heating, the latter of which is of importance for hypersonic vehicles. They further comment on the dependence on the grid for low order methods, and their failure to resolve the boundary layer accurately for turbulent flows at high speeds. Higher order methods, they explain based on literature, could possible resolve these issues, as well as be suited to simulate hypersonics with real chemistry. As such, high order methods could possibly overcome the problems related to low order methods in hypersonic flows, including with chemistry.

The lack of robustness in high order methods requires some treatment, such as positivity preservation

of the terms, due to numerical oscillations [22], shock capturing [2], or the use of artificial viscosity to dampen oscillations [2]. These and other treatments are further explained and expanded upon in section 2.5.

High order methods currently used for CFD include the Discontinous Galerkin (DG) [29], [30], the Spectral Difference (SD) [29], [31], Flux Reconstruction (FR) [1], [29], [32]–[34], and Spectral Volume (SV) [29], [35], as well as Essentially Non-Oscilatory (ENO) [36], and Weighted Essentially Non-Oscillatory (WENO) [37]. While DG is not necessarily the focus of this literature review – as FR methods were introduced in section 2.1.1 as the current ongoing research – DG methods have a long history of applications and can yield important insights and solutions to typical problems arising in higher order methods, such as when dealing with instabilities [23]. As such, they remain relevant for the study of FR applications. These are further presented in section 2.3.1, together with relevant information regarding other the other aforementioned methods. They are also several times taken into account in section 2.3.4.

FR is a more recent method introduced by Huynh [32]. As stated before, it is the focus of this literature review. It is presented in section 2.3.2.

2.3.1. Discontinuous Galerkin Methods and Hypersonic Flows

The most well established, and oldest, aforementioned higher order method is the DG method, introduced by Reed and Hill [30] for the case of the neutron transport equation. Despite this initial application, it has been used for the cases with fluids in CFD – such as for conservational laws [38], the Euler equations [39], the Navier-Stokes equations [40], the Shallow-Water equations [41], to hypersonic flows [23], and multicomponent reacting flows [42], as well as with turbulence models, such as with Large Eddie Simulations (LES) [43].

Hoskin, Van Heyningen, Nguyen, *et al.* [23] did a survey of DG methods mostly focused on hypersonics. However, this recent (2024) review also includes a survey of the state of the art of the DG methods, together with mentioning main papers on grid refinement and error estimations for DG [44]–[47], the inherent advantageous in the way it discretises the convective operator [48], [49], some applications, the ability to be used in unstructured 3D meshes, as well as the current trend of using hybridised discontinuous Galerkin (HDG) methods [49], [50], including the more recent embedded DG method [51]. Extensions to the DG methods include the aforementioned HDG, the Local Discontinuous Galerkin and the Compact Discontinuous Galerkin [2]. Basic information about these methods can also be found in the books by Abgrall and Shu [16] and by Hesthaven and Warburton [52]

Hoskin, Van Heyningen, Nguyen, *et al.* [23] further explores methods of treating instabilities, and shockwaves in the flow, typical in hypersonics, and further important for higher order methods. Numerical methods to solve the systems given by these methods are also presented.

While this shows the variety, interest and good foundation in DG methods, for the case of hypersonics, it also includes the associated problems with the robustness. Peyvan, Shukla, Chan, *et al.* [22] identified these as shockwave capturing, preserving the positivity of the species in multispecies reacting flows, and near vacuum conditions for the case of hypersonics with chemistry and DG methods, while Hoskin, Van Heyningen, Nguyen, *et al.* [23] identified mainly the importance of grid quality, and the numerical treatment of shocks. Regarding research gaps, the following themes were identified in the review: output-based error estimation (which is required for adaptive mesh refinement), turbulence modelling and nonequilibrium effects [23].

2.3.2. Flux Reconstruction and Hypersonic Flow

FR was introduced by Huynh [32] in 2007 as a new higher order method for a conservation equation in 1D. Unlike the DG scheme, it is in the strong form, rather than the weak one. It has been used to simulate in structured and in unstructured grids 2D compressible flows with the Navier-Stokes – including in the hypersonic regime, as well as with RANS turbulent models [2]. Furthermore, it has also been used to study hypersonic flows in 3D with turbulence [2]. Also in high speed flows, it has been target of uncertainty studies as well as its use with r-mesh grid refinement [53]. Outside high speed flows, it has ben studied possible the inclusion of a chemistry model to simulate a reacting flow [3], as well as applications to MHD [2].

A presentation of the method is here given. This presentation should clarify the steps of the method, as well as how the parameters of the method can affect the simulation.

The original presentation by Huynh [32] is not given here, as this one still preceded the introduction of the correction functions formulation by Vincent, Castonguay, and Jameson [54], and the fact that it is was done for a 1D formulation. Instead, the formulation used by Vandenhoeck [2], as it follows a generalisation for higher dimensions, and it more closely follows the current implementation in COOLFLuiD. The alterations required for simplex elements are also presented, as done by Dhib, Ameur, Vandenhoeck, *et al.* [18], as these allow for the use of unstructured grids, which is highly important for the representation of complex geometries.

Flux reconstruction is applied to systems of equations of the form:

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \dot{\boldsymbol{\omega}}, \tag{2.3.1}$$

which is identical to the equation 2.2.1, and only repeated here for clarity. This method, then, is appropriate to solve the typical form of the system of equations found in hypersonic flows. For clarity, is repeated here that: $\mathbf{U}$ is the conserved variable vector, $\mathbf{F}$ is the flux vector and $\dot{\boldsymbol{\omega}}$ is the vector that holds the source terms for the equation.

In the case of the Navier-Stokes equations. In their case, the source terms will be zero. The flux $\mathbf{F}$ includes both advective and diffusive fluxes.

Both the Euler and the Navier-Stokes equations can be written in the form of equation 2.3.1. Formulations for reacting flows and ionised flows, such as the one presented in [21], can also be put into this form.

The FR method focuses on discretising the $\nabla \cdot \mathbf{F}$ term. Once discretised, the source terms can be added and a time marching scheme can be employed to calculate $\mathbf{U}$ in the next time step.

2.3.2.1. Discretisation of the Domain

The domain is discretised into N_l non-overlapping elements Ω_n which occupy the entire domain Ω, which includes its boundary.

The $\mathbf{U}$ is approximated by a function $\mathbf{U}^\delta$, which is made of a set of functions u_k^δ over the entire domain Ω, with k denoting each equation in our system of equations. Given that the flux reconstruction procedure is the same for every equation, the k is dropped from here onwards.

This function u^δ is a collection of polynomial representations u_n of degree P in each element. Thus, u_n is zero outside its element Ω_n, and is a polynomial representation inside of it. Analogously, F is

approximated by F^δ, with a polynomial representation F_n of degree $P+1$ inside each element. This polynomials are allowed to be discontinuous.

More formally,

$$u^\delta = \sum_{n=0}^{N_{\text{pol}}} u_n^\delta \quad \text{and} \quad F^\delta = \sum_{n=0}^{N_{\text{pol}}} F_n^\delta, \tag{2.3.2}$$

where N_{pol} is the number of polynomial representations inside each element.

For both implementation reasons and for the development of the method, it makes sense to transform the elements in the physical domain into a single standard element Ω_S. This allows the use of the same exact procedure for every element. This is done with a mapping function $\Theta_n(\boldsymbol{\xi})$ from Ω_S to Ω_n, which can be different for each Ω_n. While the physical coordinates are given in a vector $\boldsymbol{x}$, in the standard element, they are given in $\boldsymbol{\xi}$.

For the rest of the explanation of the method, the standard element Ω_S is assumed to be 3D cube, which is applicable for 3D structured grids. The derivation is analogous for 1D and 2D grids, and very similar for simplex elements, in which a standard simplex must be defined. Formally, $\Omega_S = \left\{\boldsymbol{\xi} = (\xi, \eta, \varsigma) | -1 \leq \xi, \eta, \varsigma \leq 1)\right\}$.

Each element Ω_n has N_p solution points i. The solution at these points is considered known at the beginning of each time step as they are given by initial conditions. The polynomial u_n interpolates this solution in each time step. The solution points in the standard element are always in the same position, usually given by some point distribution associated with a standard quadrature. In the case of 1D elements, this is the Gauss-Legendre. For 2D and 3D points in structured grid, this distribution is used in each dimension, accordingly. An example of standard elements is given in figure 2.2.

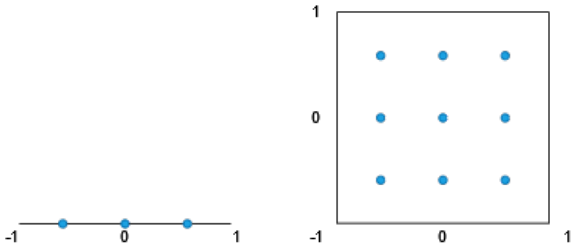


Figure 2.2: Standard elements in 1D and in 2D with a 3 point distribution, based on the work of Vandenhoeck [2]. Solution points are represented by blue dots. Axes are omitted for clarity. The solution point distribution is not drawn to scale.

Let $u_{u,i}$ be the value of the polynomial u_n at the solution point i. This change to the standard element requires that the physical quantities to be transformed:

$$\hat{u}_{n,i}(\boldsymbol{\xi}, t) = J_n u_{n,i}(\Theta_n(\boldsymbol{\xi}, t)), \quad \hat{F}_{n,i}(\boldsymbol{\xi}, t) = F_{n,i}(\Theta_n(\boldsymbol{\xi}, t)), \tag{2.3.3}$$

where J_n is the Jacobian of the transformation Θ_n. Since all calculations will be done in the standard element, the hat and the n index will be dropped, and u_i will denote the solution at the solution point i in the standard element for the rest of section 2.3.2. Analogously, F_i will denote the flux at the solution point i in the standard element also for the rest of section 2.3.2. Since the calculations are identical in

procedure at every time step, there is no index to indicate the time step in question.

Analogously, u will denote the polynomial of the conserved variable in question in the standard element from here onwards; F will denote the polynomial for the flux, also in the standard element.

2.3.2.2. Interpolation of the solution

The solution points, as aforementioned, are interpolated at every time step. To do such, a Lagrangian interpolation is employed as such:

$$u = \sum_{i}^{N_p} u_i l_{\xi i} l_{\eta i} l_{\varsigma i}, \tag{2.3.4}$$

where $l_{\eta,i}$ is the Lagrange polynomial in the η-direction; and analogous for the other directions. Analogously,

$$F = \sum_{i}^{N_p} F_i l_{\xi i} l_{\eta i} l_{\varsigma i}. \tag{2.3.5}$$

2.3.2.3. Interface solution

At the interface, flux points are defined, and denoted with the index j. A solution value u_{jf} at the interface can be calculated with the Lagrange polynomial. The coordinates of these points are easily found by extending the direction of each of the coordinates of the points to the interface.

A common solution between elements can be found by taking the values u_{jf} for each element using them to define a common value u_{Ij} through schemes such as Bassi-Rebay (BR2) scheme.

2.3.2.4. Compute Corrected Solution Gradient

The gradient is essential for the computation of diffusive fluxes at the interface. Given that the flux is discontinuous at the interface, a correction must be made such that the gradient is continuous. As such, the correction function forces that the value used at the interface for the calculation of the gradient be the common interface value u_{If} at the flux point f, while keeping the gradient continuous inside the element.

Hence, the correction is defined as a scalar function along a direction such that at one end is 1 and at the opposite is zero. Let h_f such that:

$$h_f(\boldsymbol{\xi}_f) = 1 \quad \text{and } h_f(\boldsymbol{\xi}_g) = 0, \tag{2.3.6}$$

with f being the flux point associated to h_f, $\boldsymbol{\xi}_f$ being the coordinates of the flux point f, and g the flux point opposite of f. The formulas for these functions can be found in the work of Vincent, Castonguay, and Jameson [54]. Further work with these can be found in [55], [56].

The gradient in the solution point i and in the direction d (that is, in the ξ, η or ς direction) can be defined as:

$$q_{i,d} = \sum_{i}^{N_p} u_i \frac{\mathrm{d}l_{id}}{\mathrm{d}\xi_d}(\boldsymbol{\xi}_i) + \sum_{f}^{N_f} \frac{\mathrm{d}h_f}{\mathrm{d}\xi_f}(\boldsymbol{\xi}_i) \cdot (u_{If} - u_{jf}) n_{df}, \tag{2.3.7}$$

with N_f the number of flux points, η_d the coordinate in direction d, n_{df} the component in the direction d of the normal vector to the element at the flux point f, and l_{id} the Lagrangian basis at solution point i in the d direction.

2.3.2.5. Computation of the Divergence of the Flux

The flux at the solution points $\boldsymbol{F}_i$ was already calculated during the interpolation step. To calculate the flux at the interface F_{jf} at the flux point f, an extrapolation to the flux point can be made, in the same way that u_{jc} was calculated, but with the flux solution. The flux is a function of u_i and of the corrected gradient and, hence, it can be computed at the solution points. Then, the common interface flux F_{If} at the flux point f can be computed. For an advective flux, a Riemann solver can be employed, such as a Roe solver, while for diffusive fluxes, other solvers must be used, such as the BR2.

The divergence of the flux can be then computed directly using the Lagrange polynomial formulation:

$$\nabla \cdot F = \sum_{i}^{N_p} \sum_{d}^{N_d} F_{id} \frac{\mathrm{d}l_{id}}{\mathrm{d}\xi_d}(\xi_{id}) + \sum_{f}^{N} f \frac{\mathrm{d}h_f}{\mathrm{d}\xi_f}(\xi_i) \cdot (F_{If} - F_{jf})\boldsymbol{n_f}, \tag{2.3.8}$$

with $\boldsymbol{n_f}$ the normal vector to the element at the flux point f. The index d indicates that it is a component of the vector in direction d

2.3.2.6. Changes for Simplex Elements

Dhib, Ameur, Vandenhoeck, *et al.* [18] adapted this method for simplex elements. The method above uses non simplex standard elements, which would be the appropriate one for structured grids. Furthermore, the correction functions are 1D functions which can be easily used for quadrilateral elements, but cannot be trivially extended to for simplex elements.

Dhib, Ameur, Vandenhoeck, *et al.* [18] make use of a right triangle with the catheti of side 1 as a standard element. Another change is that the corrected gradient is not computed as previously described but, instead, makes use of a correction field. Furthermore, the correction functions are no longer 1D functions with a closed form, but are given implicitly through a set of equations for which the correction function must be solved for.

An extension to tetrahedral elements (3D) was also given by Dhib, Ameur, Vandenhoeck, *et al.* [18]

2.3.2.7. Remarks on Flux Reconstruction

The FR scheme shows a clear method for the spatial discretisation required to use the method of lines for equations of the form of equation 2.3.1. It has been used for both structured and unstructured grids successfully. Its high order nature is trivially expected from the fact that used several solution inside the element to construct polynomial solutions, with the order increasing with the number of points.

An increased robustness can be found from the way it makes the flux continuous, allowing for better solutions when compared to discontinuous methods. However, this method is also, then, weak to the typical problems that higher-order methods have, such as being more unstable and being prone to unphysical oscillations, requiring special treatment not covered in the definition of the method. The treatment could pass through the use of limiters, artificial viscosity, positivity schemes [2], for example.

These problems should be more stringent for more complicated and sensible systems, such as the ones described in section 2.2.2, where very low but physical mass fractions (near the unphysical negative values) might be encountered, and a high number of coupled non-linear equations exist.

From these formulation, it can be seen that major places for alterations would be either in the conserved states vector, the flux vectors, and immediately before or after the discretisation. Also, additional source terms could be used if required, as these are added after the method.

Cases such as localised Laplacian artificial viscosity (LLAV) [2] would be equivalent to using an additional flux inside this method which is then included in the computation of the fluxes, and it is a prime example of how this method could be used with changes – in this case to the equation that is being solved. Limiters could be used after the time marching, making them mainly decoupled from the method, unlike LLAV. As such, changing the equations – by numerical treatments, changes of variables or even adding physics – or adding numerical treatments to the procedure must take into account the formulation of the method, including where it can be used, how it will be treated by the method, and how it will couple with it.

2.3.3. Artificial Viscosity in COOLFluiD

Higher order schemes require stabilising methods in order to converge. Artificial viscosity allows the addition of diffusive fluxes to the flow in key parts of the flow in order to stabilise the procedure. In particular, it diffuses the shock waves over the elements, attenuating the spurious oscillations present near discontinuities [2].

COOLFluiD makes use of a local Laplacian artificial viscosity (LLAV) to diffuse the shock. This AV is expected to only be present at the shock [2]. A presentation of the implementation is here given in order to allow the discussion of the influence of its parameters and of its influence on the solution.

The LLAV is calculated by calculating a flux equal to:

$$\varepsilon_{\text{AV}} \nabla(\mathbf{u}), \tag{2.3.9}$$

where $\mathbf{u}$ is the conserved variables and ϵ_{AV} is the strength of the AV [2]. The "local" approach is done by calculating this quantity for each point. The Laplacian part is due to the fact that the quantity used in equation 2.3.1 is the divergence of the flux, rather than the AV flux F_{AV} itself, leading to us taking:

$$\nabla \cdot F_{\text{AV}} = \nabla \cdot (\varepsilon_{\text{AV}} \nabla(\mathbf{u})), \tag{2.3.10}$$

including then second spatial derivatives. If the ε_{AV} was a constant, then the equation would be taking the Laplacian of $\mathbf{u}$.

This AV strength requires being calculated in a form that the AV only acts where intended – i.e the shock wave where spurious are present. As such, a detector is used based on detecting these spurious oscillations. Given a measure of smoothness S, one defines [2]:

$$\varepsilon_{\text{AV}} = \begin{cases} 0 & \text{if } S < S_0 - \kappa, \\ \frac{\varepsilon_0}{2}\left(1 + \sin \frac{\pi(S - S_0)}{2\kappa}\right) & \text{if } S_0 - \kappa \leq S \leq S_0 + \kappa, \\ \varepsilon_0 & \text{if } S > S_0 + \kappa, \end{cases} \tag{2.3.11}$$

where S_0 is a user given smoothness, κ is a positive value used to define the interval around S_0 where the κ varies, and ε_0 is the base strength of the AV [2]. The function $\frac{\varepsilon_0}{2}\left(1 + \sin \frac{\pi(S - S_0)}{2\kappa}\right)$ allows a smooth variation from 0 to ϵ_0 with the middle value at S_0 equal to $\frac{\varepsilon_0}{2}$.

The quantity ε_0 is given based on the Péclet number, with ε_0 being the diffusive coefficient, and the Péclet number defined by the user as a way of tuning ε_0 indirectly. It should be noted that these two quantities are inversely proportional and, hence, a larger Péclet number will lead to weaker AV [4].

The smoothness is calculated through the comparison of the highest mode with the other modes in the polynomial description of the solution by taking:

$$S = \log_{10} \frac{\langle u_m - u_m^{P-1},\, u_m - u_m^{P-1} \rangle}{\langle u_m,\, u_m \rangle}, \tag{2.3.12}$$

where u_m denotes the monitored variable at the solution point described in polynomial form and u_m^{P-1} denotes the same variable described in polynomial form but with the last mode truncated out [2]. The brackets do not denote the usual inner product between functions (the L2 norm), but rather the usual euclidean norm calculation applied to each mode [2]. The monitored variable is taken usually as pressure, and is the default in the cases in this document.

2.3.4. Problems in Higher Order Methods for Hypersonics

Following from the problems found in the previous sections, these are expanded upon here, together with solutions found or being developed. They are also reviewed in the context of COOLFluiD and with some attention to their usefullness for FR. Most of the problems focus on what was found when applying the DG methods, due to its popularity and long history. However, DG and FR both share the same problems inherent to higher order methods of non-physical oscillations and trouble with discontinuities. As such, other problems found for other higher order methods are relevant for the development of FR.

2.3.4.1. Shock Capturing

Shockwave capturing for the strong shocks encountered in hypersonics is required in order to acquire accurate converged solutions, as the oscillations typical on higher order methods – which can be big near a shock – can affect the solution drastically or prevent it if no care is taken [2], [23]. Hoskin, Van Heyningen, Nguyen, *et al.* [23] divide the possible methods into three categories: limiters, fitting and artificial viscosity, the latter is divided into physical and laplacian artificial viscosity. In the context of COOLFluiD (see section 2.1.1), laplacian artificial viscosity has been successfully used for shock capturing when paired with a shock detector, for the higher order method of FR [2], [57]. This pairing allows for the artificial viscosity to be localised, deeming the method the name Localised Laplacian Artificial Viscosity (LLAV). In the context of DG method, both types of artificial viscosity have been widely used for compressive flows [23], including for hypersonics viscous flows [23], [58]. Shock detector methods especialy important to contain the viscosity to the desired area, and are comprised of several families. Hoskin, Van Heyningen, Nguyen, *et al.* [23] enumerates several of them, from simple ones based on differences in variables between neighbouring elements, to ones that include higher order derivatives. Artificial viscosity requires finetunning a design parameter (intuitively, the amount of viscosity) in order to acquire a good solution, though such can be automatically estimated, as done by Vandenhoeck [2]. As such, it might be considered that the artificial viscosity is a successful method used for higher order methods in the context of hypersonic flows.

Shock fitting tries to realign the mesh so that the shock is aligned with the mesh [23]. It has been used for chemical reacting flows flowing against a slope [59] and for high speed flows [60] by realigning an unstructured mesh. Huang and Zahr [60] focused on the robustness of the optmization problem central to fitting the mesh and presented nine experiments, including 3D simulations and supersonic and hypersonic inviscid flows, that showed good fitting. It is also used by several state-of-the art codes, such as LAURA, FUN3D, US3D and SPARC [23] . This method is not implemented in COOLFluiD.

However, moving meshes are present in COOLFluiD for the case of r-refining, which increases the mesh density in certain regions by moving the mesh elements [53]. However, this does not align the mesh elements to the shock.

Huang and Zahr [60] surveyed three methods for limiting methods, emphasising their use of explicit timesteping. Current focus on hypersonic simulations with COOLFluiD, however, follow an implicit timestep approach for non-time-accurate simulations, which converge to a steady solution. Regardless, they divided the limiter approaches into a classical one, a local WENO deconstruction, the Runge-Kutta Discontinuous Galerkin (RKDG) minmod limiter, and the hierarchical high-order Multi- dimensional Limiting Process (hMLP) limiter. The latter was developed in Park and Kim [61], with the motivation that this limiter was already successful for FV methods. It has been improved [62], as well mixed with artificial viscosity [63]. The RKDG has been used in nonequilibrium simulations [64].

In the context of COOLFluiD, Vandenhoeck [2] surveyed limiters, and concluded that hMLP limiter was the most promising one, but found in [65] that the this method makes "the residuals to stagnate far before convergence to machine precision is reached" [2]. As such, Vandenhoeck [2] preferred to adopt the LLAV method instead. Park, Yu, Kim, *et al.* [66] compared numerically the hMLP limiter and artificial viscosity, finding numerical errors associated with hMLP, such as lack of sub-cell resolution near a shock, and how it may fail to obtain a fully converge solution for a steady state simulation – which is exactly the type of simulation that has been focused on with COOLFluiD in hypersonics with FR. However, given that this comparison is from 2014, it does not take into account the aforementioned improvements. It also does not take into account its simultaneous use with other methods.

From a shock capturing perspective, it could be argued that the LLAV already developed and implemented in COOLFluiD already allows for good shock capturing in the context of flux reconstruction, and that these alternatives should be taken into account if the current method is later found to be an impediment or incomplete to ongoing research efforts.

2.3.4.2. Positivity of the Variables

Hypersonic flows, to properly capture the necessary physics, must include chemical reactions which take place at high temperatures. This implies the use of several chemical species, such as the 11 species model [8]. Furthermore, in reentry conditions for hypersonic vehicles, the air is at a low pressure [7]. Both the density and pressure are positive thermodynamic variables, and hence should be positive when the solution converges. Negative density and pressures are non-physical and can lead to instabilities during the simulation that can lead to divergence.

2.3.4.2.1 Limiters

The first class of methods analysed are the limiter based one, where the solution is limited in each element.

A Limiting Approach for DG methods

For DG methods, Zhang and Shu [67] developed a positivity preservation scheme for the Euler equations in 1D and 2D, based on limiting density and pressure. Density is limited directly for each element based on the average density value of each element. Pressure is limited based on the calculation of a pressure value from the average density, the momentum and the total energy of the element. Both of these limitations require a hardcoded value of a small positive number to which these values are compared to. It should be noted that the DG methods has several points inside it for which the conserved variables are

evaluated, which is the reason the average density for the element is required. These points are named solution points. Let x_i be the solution points inside the element. To limit density for each element, the following must be calculated for each element:

$$\epsilon_l = \min(10^{-3}, \bar{\rho}, p(\bar{w})) \tag{2.3.13}$$

$$\hat{\rho}(x_i) = \theta_l(\rho(x_i) - \bar{\rho}) + \bar{\rho}, \quad \text{with } \theta_l = \min\left(\frac{\bar{\rho} - \epsilon_l}{\bar{\rho} - \rho_{\min}}, 1\right), \tag{2.3.14}$$

where $\rho_{\min}$ is the smallest density a the solution points x_i inside the element, and $\bar{w}$ the average of the conserved variables, that is $\bar{w} = (\bar{\rho}, \bar{u\rho}, \bar{v\rho}, \bar{E})$, which u and v the velocities in x and y respectively, and E the total energy. Intuitively, we let $w = (\rho, u\rho, v\rho, E)$, and can be evaluated at each solution point. The calculation of pressure from the state conserved is done through ideal gas relation.

Pressure can be limited indirectly for each element. The procedure for each element is now presented. Let $\hat{\vartheta}_i = (\hat{\rho}, \bar{u\rho}, \bar{v\rho}, \bar{E})$ at x_i. If $p(\hat{\vartheta}_i) < \epsilon_l$, then we solve for $t_{\epsilon i}$:

$$\begin{cases} p\left[(1 - t_{\epsilon i})\bar{w} + t_{\epsilon i}\hat{\vartheta}_i\right] = \epsilon & \text{if } p(\hat{\vartheta}_i) < \epsilon_l \\ t_{\epsilon i} = 1 & \text{if } p(\hat{\vartheta}_i) \geq \epsilon_l \end{cases}. \tag{2.3.15}$$

A new limited value for the conserved variables, deemed $\tilde{w}$ is computed as:

$$\tilde{w}(x_i) = \theta_2(\hat{\vartheta}_i - \bar{w}) + \bar{w}, \tag{2.3.16}$$

with

$$\theta_2 = \min_i \frac{s_{\epsilon i} - \bar{w}}{\hat{\vartheta}_i - \bar{w}} \quad \text{over solution points } x_i, \tag{2.3.17}$$

$$s_{\epsilon i} = \begin{cases} s_{\alpha,i}(t_\epsilon) & \text{if } p(\hat{\vartheta}_i) < \epsilon_l \\ \hat{\vartheta}_i & \text{if } p(\hat{\vartheta}_i) \geq \epsilon_l \end{cases}, \tag{2.3.18}$$

$$s_{\alpha,i}(t) = (1 - t)\bar{w} + t_{\epsilon i}\hat{\vartheta}_i. \tag{2.3.19}$$

This scheme is based on physical arguments for inviscid flow, and is extended to 2D by Zhang and Shu [67]. This method, intuitively, first limits the density directly, by comparing the average with a very small positive value, and retrieving the density only if it is positive, but blending with the average if it is negative, enforcing positivity for all terms. The pressure is limited indirectly by calculating a blending factor $t_{\epsilon i}$ to blend between the average state with and without limited density. This then allows the calculation of a limiting factor θ_2 indirectly, which depends on this blending factor. The resulting states should be positive. If the conserved variables were chosen differently, this method would have to be adjusted, as it assumes that p is not a conserved variable and that ρ is.

This method held in Zhang and Shu and was extended in [68] to include reacting terms and gravity. The non-equilibrium situation was numerically simulated for a shock tube. The reacting terms were modelled with forward and back reactions for three species, where the reaction rate coefficients were modelled by an analytical expression dependent only on temperature. The densities and pressures obtained were positive and well converged. A Mach 80 with radiative cooling was also simulated, also having positive values. Peyvan, Shukla, Chan, *et al.* [22] comments, however, how Zhang and Shu [68] predicted the wrong high pressure ratio in the shock tube problem for the three species. As such, while

this method remains an interesting method for positiveness, it does not guarantee a correct solution.

COOLFluiD FR Positivity Method

Vandenhoeck [2] and Vandenhoeck and Lani [57] used a similar positivity preserving scheme together with LLAV in order to simulate steady state solutions in COOLFluiD with the FR method for the case of the Navier Stokes, including with turbulence and in the hypersonic regime. The clear advantage of looking for a steady state solution is that the positivity method does not need to be time accurate. Furthermore, the LLAV can interact with this positivity method in order to dampen the oscillations near the shock that are requiring the use of this method. As such, the positivity scheme does not necessarily have to be changing values (limiting variables) at the steady state solution, allowing the solver to converge to a correct steady state solution. However, possible failures of the positivity scheme might lead to negative values which might cause the solver to diverge if the LLAV does not dampen the oscilations. Despite the positivity scheme being used being different from the one by Zhang and Shu [67], it is similar, and the limiter by Zhang and Shu [67] was based on physical arguments for the Euler equations in 2D and does not necessarily work for other methods, such as with viscosity or reacting terms.

The method used by Vandenhoeck [2] and Vandenhoeck and Lani [57] limits the density in the similar way as Zhang and Shu [67], using equations 2.3.13 and 2.3.14, but with the $\rho_{\min}$ being the minimum density found at the flux points rather than at the solution points. Pressure is also limited indirectly, as they use $w = [\rho, \rho\mathbf{v}, \rho e_t]$, with $\mathbf{v}$ the velocity vector, and e_t the internal energy, for the 3D Navier Stoke equations. Following the notation of the previous presented method, pressure is limited only if $p_j < \epsilon$, in which p_j is the pressure at a flux point, in which case one solves for t_j

$$p\left(t_j(\hat{\vartheta}_i - \bar{w}) + \bar{w}\right) = \epsilon_l\,, \tag{2.3.20}$$

where now t_j is used as notation to emphasise the value being related to a flux point rather than a solution point, in the FR scheme. Using this result, one limits the state w, obtaining $\tilde{w}$:

$$\tilde{w}(x_i) = t_2(\hat{\vartheta}_i - \bar{w}) + \bar{w} \;\text{ with } t_2 = \min_j(t_j). \tag{2.3.21}$$

For clarity, $\hat{\vartheta}_i$ would be equal to $(\hat{\rho}, \bar{\rho\mathbf{v}}, \bar{\rho e_t})$ at x_i. One difference to take into account between this method and the one by Zhang and Shu [67] is the use of flux points for limiting the pressure, rather than the use of solution points.

Given this, it is expected that the method by Zhang and Shu [67] might not hold for reacting flows in the hypersonic regime and, as a consequence, so too the one by Vandenhoeck [2]. However, the latter employs additionally the LLAV which has allowed him to run in the hypersonic regime – including with turbulence – but without chemistry. As such, posing a positivity preserving method as this one with additional methods might lead to better results, or to correct its deficiencies. However, special care must be taken for the interactions between the additional numerical methods [2]. Furthermore, a different way of preserving positivity might be entirely be better, namely due to the fact that for non time accurate simulations such positivity does not always have to exist, as long as it does not cause divergence and allows the solver to converge to an accurate positive solution.

2.3.4.2.2 Entropy Stable and Entropy Conserving Fluxes

Another approach to getting positive results is by imposing an entropy conservative flux [22], possibly together with other methods. Peyvan, Shukla, Chan, *et al.* [22] developed an entropy conservative flux for the hypersonic case with real chemistry for cases with the DG method, and added an order blending method to account for oscillations near shocks. Damping oscillations near shocks by changing the order of the method is also present in p-refinement methods [69].

Entropy stable (ES) schemes gained traction in the last decade [70], as they provide additional stability to the method [70]. Entropy conservative (EC) schemes are schemes which preserve entropy in the integral form, allowing for the entropy being conserved for the entire domain [70]. Otherwise, the ES scheme can be such that it is just non-decreasing instead [70].

Slotnick, Khodadoust, Alonso, *et al.* [71] claimed that ES schemes could help provide large advancements in the area of "incomplete or inconsistent convergence behaviour" as robust new models for stability. They are clumped together as solution in this area with monotone and positive preserving schemes. While EC schemes are not necessarily positive preserving [70], they do allow for robustness of the solution, diminishing the impact of negative terms, or even diminishing large oscillations, thereby possibly preserving the positivity in certain occasions. Both of these would help the solver converge to a steady state positive solution, rather than diverging due to negative terms.

Entropy Stable Fluxes for Ideal Gas

The approach by Gouasmi, Duraisamy, and Murman [70] uses entropy fluxes to build a conservative scheme for the multicomponent Euler equations. Peyvan, Shukla, Chan, *et al.* [22] based themselves on this approach to derive the same method but applied to reacting hyperonic flows, where the internal energy is dependent on temperature, through the use of the Rigid-Rotor Harmonic-Oscillator model, which gives a different expression than the one used by by Gouasmi, Duraisamy, and Murman [70]. As such, it is evident that these methods depend on the formulation of the internal energy and its dependency on temperature, thereby requiring new derivations for different models.

The entropy fluxes derivation overview for 1D is given here. An entropy conservation equation of the form

$$\frac{\mathrm{d}(s_s)}{\mathrm{d}t} + \frac{\mathrm{d}(F_S)}{\mathrm{d}x} = 0 \tag{2.3.22}$$

with entropy for each species $_s$. One must, then, find the fluxes F_S. If ideal gas is assumed, the following expression is used:

$$s_s = \int \frac{\mathrm{d}e_s}{\mathrm{d}T}\mathrm{d}T - R_s \ln(\frac{1}{\rho_s}), \tag{2.3.23}$$

where e_s is the internal energy of species $_s$, R_s is the ideal gas constant for that species, the index $_s$ indicates that it relates to species $_s$, T is temperature and ρ is density.

Entropy s can be given by

$$s = \sum_{i=0}^{N_s} \rho_i s_i, \tag{2.3.24}$$

where N_s is the number of species, if needed. Entropy variables ω_s for each species are defined as:

$$\omega_s = \frac{\mathrm{d}(\rho_s s_s)}{\mathrm{d}U}, \tag{2.3.25}$$

with U being the conserved variables. Peyvan, Shukla, Chan, *et al.* [22] uses the Gibbs expression to acquire the ω_i. It is also possible to expand the derivative as $\frac{\mathrm{d}(\rho_s s_i s)}{\mathrm{d}U} = \frac{\mathrm{d}(\rho_s s_s)}{\mathrm{d}Z}(\frac{\mathrm{d}(U)}{\mathrm{d}Z})^{-1}$, with Z being the primitive variables. The entropy fluxes are then defined by the relation:

$$(\omega_s^+ - \omega_s^-) \cdot F_{S_s} = \left((\rho_s R_s u_x)^+ - (\rho_s R_s u_x)^-\right), \tag{2.3.26}$$

where + and − denote the left and right of a jump in the variables, and u_x denotes the velocity normal to the jump (x direction in the 1D case). This condition imposes that the fluxes preserve the entropy across jumps. By solving for F_S, one can then use theses fluxes in the entropy preservation equation 2.3.22

Assuming that the expression for the internal energy is available, derivation of the entropy fluxes might be possible. As such, this could improve the robustness of the solution. However, this also imposes that additional fluxes must be added to the physics, which requires changing or redoing the physics module of the CFD solver. In the case of COOLFluiD, this model would have to be integrated into a large code, which could lead to implementation problems or delays. The level of complexity of implementing of this method, then, might be higher than apparent due to the surrounding factors of the code which must integrate it. Furthermore, additional fluxes impose an additional computation cost on the method.

Entropy Stable Fluxes with Order Blending

Peyvan, Shukla, Chan, *et al.* [22] paired this method with an order blending method, which sought to lower the order of the method near shocks, hence allowing for fewer oscillations, complementing the robustness given by the entropy preservation. Using several methods together to increase robustness is in line with the approach of Vandenhoeck [2] in the context of COOLFLuiD when he used LLAV together with positivity preservation. However, as in that case, interactions between these methods should be taken into account as it might impede the correct use of either method, and require a reformulation.

Entropy Stable Fluxes with Reacting Terms

Ching, Johnson, and Kercher [72] developed a ES scheme for the 1D Euler equations with reacting terms discretised with a DG method which is entropy-bounded and positivity preserving. They tested this method for a thermal bubble, a shock tube and for a detonation wave. Only the latter entails non-equilibrium (with four chemical species), while the others seek to check and advection case and a discontinuous case, respectively. An explicit time stepping method with CFL of 0.1 is used, and thus this method is not necessarily appropriated for the case of implicit time steps for larger CFLs. The case for low pressure was not studied. Despite this, it showed only positive terms in the simulations.

Entropy Stable Fluxes from Limiting

Ching, Johnson, and Kercher [72] built on a body of work of proofs for entropy preservation and boundedness in order to develop a limiter procedure that could be shown to impose a minimum entropy bound. This limiter procedure implies limiting the density, the positivity of the concentration of the species, the shifted internal energy and only at the end the entropy. The entropy is limited indirectly by defining a limiter for the conserved variables based on entropy. The shifted internal energy is defined in the paper, and serves as a concave function required in the limiting procedure. A concave function based on the entropy and on the minimum bound is defined such that it can be used in the limiter for the entropy. The limiters are all based on blending the state with its average, using a blending variable, which is the focus of the definition of the limiter procedure. For clarity, the last step of the limiter is

given here:

$$w^{(4)} = \bar{w} + \frac{\chi(\bar{w})}{\chi(\bar{w}) - \min\left(\chi\left(w^{(3)}\right)\right)}(w^{(3)} - \bar{w}), \tag{2.3.27}$$

where w is the conserved variables, $\bar{w}$ the average conserved variables, $w^{(4)}$ the variable limited in the 4th limiter step (that is, after limiting density, concentration of the species, and the shifted energy). Analogously, $w^{(3)}$ is the variable limited in the 3$^{\text{rd}}$ limiter step. The function $\chi(w) = \rho s - \rho s_{\text{minimum value}}$, such that the entropy is bounded to $s_{\text{minimum value}}$. Several ways of calculation of this bounded value is also given by Ching, Johnson, and Kercher [72], all based with comparing the entropy in all elements of the domain. A version with a relaxation constant is also given. A user defined boundary is also possible, since such is just calculating the boundary by defining a constant.

The above method is used by Ching, Johnson, and Kercher [72] with a artificial viscosity as well as with modified fluxes, in order to improve the robustness of the solution. They also note that the correct Rienman Solver should be chosen, namely Godunov, Lax-Friedrichs, HLL, or the HLLC.

An important downside is important to mention: this method assumes an explicit time marching scheme with a strict CFL, which is also derived in the paper. In particular, the theoretical formulation focuses on first establishing a set of variables with the condition that they are positive, and then analysing the restrictions and methods which make sure that the these variables will still belong to this set in the next time step; that is, that they will remain positive.

The blast wave simulation includes a simulation without artificial viscosity to show that the entropy limiter allows for smaller oscillations. This method differs from the conservative entropy methods by just bounding it rather than trying to keep it conserved with entropy fluxes. Furthermore, it includes a vast mathematical foundation that could be exploited for certain applications. However, it is highly based on the minimum entropy principle for its proofs, as derived by Gouasmi, Duraisamy, and Murman [70]. As such, if this principle is not true, the limiter does not necessarily work in general.

2.3.4.2.3 Entropy Based Filtering

An alternative method also based on entropy is entropy based filtering [73]. Entropy based filtering for DG methods has been applied to the Navier-Stokes [73] and to the MHD equations [74]. The conserved variables u are constructed for each element in the case of nodal DG and FR methods as:

$$u = \sum_{i=0}^{N} u_i l_i, \tag{2.3.28}$$

in which l_i is the associated function to each node, which can be a basis function of the Lagrange polynomials. Then, this formulation can be changed to a modal formulation with:

$$u = \sum_{i=0}^{P} \hat{u}_i \psi_i, \tag{2.3.29}$$

with ψ_i the basis functions and $\hat{u}_i$ the coefficient for each basis function. P is the order of the polynomial, and the basis functions are assumed to be polynomial basis functions, such as the Legendre polynomials.

As such, a filtered version can be constructed as:

$$\tilde{u} = \sum_{i=0}^{P} \mathscr{H}_i(\hat{u}_i)\psi_i, \tag{2.3.30}$$

with $\mathscr{H}_i$ being the filter. Dzanic and Witherden [73] derived a filter in the following form:

$$\mathscr{H}_i(\hat{u}_i) = e^{-\xi_{\text{filter}} p_i^2} \hat{u}_i, \tag{2.3.31}$$

where p_i is the order of the mode (in case of polynomials, it is the order of the nomial), and ξ a parameter computed based on entropy. This filter has the properties of being dissipative:

$$\mathscr{H}_i(\hat{u}_i) \leq \hat{u}_i, \tag{2.3.32}$$

as well as the property of being conservative

$$\frac{1}{V_{\text{element}}} \int_{\Omega} \tilde{u} \, \mathrm{d}V = \bar{u}, \tag{2.3.33}$$

with V denoting the volume, V_{element} being the volume of the element Ω, and $\bar{u}$ being the average of u, element wise.

Intuitively, then, the different modes can be filtered as to lower their contribution. The value ξ_{filter} must still be determined. For such, Dzanic and Witherden [73] formulate the following optimization problem which, when solved, returns ξ_{filter}:

$$\xi_{\text{filter}} = \arg\min_{\xi \geq 0} \text{ s.t } \{\Gamma(\tilde{u}(\xi)) \geq 0, \sigma(\tilde{u}(\xi))) \geq 0, \forall_i \in S\}, \tag{2.3.34}$$

where Γ and σ are functionals and $S = 0, 1, 2, ..., P$. This way, the minimum value greater or equal to zero that obeys those restriction is searched for. Dzanic and Witherden [73] claims that a local minimum suffices. One should note that the case for $\xi_{\text{filter}} = 0$ returns an unfiltered version and, as such, those functionals are restrictions that impose the need to filter. The role of entropy is found in these restrictions. Γ is a functional such that positivity constraint for primitive variables, such as pressure or density. Indeed, Dzanic and Witherden [73] define for the case of the Navier-Stokes:

$$\Gamma_0(u) = p - p_{\min}, \quad \Gamma_1(u) = \rho - \rho_{\min}, \tag{2.3.35}$$

with p being pressure and ρ density, and the $_{\min}$ meaning the that it is a minimum value imposed, usually very small and positive. The functional σ is a positive concave function that takes the role of entropy. In the case of a physical system such as the one described by the Navier-Stokes with idea gas, physical entropy can be used. Indeed, Dzanic and Witherden [73] define:

$$\sigma(u) = \rho \log(p\rho^{-\gamma}) - \sigma_{\min} + \epsilon_\sigma, \tag{2.3.36}$$

with γ being the heat capacity ratio. The value $\sigma_{\min}$ corresponds to the minimum σ computed in the element in the previous iteration of the method. The value ϵ_σ corresponds to a user imposed tolerance that Dzanic and Witherden [73] claim to help with numerical errors associated with machine precision.

The above formulation allows the use of a filter that imposes positivity of density and pressure. These

values can become negative due to numerical effects in hypersonic simulation cases with higher order methods, and as such this method can be of value. Furthermore, it is rooted also in only increasing entropy of the system, which could help the robustness of the method, in the same line with ES and EP schemes. It is not clear whether the optimization scheme will have a large impact on the simulation in terms of both computing time and in terms of diverging or converging too slowly. Furthermore, the exact method for such optimization is not necessarily clear. Dzanic and Witherden [73] used 20 iterations of a bisection approach to calculate ξ_{filter} if the constraints were not met for the element, while also advising that bracketing methods such as Brent or Illinois methods could possibly be used. The fact that the filter is only applied at all when the constraints are not met does allow for better computation time. This filter could possibly be used with other ways of detecting troubled elements besides checking the constraints, and could also benefit from using more formulations than just the minimum ξ_{filter} that satisfies the constraints. As seen in other methods, modal energy based evaluations can be used to detect troubled elements [75], and smoothness detectors – such as the one used for shock detection by Vandenhoeck [2] – could also be used to affect ξ, as Vandenhoeck [2] used this smoothness detector to change the coefficient of his implementation of LLAV.

2.3.4.2.4 Flux Order-Blending

Returning to the work of Peyvan, Shukla, Chan, *et al.* [22], the use of an order blend method can help with oscillations by lowering the order of the method near the shock. This work is a direct use of the formulation developed by Hennemann, Rueda-Ramírez, Hindenlang, *et al.* [75]. This formulation was developed for DG with flux splitting. The idea focuses on taking the flux for the interior nodes in the element (excluding the left most and the right most node, in 1D), and blending two flux formulations: one high order and on low order. In the context of DG, these nodes appear in the volume terms in the equation, and not on the boundary terms. Thus flux blending can be taken as:

$$f_{\text{blended}} = \alpha f_{\text{order 0}} + (1 - \alpha) f_{\text{High Order}}, \tag{2.3.37}$$

where the low order blending flux method can be taken as finite volume formulation as in [75].

The term α is computed directly based on the element. In the work of Hennemann, Rueda-Ramírez, Hindenlang, *et al.* [75], α is computed based on the energy of the two highest energy modes of the element. In the context of nodal methods – such as in the FR formulation – such requires a change of basis for this computation. This energy is compared to a threshold in order in order to calculate an α between 0 and 1. This α is then clipped based on minimum and maximum values such that it becomes 0 bellow a certain value, and 1 above another one, for computational efficiency.

This method can, thus, be synthesised as follows. Define a maximum energy E_{max}:

$$E_{\text{max}} = \max\left(\frac{m_P^2}{\sum_{i=0}^{P} m_i^2}, \frac{m_{P-1}^2}{\sum_{i=0}^{P} m_i^2}\right), \tag{2.3.38}$$

with m_i being the coefficient of the term of order i of the polynomial of order P that represents the variable ρp. This variable choice was made to detected discontinuities in p and in ρ by Hennemann, Rueda-Ramírez, Hindenlang, *et al.* [75], but could be changed to another if such is deemed necessary. A threshold $T_{\text{threshold}}$ is defined as:

$$T_{\text{threshold}} = a_{\text{threshold}} \cdot 10^{-c_{\text{threshold}}(N+1)^{\frac{1}{4}}}, \tag{2.3.39}$$

where Hennemann, Rueda-Ramírez, Hindenlang, *et al.* [75] determined $a_{\text{threshold}} = 0.5$ and $c_{\text{threshold}} = 1.8$. They then determined α to be:

$$\alpha = \begin{cases} 0 & \text{if } \alpha < \alpha_{\min} \\ \frac{1}{1+\exp\left(\frac{-9.21024}{T_{\text{threshold}}}(E_{\max}-T_{\text{threshold}})\right)} & \text{if } \alpha_{\min} \leq \alpha \leq 1-\alpha_{\min} \\ 1 & \text{if } \alpha > 1-\alpha_{\min} \end{cases} . \tag{2.3.40}$$

2.3.4.2.5 Mesh re-refining

An alternative way to keep the values positive might be to better resolve the region of high oscillations. Basile, Chapelier, de la Llave Plata, *et al.* [76] presented h- and hp-refinement strategies for DG methods based on a-posteriori error estimations. H-refinement is based on refining the mesh when needed, while p-refinement changes the order of polynomial used in the DG method in order to increase the order. A-posteriori error estimations are presented, namely based on evaluating the jump in variables and in the difference in energy of the highest modes. Basile, Chapelier, de la Llave Plata, *et al.* [76] advices the use of both error estimation methods simultaneously, and proposes a way of integrating both results. A smoothness indicator was used to indicate if the error was due to a discontinuity in the flow or not, hence allowing which refinement to use in the case of hp-refinement. For near discontinuities, h-refinement is usually used, while p-refinement is usually used for smooth regions. This is in line with the previous methods which tried to lower the flux order near the discontinuity – such as the flux limiters – given that the oscillations are linked to the high-order of the method. Hence, refining the grid might better resolve near the shock, while increasing the order might worsen the solution there.

Their implementation requires a re-mesher to refine the mesh in the correct place. This requires the integration or the implementation of a method for refinement, which might not be trivial. Given that h-refinement would the more appropriate to keep positive values near shocks, a re-mesher – or an equivalent method – would be crucial. However, this method creates additional elements, which should be avoided if possible in order not to increase computational time. As such, if other methods which do not require refinement were to properly resolve the elements near the shock as to converge to an accurate solution, it would be preferential to avoid this method.

However, if needed, by re-meshing automatically, it would lower the need of trying to capture the shock well enough by hand to avoid discontinuities. An alternative to this method, however, could be r-refinement, as presented by Ameur [53] already in the context of COOLFluiD, which moves/distorts the elements in order to increase the mesh density where needed at the expense of lowering it somewhere else, while keeping the total number of elements the same. In the same line as before, due to the complexity of this one, as well as not being necessarily focused in diminishing numerical oscillations due to the use of higher order methods, a simpler method should be preferential, if one is available.

2.3.4.2.6 Positivity Preserving Time Discretisation

While the previous methods – focused on positivity preserving methods and ways to control the numerical oscillations typical of higher order methods – were related to how the domain and fluxes are discretised, there are also a category of methods related to time discretisation [77]. Blanes, Sergio, Iserles, Arieh, and Macnamara, Shev [77] presented a number of explicit time marching schemes that preserve positivity, namely the Patankar methods, which are based on the Runge Kutta schemes. Ciallella, Micalizzi, Öffner, *et al.* [78] applied a Patankar method to the shallow water equations. They also

claimed that the implicit Euler method is unconditionally positive, contrary to the explicit version. For the case of steady state simulations with implicit time marching methods, negative values for the variables should not arise due to the time discretisation – only due to the space discretisation – and hence this should not be the focus. However, if time accurate simulations with explicit methods were to be used, these methods could be advantageous at keeping the variables positive.

2.4. Research Gap

In the introduction section, the need for accurate CFD simulations in the realm of hypersonics was clearly identified and motivated. The software COOLFluiD was also briefly discussed, from its past uses to the ongoing research with it. In section 2.2, the physics which must be captured were identified, with special interest in nonequilibrium conditions, and the available models for chemistry and multiple temperatures. Higher Order methods – with particular emphasis on DG and FR methdos – were discussed and motivated in section 2.3. The interest in their use, as well as the on going efforts to make use of FR methods in COOLFluiD, were also discussed. Current problems with higher-order methods were discussed, namely in the context of hypersonics. From this analysis, it is was clear that the extension of FR methods to hypersonics with non local thermodynamical equilibrium has not been made and it could be of great interest. The validation of the physics model with non local thermodynamical equilibrium was mentioned to have been done for FV in COOLFluiD already, bringing an opening to now extend it to FR methods. The treatment of FR methods with source terms has, as aforementioned, been already treated in COOLFluiD. As such, using FR methods in COOLFluiD with these models and nonequilibrium conditions seems to be the next step, and is the current identified research gap.

From DG methods, it was identified through consulting literature that shock capturing and the positivity of the variables were difficult problems due to the strong nature of the shocks, as well as their important roll in the physics. These problems also exist with FR methods, which lead to the use of a LLAV method and a positivity limiter in [2]. However, the simple approach of the limiter does not necessary work for multiple species with chemistry and multiple temperature models. This can be easily seen by the fact that the limiter used by Vandenhoeck [2] makes use of a small threshold, and species mass fractions are allowed to go as low as zero and still be physical.

As such, inside the aforementioned research gap, there exists a gap in how to preserve the positivity of the variables, or increase the robustness to the negativity of the variables of the solver, as to allow the convergence to steady state solutions in a reasonable amount of time.

Hence, the problem can be identified as:

Thermodynamical nonequilibrium flows in the hypersonic regime with finite chemistry are required for accurate simulations of the hypersonic regime. The FR method included in COOLFluiD currently does not support these flows.

This problem has the sub problem:

FR method requires either a better positivity preservation method or better robustness to unphysical negative values.

This seems to imply the question:

How to keep positivity of the variables or increase the robustness to unphysical negative variables for the FR method?

While the above is two questions in one, both options are available, and are highly related. By finding a way to keep the variables positive without endangering the convergence of physicality of the simulation, it would also make it more robust to negative values, as likely the positivity method is either overriding negative values or preventing the simulation of simulating them in particular steps. Given that a steady state solution must converge to positive values in order to be physical, it is obvious that increasing robustness as to allow this would imply that the values are all positive at the end of the simulation.

In section 2.3.4, not only problems were tackled but also solutions to them in literature were found and discussed. These solutions give insights into which methods are currently in use to fix these problems, and how they can be adapted and integrated in order to attack the above stated problems. Discussing these strategies is the focus of section 2.5.

2.5. Positivity Preservation Strategies

Given the problem statements and research question in the previous section, a comparison between the positivity preserving strategies is here presented. An overall view is present in table 2.1.

Table 2.1: Comparison between methods for positivity preservation.

Method	Solves Issue	Implementation Advantages	Implementation Issues
Limiter	Due to the requiring a small positive coefficient it might not be advisable for simulations with very small mass fractions and strong oscillations.	Previously done. Done after the FR method	Could interact with other methods. E.g. it has been seen stalling convergence when used with a LLAV if the parameters are not appropriate.
Order-Blending	Could solve the problem near the shock	Additional flux can be computed independently. Favours modular approach, with separate calculation of fluxes.	Must be implemented in FR method. Implies detecting elements in FR method loop. Implies adjusting the α calculation for us.
EC Method	Could increase robustness enough to allow convergence to a physical solution, but it is usually paired with other methods such as order-blending	There are already physical modules in place, and adding additional equations of the same time, despite not being trivial should be easier than making it from the ground up	Requires a new physical module. Requires derivation of the equations for our case. Likely requires another method, such as positive preservation method or a way to treat the oscillations near shocks – though maybe the LLAV might already help with that. The use of additional equations in the physical module which are coupled to the already in use ones might imply unforeseen consequences in convergence and required additional treatments.
Entropy-Based Filtering	Could help decrease the oscillations as to allow convergence. Could interact correctly with the LLAV method in a similar way to the currently used positivity enforcing limiter.	Could be done as an independent step. Does not imply negative values. Is conservative and always decreasing.	Requires a change of basis. The parameters might not be appropriate. Requires identification of the troubled elements.
H-refining	Could indeed decrease oscillations near the shock.	Does not require touching neither the physical modules, nor the FR modules.	Requires an external re-mesher. For parallel treatment of meshes, remeshing could be hard to implement. Will increase computation time and the number of elements, which goes against the advantages of using a coarse grid with FR.
Patankar method	No. It could be used to improve time accurate solutions with explicit methods where forward Euler methods might not be appropriate.	Beyond the scope of this review	Beyond the scope of this review

From table 2.1, which is based on already provided information in section 2.3.4, it can be seen that limiter, order-blending and entropy based filtering approaches are the easiest to implement ones. From these two, limiters have the weakness of requiring a threshold minimum value which is a user defined

small positive value. Given that small mass fractions are to be expected and physical, this method would not only requiring tuning, but it might prevent good convergence to the physical solution. Order-blending tackles instead the fluxes in order to decrease the order of the flux near the shock. As such, the oscillations near the shock are smaller, which might allow for better convergence and more robustness.

Order-blending also has good properties for implementations, allowing to impose the method inside the loop where the fluxes are computed, and then couple them together. As such, order-blending seems to be the most promising method to increase robustness easily, but would require an additional positivity preservation method.

EC methods have a longer more difficult implementation, as seen in table 2.1. However, they are the ones that could bring more robustness, namely together with order-blending. As such, if order-blending is found to be promising, but not enough, this method could – together with order-blending – improve the simulation.

Entropy filtering could possibly be used instead of order-blending, as it promises better robustness, and it has an implementation which could be seen to be of lower difficulty than order-blending. Furthermore, it alters the states instead of the fluxes, making it more apt to be used with an optimisation algorithm which could force the solution to be positive, for example. It is also more predictable in the sense that alters directly the states when trying to force positivity of those.

Patankar methods should not be taken into account until time accurate simulations are tried with explicit method, and only in the case that more well established methods, such as forward Euler, do not work.

H-refining requires additional remeshing software to be integrated and would imply an increase in computation time due to increasing the number of elements, as well as the necessity of changing the number of elements dynamically in the code. This change would also have to work with the computations in parallel for the grid. All of the former implies that the other mentioned methods would be preferable form an implementation standpoint.

However, as seen from the methods presented, no singular method allows definitely for positivity. As such, it seems to suggest that a novel approach, possibly combining with these methods should be developed.

Based on the above analysis, a novel approach is found to be worth pursuing. Furthermore, the entropy based filtering could also help guarantee positivity by acting directly on the states, and possibly behaving better than the limiting approach. Its implementation code wise is feasible. Thus, these two methods can be chosen as initial targets of research for enabling positivity in COOLFluiD for TCNEQ simulations. Order blending is found a possible future endeavour, of higher complexity. Robustness wise, EC methods are found to be complex approaches which could be used in the future.

2.6. Supersonic Thermal and Chemical Nonequilibrium Cases

Hypersonic reentry entails not only a multitude of phenomena to capture, but also a wide range of values for the thermodynamic and chemical variables [7]. Furthermore, the shock wave distance from the body is also of importance [7]. Identifying cases which entail reacting and thermal nonequilibrium phenomena is a must for the successful design and testing of any numerical model for their simulation, as well as for diagnosing unexpected behaviours during its application. This section focuses on briefly

stating these cases with contextual information regarding their usefulness.

2.6.1. High Enthalpy Shock Cylinder

The High Enthalpy Shock Tunnel Göttingen (HEG) at the German Aerospace Center (DLR) allows for the testing of reentry flows with relaxation phenomena [79]. This tunnel has been used for testing a cylinder under hypersonic stream conditions. Measurements of pressure and heat transfer were available through the use of thermocouples and pressure traducers on the cylinder. Shock wave position and density were measured through optical techniques.

In the work of Karl, Schramm, and Hannemann [79], it is argued that this case can be used for CFD validation. Knight, Longo, Drikakis, *et al.* [27] employed several numerical methods to simulate the experimental case, using the same boundary conditions and species model (5 species) between the models. Good agreement with the experiment was found, based on the pressure and heat transfer on the surface of the cylinder, and flow structures. More recently, Sharma, Giangaspero, Munafò, *et al.* [6] used this case to validate a FV volume formulation in COOLFluiD with the species model given by PLATO. Good agreement was found. Both of these studies provide relevant free stream conditions in terms of thermodynamic variables, velocity and species concentrations, as well as flow field variables throughout the simulations. As such these studies give informations not only about the typical behaviour and quantities in blunt body hypersonics, but are also possible validation cases that can be used to test numerical methods.

In the work of Knight, Longo, Drikakis, *et al.* [27], the shock wave standoff distance is not clearly stated. In the work of Sharma, Giangaspero, Munafò, *et al.* [6], for a cylinder of 45 mm radius, the shock wave stands at circa 10 mm from the body, based on the available flow field plots. In the cases which relate to the cylinders studied in both works, a difference between the translational temperature and the roto-vibrational temperature is found in the order of the thousands of Kelvins, emphasising the need for a dual temperature mode. This is seen in the shock layer. A pressure jump up to 10 times the freestream value is found [6]. Heat transfer was validated in the work of Sharma, Giangaspero, Munafò, *et al.* [6] and of Knight, Longo, Drikakis, *et al.* [27]; pressure on the surface was also validated in the work of Knight, Longo, Drikakis, *et al.* [27].

These two works show the need for multiple species and temperature models, as well as give a case for validation. Furthermore, they contextualise the expected values for the temperature, pressure and shock wave position. The values for the species is only explicitly given for N in the work of Knight, Longo, Drikakis, *et al.* [27].

This test case was, then, both studied experimentally and through simulations. Quantities of interest such as wall measurements of heat flux and pressure and the shock wave stand off distance are known. Thus, this case presents itself not only a case of interest for verification of the method, but also for validation.

A deeper analysis and comment on the physics and numerical and experimental setups can be found in the respective papers [6], [27], [79].

2.6.2. The Atmospheric Reentry Demonstrator Capsule

The Atmospheric Reentry Demonstrator (ARD) capsule was a guided sub-orbital reentry vehicle which sought to showcase reentry vehicle development capabilities as well as gather flight data [6], [80]. The

case has been studied numerically by Sharma, Giangaspero, Munafò, *et al.* [6] using COOLFluiD with PLATO, where it was compared to the numerical study of Yu, Qiu, and Takahashi [81].

The free stream conditions for this case differ from the case of HEG cylinder in that the molecular oxygen has a large mass fraction for the ARD case, and the atomic oxygen dos not exist in the free stream. In contrast, the HEG cylinder has more than 0.2 of mass fraction of atomic oxygen and less than 0.1 of molecular one. The free stream temperature is also more than 3 times lower in the ARD case than in the HEG cylinder one. This is related to the ARD case being tested at 65 km altitude, while the HEG cylinder being a laboratory test with these conditions chosen. Regardless, this case gives a new realistic case useful for validation purposes, as well as for seeing flow and nonequilibrium effects.

In the studies by Yu, Qiu, and Takahashi [81] and by Sharma, Giangaspero, Munafò, *et al.* [6], nonequilibrium effects are present in the temperatures, where the translational temperature along the stagnation line peaks at 15000 k, while the roto-vibrational one peaks at circa 9500 K, keeping still a difference along that line, despite getting closer. Sharma, Giangaspero, Munafò, *et al.* [6] shows a large degree of ionisation through the electron species plot, in accordance with the work of Yu, Qiu, and Takahashi [81], where different degress of high ionisation are also found at several conditions. The electron number density increases by over 2 orders of magnitude after the shock wave in the study of Sharma, Giangaspero, Munafò, *et al.* [6], enveloping the capsule.

The validation done by Sharma, Giangaspero, Munafò, *et al.* [6] for this case reaffirms the use of PLATO with COOLFluiD to study complex hypersonic physics, while also showing typical order of magnitude present for the variables and their evolution in the flow field, and important physics behind the shock wave, such as ionisation.

The work by Yu, Qiu, and Takahashi [81] further emphasises the effect of dissociation of N_2 into N in the flow field and electron generation.

This case, then, presents itself a good case for validation and reference of the expected physics and values. The exact setup of the numerical studies and the indepth analysis of the physics can be found in the respective papers [6], [81].

2.6.3. The Qarman case

The Qarman QubeSat was part of a study by the Von Karman institute and was already object of study in the context of COOLFluiD [2]. In the work of Vandenhoeck [2], a five species air model was used for two temperature thermochemical nonequilibrium simulation. A flux reconstruction method was used in COOLFluiD. However, PLATO was not used, as the work predates the testing of the coupling of PLATO with COOLFluiD by Sharma, Giangaspero, Munafò, *et al.* [6].

This free stream conditions for this case concern a case in the hypersonic regime (Mach 8.46) with low pressure and density (39.53 Pa and $1.970 \cdot 10^{-4}$ kg/m^3, respectively), with multiple temperatures, making it suitable for further studies in positivity for hypersonic thermochemical nonequilibrium cases. This case was simulated by Vandenhoeck [2] in higher order (P2), further strengthening its interest in higher order simulations.

A well defined stand off distance is presented in [2], as well as plots for pressure and temperature T, giving further information for verification of the case. The results regarding heat flux were given for the wall and were compared to the FV formulation, also studied with COOLFluiD, finding a good

agreement at the stagnation point and an over estimation at the shoulder of the QubeSat for the P2 FR simulation. The case shows, then, a work in progress in COOLFluiD for FR, as well as a means of comparison: with well converged FV simulations.

2.6.4. The Kranc case

The report by Kranc, Yuen, and Cumbel [82] concerns a experimental study of influence of the use of an electromagnet in on the drag of a body (axisymmetric probe) under a free stream of argon plasma. The work focuses on studying the forces on the body – total drag, and pressure and viscous forces – depending on the force of the magnetic field, thereby allowing for the derivation of empirical relations. The measurements were done through optical measurements as well as through direct measurement of the force applied on the probe. Pressure and viscous forces were derived from measurements, but not measured directly.

The free stream of argon plasma consists of argon, ionised argon and electrons. The nature of this flow can be, then, reactive. This study, then, considers a reacting case which is different from the ones with air which have been presented so far.

This free stream flow is at circa 3000 m s^{-1}, with a temperature of 1100 K [82] . The Mach number reported is of 4.75 [82] , making it supersonic, as it evident from the shock wave seen during the experiments [82].

Both the chemical composition described as well as the free stream conditions make this case studied one of relevance for chemical nonequilibrium in supersonic conditions. Furthermore, it concerns, as previously stated, a different case in terms of species, making it of special interest for any study which seeks to test models against a wider spectrum of cases. It should be noted that PLATO contains the modelling for species involved in this study – electrons, argon and ionised argon – making it of further interest for simulations with COOLFluiD coupled with PLATO.

The empirical relations regard mainly the variation in values of interest – such as drag and shock wave stand off distance – based on the magnetic field. Not all values for the case without the magnetic field are readily accessible in this report. Some values are reported in cited references.

While this report does not give direct access to validation, it makes itself a good case for verification as well as for testing the extension of any model and methods to other species sets rather than those for air. The shock wave stand off distance, which is dependent on the chemical reactions behind the shock [7], may serve for verification. In the context of CFD, mesh refinement studies or increasing the order of the method for a fixed mesh may serve to show that the case is fully converged in the absence of better validation quantities, such as wall quantities.

2.7. Validation

Validation of the numerical method is required in order to guarantee that its usability in solving the complex flow phenomena for which it is designed, as well as see the effects of the newly developed solutions, which could impact the validity of the method.

For validation, numerical test cases of reference must be identified. The cases used for validation in the work of Sharma, Giangaspero, Munafò, *et al.* [6] could possibly suffice. They comprise three blunt body reentry vehicles and one test model (a cylinder) which were numerically simulated using a finite volume

framework with non thermochemical equilibrium using PLATO with eleven species. All of these relate to the goals of the current research, which is focuses on enabling using FR for non thermochemical equilibrium in hypersonics. Furthermore, they make use of the same species model, enabling a better comparison in terms of species densities. These cases have also been successfully validated, granting a certain degree of trust to their use in validation.

As such, validation could be based on the HEG cylinder, the Orbital Re-entry Experiments capsule, the RAM-C II (from the Radio Attenuation Measurement (RAM) experiment), and the ARD capsule. The HEG cylinder and the ARD capsule have been briefly discussed in section 2.6. Further details regarding the cases can be found in the work of Sharma, Giangaspero, Munafò, *et al.* [6]. Given the validation potential of the HEG case, as well as its extensive study in Knight, Longo, Drikakis, *et al.* [27], this case should be prioritised in its use for validation. Furthermore, the simple geometry of the cylinder makes it easier to be simulated while negating possible unimportant complications due to additional non interesting physics motivated by the geometry, or due to additional computational efforts. The Qarman case, due to its also simple geometry and previous studies in COOLFluiD, could possibly further be used for testing the developed methods for when air is used as a fluid.

In addition to these cases, the presented Kranc case in section 2.6.4 can be used for verification of the method, while also testing its modularity for other species sets. It should be selected together with the HEG case in order to further complete the study of any positivity methods developed.

2.8. Literature Review Conclusion

Thermochemical nonequilibrium flows were identified as important flows during reentry, impacting the vehicle through high heating and loads. The heat transfer during this process was seen as an object of extensive study over several decades and still an active research field. Given the specific conditions of reentry, direct measurements or on ground testing are hard to execute. As such, CFD simulations were seen here as identified as a viable and easier way to estimate the loads and heat transfer to the vehicle.

However, CFD simulations require careful modelling of the physics involved as well as careful setups. As such, the important physics were presented, justifying the thermochemical nonequilibrium nature of the flow, and clarifying the interplay between chemical and flow properties. Emphasis was put on the importance of the shock wave, and its effect, as well as in the order of magnitude the characteristic time of the flow and chemical reactions. Air species chemical models were discussed, emphasising Parker's 11 species model.

Regarding the discretisation, the flux reconstruction method was presented and described as a higher order method currently under development and already tested for hypersonic equilibrium flows with success. An interest in extending the use of this method for thermochemical nonequilibrium flows was, then, identified. The current code base – COOLFluiD – was briefly introduced, and seen as already including the required FR framework as well as a validated coupling to a deemed appropriate chemical model for thermochemical nonequilibrium – PLATO – which gives a good foundation for extending the FR to thermochemical nonequilibrium.

Given the intent to do this extension and the physics involved, problems regarding the simulation of these flows with FR were identified. Among them, the positivity problem was found to be an active research problem, with several works related to it. However, it was seen that none of the solutions were applied to such complicated systems as one as PLATO, in FR, with 11 chemical species. Furthermore,

it was seen that several of these works were done for Galerkin methods with explicit time stepping, the latter condition being fundamental to several of the approaches. While robustness methods were identified which could auxiliate positiveness, these were not seen to be shown to guarantee positiveness. The positivity methods were compared to the one currently in use for FR in COOLFluiD – which does not work for the thermochemical nonequilibrium with PLATO – and were found to be remarkably similar.

A clear research gap was identified: the need for positivity for this type of flows for FR. A research question was, thus, presented. From this discussion, strategies were identified from literature to ensure positiveness, while asserting that no strategy currently in literature could definitely ensure positiveness. Validation methods and contextualization of values which were to be expected in the simulations were discussed.

This literature review enabled a research direction with the required contextualization. The research itself will be presented and discussed in the following chapters of this thesis.

3

Filtering for Positivity

This chapter concerns the use of filtering as a method for positivity for flux reconstruction (FR) methods. In particular, it tackles the application of a filtering method as a replacement of the previous in use positivity method in COOLFluiD. For this initial case, an established test case with the Euler model (inviscid case) is used as a means of comparison and of verification that the filtering can be used as a viable alternative to the previous positivity method.

This step is an important precursor to the case with thermochemical nonequilibrium. It establishes both confidence in the method as a viable alternative, as well as a path to integration of a more complicated method with the current code.

3.1. Introduction

As emphasised in the literature review, the positivity of the variables poses an issue for higher order methods and is a considerable barrier for thermochemical nonequilibrium simulations with COOLFluiD – currently not allowing a single iteration. As such, it is the target of research.

Of the methods presented in the literature review, the entropy-based filtering is the most direct substitute for the previous positivity method, in the sense that it too targets the states at the solution points at the beginning of the spacial discretisation and enforces positivity on them based on their values. Furthermore, assuming that the mean mode of the variable being filtered is positive, it will guarantee positivity – the same as the previous method [73]. The problem with this assumption is dealt with in chapter 4, where a new method is developed assuming weaker assumptions.

In addition to this positivity enabling, filtering allows to lower higher order oscillations, increasing robustness. Furthermore, the use of entropy considerations helps in preventing entropy violations due to oscillations in the solution, which is expected to have a positive impact in terms of robustness [73]. The use of entropy, however, assumed a CFL smaller than 1, as waves are expected not to travel more than two elements in the derivation presented by Dzanic and Witherden [73]. Despite this constraint, filtering can be done without the entropy consideration as well while maintaining its increased robustness relative to the previous positivity method due to the damping of the higher order oscillations. As such,

this method could be an improvement on the previous one in terms of robustness while maintaining the same base assumptions.

Expanding on the filtering idea, while the filter used by Dzanic and Witherden [73] is an exponential filter, any filter that converges the states at the solution points to the mean mode could work in keeping positivity. However, intermediate solutions, which are desired over fully filtered ones, could have negative or positive impact on the simulation depending on the choice. It cannot be *a priori* ruled out in general that every filter will lead to a more robust solution or not lead to a blow up. Given that the exponential filter was seen to give results in the work of Dzanic and Witherden [73], it is currently preferable as an object of study.

This choice was supplemented with a convolution filter with the Fejér kernel – denominated here as the Fejér filter. This filter is presented in section 3.4. It constitutes a way to apply the Cesàro summation in directly to a function within an interval without transforming it to the Fourier domain. As a consequence, it targets directly the Gibbs phenomena [83].

3.2. Particular Research Questions

Referring back to research gap (see section 2.4), a method to impose positivity is sought after. Given that filters act directly on the states and can, hence, directly impose positivity, they, among the other presented options, constitute a way to make sure, under certain assumptions, that there will be a positive solution. The assumption is, usually, that the mean mode is positive, since a filter converges the solution to it. This is in contrast with robustness focused methods where neither positivity is guarantee necessarily, nor do they act to correct a negative value if it occurs.

As such, filters are sought to be used for positivity, and a broad research question can be written as:

Can a filter be used to guarantee positivity in a FR simulation?

This question is refined to a particular model. First, in this chapter, the inviscid non reacting case (Euler equations) are tackled. This is a simpler case which should help verify that the method can be used, as well as develop the integration related knowledge for more complex or more general cases. As such, the question becomes:

Can a filter be used to guarantee positivity in a FR simulation for the inviscid non reacting case?

Given that this cannot be tested in general (for all possible cases), a set of particular cases must be chosen. The wedge case – to be defined and motivated in section 3.7 – was chosen given its shocks, which are expected to create positivity issues. This case will, then, target the research question.

FR in COOLFluiD already makes use of a positivity method – to be described in section 3.3 – and, as such, it should be expected that the new positivity method, through filtering, does not lead to a different converged solution nor leads to a divergence in the same conditions as the previous method. If such were the case, it would be having a negative effect on the solution, either by distorting it or by preventing its convergence. As such, a second research question is posed.

Does the implemented filter converge to the same solution as the previous method for the inviscid non reacting case in FR?

These research question imply an implementation of a filter, motivating the development of a framework

for future implementations for more complex cases. They also allow for gaining confidence in the use of a new method in substitution of the previous one.

3.3. Previous Positivity Method

The previous positivity method has already been briefly introduced in the literature review. It is here repeated in a more complete form for clarity and analysis. The version here presented is based on the work of Vandenhoeck [2], and the notation is adapted in order to be consistent with that work.

As explained in section 2.3.2, each element in flux reconstruction includes a set of solution points, each of which hold the states of the simulation. These states are usually either the conserved variables or the primitive variables in the context of COOLFluiD. Assuming the use of conserved variables $w = [\rho, \rho\mathbf{v}, \rho e_t]$ – with ρ the density, e_t the specific energy and $\mathbf{v}$ the velocity in vector form – the positivity constraints are to keep the density and the pressure positive, and are written as follows:

$$\rho > 0, \text{ and } p = (\gamma - 1)\left(\rho e_t - \frac{1}{2}\frac{\left\|\rho\mathbf{v}^2\right\|}{\rho}\right) > 0, \tag{3.3.1}$$

with $\|\|$ the usual vector norm.

From these states, the average density $\bar{\rho}$ and the pressure of the average states $p(\bar{w})$ is computed. A small number ϵ_l is defined as:

$$\epsilon_l = \min(10^{-3}, \bar{\rho}, p(\bar{w})). \tag{3.3.2}$$

This ϵ_l is used to limit the solution. For the densities, a limited density $\tilde{\rho}_i$ is calculated at each solution point i as:

$$\tilde{\rho}_i = t_1(\rho - \bar{\rho}) + \bar{\rho} \text{ with } t_1 = \min\left(\frac{\bar{\rho} - \epsilon_l}{\bar{\rho} - \rho_{\min}}, 1\right) \tag{3.3.3}$$

where

$$\rho_{\min} = \min_j(\rho_j), \tag{3.3.4}$$

where ρ_j is the density at a flux point j .

The pressure is limited by calculating a component t_j for each flux point such that:

$$p\left(t_j(\tilde{w} - w) + \bar{w}\right) = \epsilon_l \text{ with } \tilde{w} = [\tilde{\rho}, \rho\mathbf{v}, \rho e_t]. \tag{3.3.5}$$

With this, the limiting of the solution points is done by taking $\tilde{\tilde{w}}$ as the limited value at each solution point and equal to

$$\tilde{\tilde{w}} = t_2(\tilde{w} - \bar{w}) + \bar{w} \text{ with } t_2 = \min_j(t_j). \tag{3.3.6}$$

Equation 3.3.6 finalises the implementation of the method and gives the limited solution at each solution point, which should be positive. With this clear presentation, the major limitation of this method can be addressed. If the mean modes of used to define the ϵ_l are negative, then ϵ_l must be instantiated with a small positive number, here taken as 10^{-3}. While this number can be defined as a smaller one, it is not a proper estimate of the values – as it is a fixed number that makes use of no information during the simulation besides that the mean modes are negative – but merely a numerical guess to prevent negative values.

The weakness of the positive mean mode ties into a possible extension to multiple species. The natural extension to this case is to take:

$$\rho_{is} > 0 \tag{3.3.7}$$

where ρ_{is} is the density of species s at the solution point i. As such, the method would then be extended to use the mean mode of each species $\bar{\rho}_{is}$ as a way to insure the positivity statement. However, given that densities in multispecies methods can be orders of magnitude smaller than the density of the fluid, avoiding these negative mean modes is not trivial. Furthermore, the numerical guess used in case these cases would also necessarily be appropriate to prevent a blow up.

Returning to the original formulation, while limiting with the mean mode based on the values at solution and flux points should lead to solutions closer, in some sense, to the mean mode, this process does not tackle higher order oscillations directly, unlike filtering procedures. Therefore, a filtering technique might improve upon this method.

3.4. Fejér Filter

Discontinuities in the solution, such as the ones given by shockwaves, create regions of high oscillations due to the Gibbs effect [2]. While this is indeed alleviated by the artificial viscosity, negative solutions can still take place near these zones. As such, filters which are designed for or tackle directly the Gibbs effect are of interest. Gottlieb and Shu [83] details how filters whose kernel is always positive tackle the Gibbs effect. This includes the Fejér filter as well as the exponential filter. The latter is tackled in section 3.5, with an implementation in the modal polynomial domain, rather than in the Fourier one seen in Gottlieb and Shu [83].

The Fejér filter, among the ones presented by Gottlieb and Shu [83], was chosen due to its possible implementation as a convolution filter directly on the solution states, making its implementation possible without any transformations to the Fourier domain. The definition and derivation of the FR specific implementation are given in this section.

The Fejér convolution is defined as:

$$F_n(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f_{\text{example}}(x') S_n(x - x') \mathrm{d}x' \tag{3.4.1}$$

for some function real f_{example} defined on the interval $[-\pi, \pi]$, assuming that this integral exists. $S_n(x)$ corresponds to the Fejér kernel, and it is defined as [84]:

$$S_n(x) = \frac{1}{n} \left(\frac{\sin(\frac{n\pi}{2})}{\sin(\frac{x}{2})} \right)^2 . \tag{3.4.2}$$

Let g be a linear transformation from the interval $[-\pi, \pi]$ to the interval [-1,1] which admits an inverse. This allows the definition of the Fejér filter on the standard element. Two orthogonal coordinate systems are defined: (x, y) on the standard element and (α_c, β_c) on the transformed standard element from [-1,1]x[-1,1] to $[-\pi, \pi]$x$[-\pi, \pi]$. Given that the values in the standard element are defined on the [-1,1]x[-1,1] square, the function $\mathscr{k}$ is here taken as a real function defined on [-1,1] instead of f_{example}.

For the 2D case with structured meshes, this formulation can be easily extended to a double integral:

$$F_n(x,y) = \left(\frac{1}{2\pi}\right)^2 \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} \hbar(g(\alpha_c'), g(\beta_c'))S_n(g^{-1}(x)-\alpha_c')S_n(g^{-1}(y)-\beta_c')\mathrm{d}\alpha_c'\mathrm{d}\beta_c'. \tag{3.4.3}$$

Then the Fejér filter can be defined in the same interval as a standard element as:

$$F_n(x,y) = \left(\frac{1}{2\pi}\right)^2 \int_{-1}^{1}\int_{-1}^{1} \hbar(x',y')S_n(g^{-1}(x)-g^{-1}(x'))S_n(g^{-1}(y)-g^{-1}(y')')J_g^2\mathrm{d}x'\mathrm{d}y', \tag{3.4.4}$$

where J_g is the Jacobian of the linear transformation g, which is constant due it being linear.

Let $\hbar(x,y)$ represent some scalar quantity of interest in the standard element. Then, due to the definition of the flux reconstruction method on square elements, it can be represented by a sum of Lagrange polynomials as:

$$\hbar(x,y) = \sum_i^{N_p}\sum_j^{N_p} \phi(x_m,y_k)l_j(x_m)l_i(y_k), \tag{3.4.5}$$

where x_m and y_m are the coordinates of the solution points which the Lagrange polynomial interpolates, and $\phi(x_m,y_k)$ is the evaluation of the scalar quantity at those points. This can be substituted into our definition of the filter

Given that the integration is numerical – and in the case of flux reconstruction, it can be done with Gauss–Legendre quadrature given that the points are places in the Legendre points – the integral can be swapped for a summation. The integration of some function $h_{\text{example}}(x,y)$ samples at the solution points (x_m,y_k) would be:

$$\sum_k^{N_p}\sum_m^{N_p} h_{\text{example}}(x_m,y_k)w_{\text{quad},k}w_{\text{quad},m}, \tag{3.4.6}$$

where $w_{\text{quad},k}$ and $w_{\text{quad},m}$ are quadrature weights.

Substituting the Lagrange polynomial formulation and the numerical quadrature into equation 3.4.4, one obtains:

$$F_n(x,y) = J_g^2\left(\frac{1}{2\pi}\right)^2 \sum_k^{N_p}\sum_m^{N_p}\sum_i^{N_p}\sum_j^{N_p} \phi(x_m',y_k')l_j(x_m')l_i(y_k')S_n(g^{-1}(x)-g^{-1}(x_m'))S_n(g^{-1}(y)-g^{-1}(y_k')')w_{\text{quad},k}w_{\text{quad},m}. \tag{3.4.7}$$

Given that, by construction, the Lagrange polynomials are either 1 or 0 at the solution points, the solution simplifies to:

$$F_n(x,y) = J_g^2\left(\frac{1}{2\pi}\right)^2 \sum_k^{N_p}\sum_m^{N_p} \phi(x_m',y_k')S_n(g^{-1}(x)-g^{-1}(x_m'))S_n(g^{-1}(y)-g^{-1}(y_k')')w_{\text{quad},k}w_{\text{quad},m}. \tag{3.4.8}$$

This last formulation can be applied directly to the states using simple loops, and constitutes a much simpler version of the initial analytical formulation. The n can also be understood as the strength of the filter.

The linear transformation g to be used can be taken as:

$$g(x) = \frac{1}{\pi}x, \tag{3.4.9}$$

and $J_g = \pi$.

3.5. Exponential Filter

The exponential filter follows the form given in the work by Dzanic and Witherden [73]. This form was already briefly discussed in section 2.3. From an intuitive perspective, the idea behind using the filter centres on filtering the states in an element until all states are positive. If the all the mean mode values for the states are positive, then this solution works – as the filter will at max strength, by definition, return a mean mode.

The definition of the filter is firstly given in section 3.5.1, while its implementation is discussed in 3.6.

3.5.1. Defining the Exponential Filter

Returning to the formulation of the FR method given in section 2.3.2, it is noted that each standard element Ω_s includes N_p solution points, as previously detailed in figure 2.2. Each solution point is associated with primitive variables, which for 2D would could be defined as $W = \rho_0, ..., \rho_n, v_x, v_y, T, Tv$, possibly with the omission of Tv. Both the conserved variables U and the fluxes F are functions of the primitive variables W.

The filter will act on the entire element, on all solution points, for each primitive variable. It must be recalled that in the FR framework, the solution over an element is given by the Lagrange interpolation over solution points. As such, the solution at the solution points completely determines the solution over the element.

As notation note, this section does not make use of the summation convention for repeated indexes which can be found in aerodynamics literature. All summations are given explicitly. The primitive variable w in an element is defined over the element as:

$$w = \sum_{i=0}^{N_p} w_i l_i, \tag{3.5.1}$$

with l_i the Lagrange polynomial basis. To apply the filter, a modal representation is required. As such, w can instead be defined as:

$$w = \sum_{i=0}^{P} \hat{w}_i \psi_i, \tag{3.5.2}$$

where ψ_i is a function of the modal basis and $\hat{w}_i$ its coefficient. The modal basis is obtained from the Lagragean basis through a basis change.

Returning to equation 2.3.30 (rewriten and adapted in equation 3.5.3 for convenience), the filter is then defined on the coefficients of this modal representation:

$$\tilde{w} = \sum_{i=0}^{P} \mathscr{H}_i(\hat{w}_i)\psi_i, \tag{3.5.3}$$

with $\mathscr{H}_i$ being the exponential filter. For computational and convenience reasons, the filter is adapted to:

$$\mathscr{H}_i(\hat{w_i}) = \hat{w_i} \cdot \xi_{\text{strength}}^{i^2},\ \xi_{\text{strength}} \in [0,1],\ i \in \mathbb{N} \setminus \{0\}, \tag{3.5.4}$$

$$\mathscr{H}_0(\hat{w_0}) = \hat{w_0}. \tag{3.5.5}$$

This filter filters the higher frequencies more than lower frequencies, which lowers the high numerical oscillations, and should help acquire a positive value. The filter strength ξ_{strength} has values between 0 and 1. As such, it fully filters the solution when set to 0, and does not filter at all for a strength of 1.

While this mathematical formulation is useful, it does yet fully solve the problem. The filter, for it to be useful, should return a filtered value $\tilde{w}_i$. As such, the filtered solution could be defined as:

$$\tilde{w} = \sum_{i=0}^{P} \tilde{w}_i l_i, \tag{3.5.6}$$

which is the useful form for the FR method. As such, a solution for $\tilde{w}_i$ should be found. As seen in 3.5.6, the filtered values are the coefficients of the Lagragean basis and, as such, it can be obtained through a change of basis. Hence:

$$\tilde{w}_i = \mathcal{V}(\mathscr{H}_0(\hat{w_0}), ..., \mathscr{H}_P(\hat{w_0})) = \mathcal{V}(\mathscr{H}_0(\mathcal{V}^{-1}(w_0)), ..., \mathscr{H}_P(\mathcal{V}^{-1}(w_P))), \tag{3.5.7}$$

where $\mathcal{V}$ is the change of basis function. If $\mathcal{V}$ is a linear transformation, equation 3.5.7 can be recasted in matrix form as:

$$\begin{bmatrix} \tilde{w}_{i0} \\ \vdots \\ \tilde{w}_{iN} \end{bmatrix} = \mathcal{V} \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \xi_{\text{strength}}^{1} & 0 & \cdots & 0 \\ 0 & 0 & \xi_{\text{strength}}^{4} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \xi_{\text{strength}}^{P^2} \end{bmatrix} \mathcal{V}^{-1} \begin{bmatrix} w_{i0} \\ \vdots \\ w_{iN} \end{bmatrix}. \tag{3.5.8}$$

Equation 3.5.8 is the matrix form of the exponential filter, and the one to be used, as it filters each primitive variable accordingly. The matrix $\mathcal{V}$ maps the modal basis to the Lagrangian basis. For the case of quadrilateral 2D meshes, the modal basis is the Legendre polynomials.

3.6. Implementation

This section regards the implementation of the filtering techniques for FR for the Euler case. It concerns both the developed methods and code as well as the framework into which the code was integrated.

3.6.1. Abstract Implementation

The implementation of the filter follows the simple idea that the filter should have enough strength to for the solution inside an element to comply with some set of constraints. This requires that both this set and a form to find the strength of the filter are defined. In accordance with Dzanic and Witherden [73], the set of constraints contains the following two constraints:

$$\Gamma_0(w) = p - p_{\min}, \quad \Gamma_1(w) = \rho - \rho_{\min}. \tag{3.6.1}$$

To these constraints, a third one is added based on entropy

$$\Gamma_2(w) = s - s_{\min}. \tag{3.6.2}$$

where s denotes the specific entropy.

The constraint condition is then defined as:

$$\Gamma_0 > 0\,, \Gamma_1 > 0\,, \Gamma_2 > 0. \tag{3.6.3}$$

The variables $\rho_{\min}$ and $p_{\min}$ are constant values given by the user, and set to a small positive number or 0. The small positive number serves to avoid the influence of numerical errors. The $s_{\min}$ is defined, instead, as the smallest entropy value of the element and its neighbouring elements. Therefore, it must be updates at every time step. This method makes sense for when the CFL is smaller than 1 and the evolution is physical. When such happens, waves only come from neighbouring elements, and the entropy therefore cannot go bellow $s_{\min}$ without violating the second law of thermodynamics. Numerical oscillations which lower the temperature at certain solution points might lead to lower entropy, thereby violating this constraint and compromising the physical evolution. Given that Dzanic and Witherden [73] applied this method to cases with explicit time integration, such constraint is to be expected.

However, for implicit methods with higher CFL, this might not be valid and, as such, this constraint should not have to be always active in a successful implementation. This seems to suggest to make a dynamic option which can be turned on and off during runtime.

Another issue might come from heat transfer at the wall. Lowering the temperature of the flow through heat transfer is a physical way to lower entropy (while increasing it elsewhere, thereby either maintaining constant or increasing the entropy of the entire system). However, entropy at the wall is not taken into account, leading to a mismatch between the formulation of the constraint and its application. However, since it might have a good applicability for the rest of the flow, namely near shocks, it can still be of interest.

For the Euler case, ideal gas is assumed, as well as the use of conserved variables for the simulation. As such, the constraints $\Gamma_1 > 0$ is given by equation:

$$p - p_{\min} = (\gamma - 1)\left(\rho e_t - \frac{1}{2}\frac{\left\|\rho \mathbf{v}^2\right\|}{\rho}\right) - p_{\min} > 0. \tag{3.6.4}$$

Since

$$s = c_v \ln\left(\frac{p}{\rho^\gamma}\right) + \text{constant} \tag{3.6.5}$$

for ideal gas with constant heat capacity at constant volume c_v, and with ln the natural logarithm being a monotonic increasing function. Since there is only an interest in comparing whether it increases or not, the entropy constraint is given a simplified way instead by:

$$s - s_{\min} = \frac{p}{\rho^\gamma} - s_{\min} > 0. \tag{3.6.6}$$

Equation 3.6.5 can be derived from the expression for entropy variation from a reference state for ideal

gases.

With the constraints better defined, a way to impose them with a filter must be defined. Figure 3.1 shows a schematic of this process. The idea is based on checking whether filtering is required for a given element and then – if it is required – finding the strength required such that the filter enables the constraints before actually applying the filter. Already sketched also in figure 3.1 is the approach to find this filter. Some sort of loop based on evaluations of the solution points' values post filtering is to be used to find which strength of the filter is enough to have the constraints met. Implicitly, an optimisation loop is being described, where the least amount of strength of the filter is sought after (as to alter the solution as little as possible) that allows the constraints to be met. The approach that will be used will be one of bracketing to be discussed in section 3.6.3.

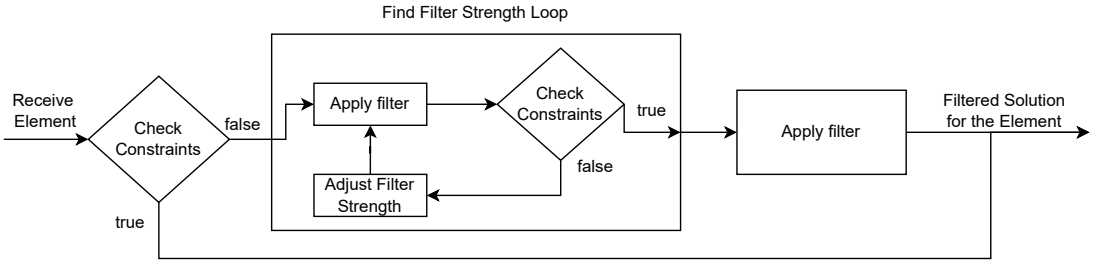


Figure 3.1: Filtering approach.

3.6.2. Framework

The positivity method to be implemented must be integrated into the larger code of COOLFluiD. The framework into which this method is implemented is given here. The focus here is on spacial discretisation part. This part may be found, at the highest level of abstraction, in the FluxReconstructionSolver class, in particular in the computeSpaceResidualImpl function [4]. This function executes the discretisation of the spatial domain, computing the divergence of the fluxes at the solution points, as well as the Jacobian for implicit methods. A simple schematic of this function is presented in figure 3.2. The positivity scheme precedes all other calculations in order to enable at the start of each iteration positive values for the calculation of the fluxes.

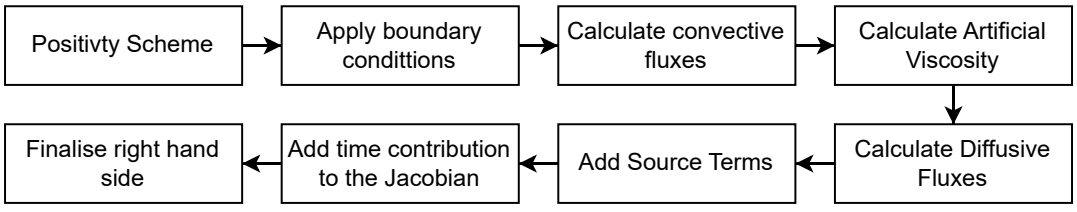


Figure 3.2: Spatial discretisation in COOLFluiD

This discretisation class is included in the wider framework of COOLFluiD as seen in figure 3.3. The positivity is highlighted in green, and concerns the part which is to be altered. The positivity is then applied in a twofold implementation. A class named BaseFiltering is defined and used as the general class for filtering. Its "execute" function is run in the "positivity scheme" part of figure 3.2 in order to ensure positivity. A specific class for filtering for Euler cases is to be, then, defined. A different class for

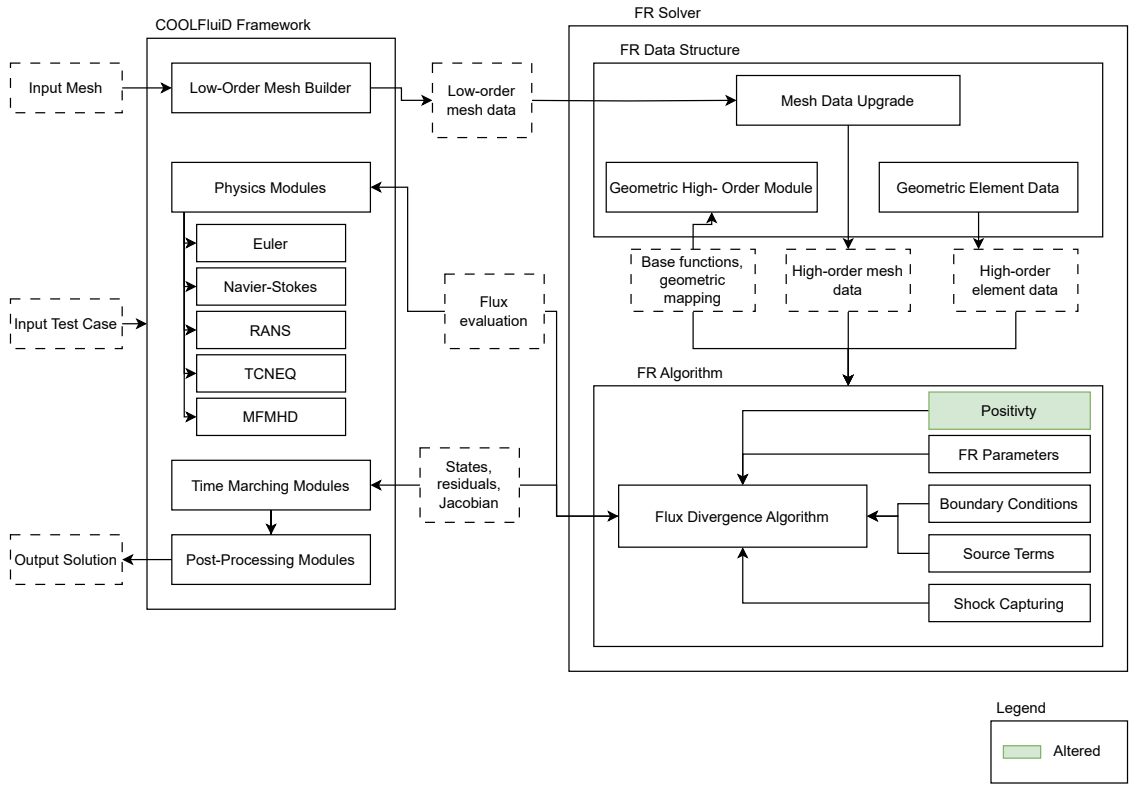


Figure 3.3: Positivity method in the wider framework of COOLFluiD. The image is adapted from the work of Vandenhoeck [2]

filtering cases for thermochemical nonequilibrium for multiple species will be addressed in a different chapter.

This twofold implementation is required since some parts of the method are physics specific – such as the ideal gas assumption for the constraints – while others are not – such as looping over the elements of the domain in order to apply positivity to them. The execute function for the BaseFiltering class is given in schematic form in figure 3.4. It loops over the elements in order to apply filtering whenever the constraints are not met for the element. It also provides access to the values stored in the solution and in the flux points during the load element part. The check of the constraints is physics specific and is not implemented in the BaseFiltering class, but it is instead delegated to a more specific filtering class. The compute filter, despite being present in the BaseFiltering class, was also delegated to a more specific class in order to make use of those class specific functions and variables. This will be detailed in section 3.6.3.

The calculation of the entropy is implemented in general for some to be defined entropy function. The exact entropy is dependent on the physics of the problem and, like the check constraints step, it is delegated to more specific filtering class. In the BaseFiltering class, the calculate entropy step loops over the neighbouring elements to the current element, calculates the entropy for each solution point in each element, and saves the smallest one.

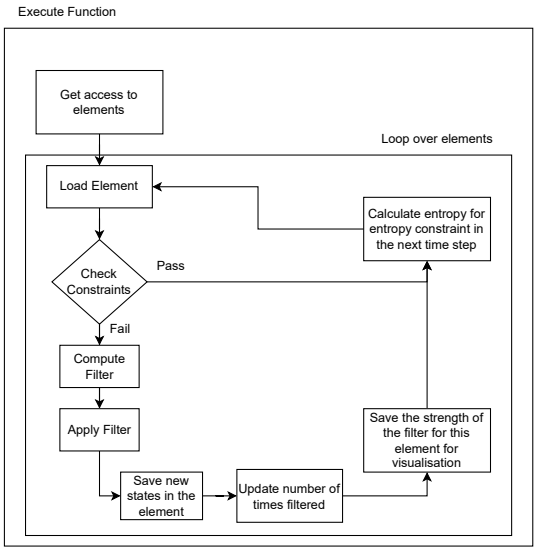


Figure 3.4: Execute filtering function used in the BaseFiltering class. Once all elements are loaded and filtered, the function exits the loop and finishes running.

3.6.3. Filtering Implementation

Given the high level view of the framework in the previous section, a path for implementation must be taken. Given that the BaseFiltering class assumes some more well defined filtering class specific to the physics of the problem, the class FilteringEuler2D is employed. It concerns the Euler cases in 2D. This class must support the calculation of the constraints, of the filter and of the required physical quantities.

The filters described in this chapter, however, do not depend on the physics and, therefore, can be implemented in the BaseFiltering class. This is also added to the class. Further alterations to the execute function are added for the support of several filter choices.

3.6.3.1. FilteringEuler2D class

The FilteringEuler2D class is subclass of BaseFiltering. This allows for it to be instantiated directly instead of BaseFiltering as the positivity method related class and to run the inherited member functions, such as the execute function. This is a common design choice used in COOLFluiD in order to enable modularity between physical models without repeating unnecessary code. As such, another filtering class for another physical model could be chosen instead for another physical model and still make use of BaseFiltering's functions as long as it is its subclass.

This filtering implementation creates a foundation for more complicated implementations, such as for more complex physics cases. Thus, the work here developed gives direct implementations in the BaseFiltering class which can be used for other physics based cases (such as with thermal and chemical non equilibrium), as well as indirect implementations for other filtering classes by asserting design ideas.

Figure 3.5 asserts this inheritance in schematic form and shows the functions present in the class, together with a summary of their use (the summary omits non relevant uses, such as the initialisation). As such, FilteringEuler2D has access to all the public and private functions and variables of BaseFiltering, as well as the member functions shown in figure 3.5.

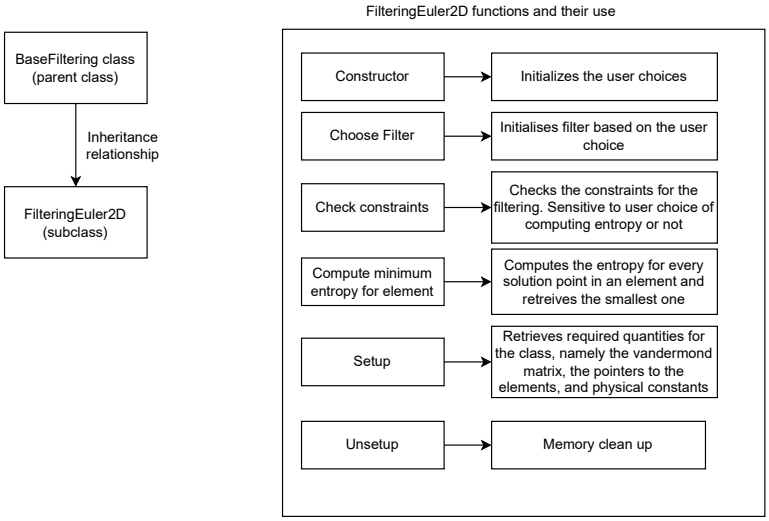


Figure 3.5: FilteringEuler2D class

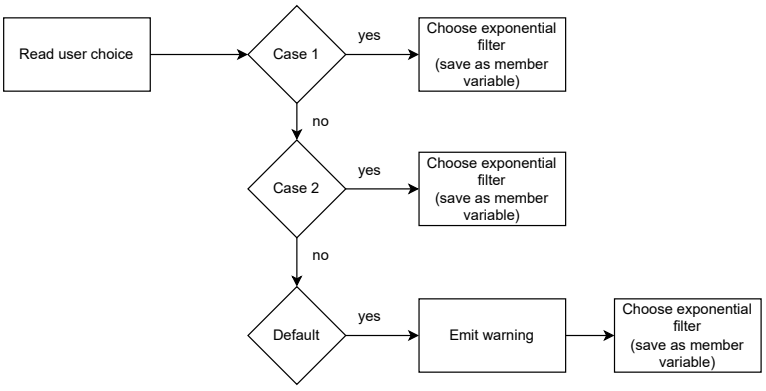


Figure 3.6: Choose filter function implementation

3.6.3.1.1 Filter Choice

The need to test several filters implies that the choice of filter is somewhere made in the code. In this case, it was chosen to add to the execute function of BaseFiltering before the loop over the elements. This choice allows the use of the same filter for all elements for the current time step even if the user changes the choice dynamically during the run. Furthermore, while for this simpler filters, it might not make sense to change during the run, filters with tuning parameters or used together with other methods might benefit from this choice. This way, the filter is only initialised once per time step, instead of per element, which would be unnecessary. The change can be seen in the new schematic in figure 3.7.

The choose filter function (see figure 3.5) takes the option given by the user and initialises a lambda function which runs the filter chosen and saves it as a member variable. This approach allows the filter to be invoked with the same variable name in all implementations in this class. Two options are given: the Fejér filter and the exponential filter, both defined in the parent class. A third default option is also given (in case of wrong user input) which defaults to the exponential filter and emits a warning to the terminal. This is presented as a schematic in figure 3.6.

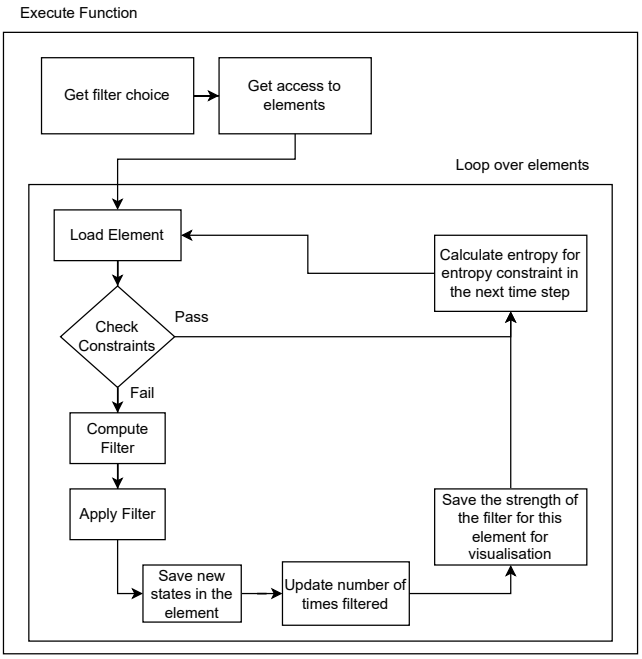


Figure 3.7: Execute function for the FilteringEuler2D class.

3.6.3.1.2 Check Constraints and Compute minimum entropy functions

The check constraints function is tackled here. For the Euler implementation it is composed of two major loops, followed by a final check. It first loops over the solution points and saves the smallest pressure, density and – if it is to be checked – entropy. The second loop concerns the flux points, for which the same procedure is done. The calculations of the pressure and entropy follow equations 3.6.4 and 3.6.5, respectively. The smallest values are compared to the constraints, which can be found in equation 3.6.3. The entropy constraint is included based on user choice. The constraint violation is returned as a boolean value.

The function to compute the minimum entropy follows a similar procedure. It loops over the solution points, saving the minimum entropy, followed by looping over the flux points, and saving the minimum entropy found. The smallest entropy among all points – both solution and flux points – is then returned.

3.6.3.2. Fejér Filter

The implementation for the Fejér filter is done in the BaseFiltering class as it does not depend on the physics of the problem. It is based on equation 3.4.8. At the time of being called, the current element is known and saved in a member variable. This was done in the execute function BaseFiltering, during the load element step. As such, the Fejér filter has access to the contents of the current filter element.

The filter loops over all solution points of the filter and applies the numerical integration described in equation 3.4.8 through the use of for loops. Importantly, it saves this in a new variable which is returned by the function. This impedes accidental overwrites of the solution states. The method is described in figure 3.8.

The kernel and variable transformations are given in abstract in figure 3.8. The kernel is computed with

equation 3.4.2. The change of transformation corresponds to equation 3.4.9.

Since the strength of the filter is an integer between 1 and infinity, the value supplied to the function, if it does not have this into account, must be adjusted. Omitted in the diagram, a strength transformation is executed in the calculation of the Fejér Kernel. The function assumes that the strength is given between 0 and 1, with 1 being the weakest (since this is what happens for the exponential filter). This is corrected by linearly mapping it to between 1 and 51 by applying the following transformation:

$$f_{\text{new,strength}} = \lfloor (f_{\text{strength}} \cdot 50 + 1) \rfloor, \tag{3.6.7}$$

where f_{strength} represents the strength.

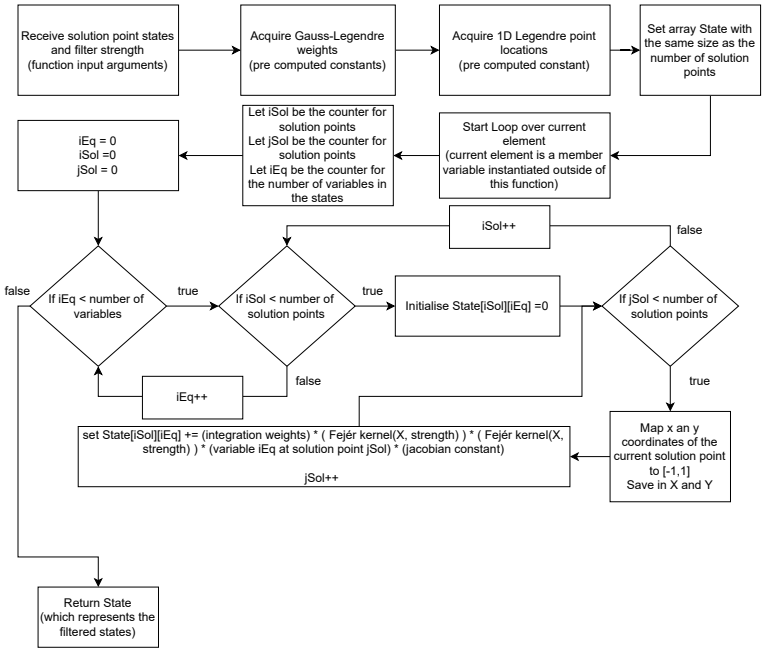


Figure 3.8: Fejér Filter function

3.6.3.3. Exponential Filter

Following the same logic as the implementation of the Fejér filter, the exponential filter function receives a filter strength and access to the solution points values. This function initialises the filter based on the order of the of the the spatial discretisation (e.g. P1, P2, ...) and the strength given, according to the description given in section 3.5.1. This filter is applied to the solution points to yield new filtered values. This implementation is given in figure 3.9. The Vandermonde matrix is assumed known – as it is already present in COOLFluiD – and it is initialised in the setup function as a member variable.

3.6.3.4. Compute Filter

In order to compute the filter strength required, a process must be defined. The idea behind computing the filter strength falls unto the following requirement: the filter should filter as little as possible. This translates to using the least amount of strength in the filter, and it stands on the idea that the filter is used to correct the solution based on constraint violation and should not be used for more than that, lest it alter the solution too much.

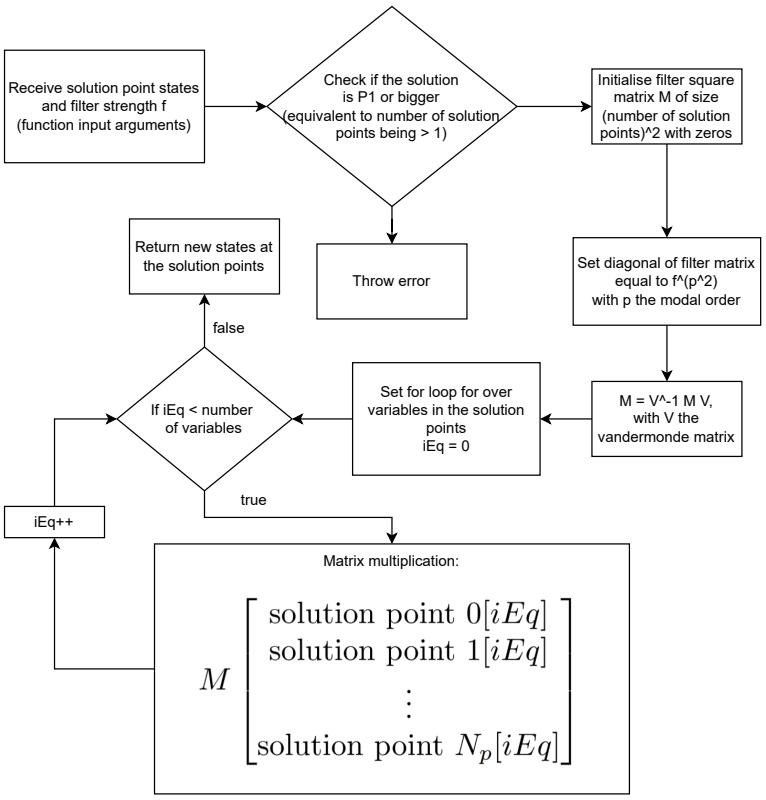


Figure 3.9: Exponential filter function diagram. The exponential filter is applied for each variable (density, momentum, etc...) to all solution points at the same time through a matrix multiplication.

As such, this requirement can be achieved, within a tolerance, by a numerical optimisation routine. This can be achieved through simple bracketing, following the same approach as Dzanic and Witherden[73]. As such, the bracketing searches the least amount of strength required for the filter such that the constraints are met. The bisection method is here employed, with the interval for the bisection search being [0,1]. For strength equals 1, the exponential filter will not alter the solution, as 1 to the power of any real number is 1. For zero, the exponential filter will filter as much as possible, leaving only the mean mode of the solution unaltered, and all other modes equal to 0. As such, these values are good candidates for delimiting the interval of strength for the filter.

For other filters, this value can be altered simply by an affine transformation. This was done for the case of the Fejér filter.

This bisection method applies the filter chosen in the chose filter function (see section 3.6.3.1.1) and tests it with the check constraints function (see section 3.6.3.1.2), before further bracketing.

3.6.4. User inputs

The method, in order to run, requires certain values which are not trivially chosen automatically, namely in between cases. The most trivial example is the minimum values for the constraints. As such, certain choices are left to the user in order to give more case by case freedom. These have, however, defaults in case they are not initialised by the user, with the goal of having already sane initialisations.

For the BaseFiltering class, the user choices were already defined and do not concern the normal functioning of the filtering method presented, with exception of the freeze iteration option, which freezes the amount of filtering after a certain amount of iterations. Since the goal is for the filter to stop being active in the end, this option is not used in practice in this thesis, and is instead initialised to a large number of iterations that will never be achieved. Due to this, the check for this variable was omitted in the diagrams concerning the execute function in BaseFiltering.

The choices for the FilteringEuler2D class are the following:

- Filter choice – choses between the exponential and the Fejér Filter. Initialised to the exponential filter.
- Minimum density for the density constraint. Initialised to 10^{-8}.
- Minimum entropy for the entropy constraint. Initialised to 10^{-4}.
- Minimum pressure for the pressure constraint. Initialised to 10^{-8}.
- Variables to be checked – chooses between using the density, pressure and entropy constraints, or just using the density and pressure ones; Initialised to include all the constraints.
- Bracketing method precision – the stop condition for the bracketing method. The Bracketing method stops once the values are less apart than this number. Initialised to 10^{-8}.

Units are not given in the initialisation values since the code merely uses this as a numerical comparison between this value and the ones in the elements and these. As such, they are the same units as the ones in the elements, which are SI units to the date of writing.

3.7. Application to the Wedge Case

The two filters, post implementation, require being tested against the previous in use method in order to ensure that they can substitute it. For such, the wedge case was chosen. It consists of an inviscid 2D case where supersonic air impacts a wedge, together with the presence of a slip wall boundary on the top of the mesh, leading to a shock successive reflections. This is complemented by the air remaining supersonic after the first shock, which leads to an expansion fan at the top of the wedge. Given that the shocks are a place for high oscillations and positivity related issues, this case serves as a way to test the filters against these issues.

The Wedge case may be found in the work of Vandenhoeck [2], where the mesh, boundary and initial conditions are taken from. However, the context of said work is for adaptive mesh refinement with flux reconstruction, whilst the current test in this document described does not concern adaptive mesh refinement.

3.7.1. Method Fundamentals

Two methods are here developed to analyse the influence of the filters. In the first method, the cases are run to convergence in a higher order simulation, complemented with a mesh refinement study. The second method consists in running the cases without AV, letting the residuals stall. The goal is to see if the filters have a negative influence on the residuals in the absence of the AV, or lead to a blow up. The case chosen should not blow up for the previous in use positivity method.

The simulations are run for the previous in use method, the exponential filter and for the Fejér filter

with the same conditions. The results for density, and energy per unit volume are to be plotted and compared. Comparisons between the fields with the new methods and the old ones are made through the use of the relative difference for each grid point. A root means square for an ensemble difference quantisation can be used in case the relative difference is non zero. As it will be seen in the next sections, this will not be the case.

Section 3.7.2 details the setup of the case from pre running standpoint – that is, it details the mesh and the boundary and initial conditions of the simulations. Section 3.7.3 concerns the simulations setup that does not concern the mesh and boundary condition. Section 3.7.4 concerns the presentations of the data obtained and its analysis. Section 3.7.5 presents the mesh convergence analysis study. The second part of the analysis – the no AV cases – is presented in section 3.7.6. A summary and conclusion is given in section 3.7.7.

3.7.2. Mesh and Conditions

The mesh used in this case is presented in figure 3.10, together with its physical dimensions. It consists essentially of one by three meter domain with a wedge on the bottom part of the mesh at 0.5 m from the inlet. The wedge increases the height by 0.13 m, and spans the length of 0.5 m. The mesh is purposely coarse, typical of higher order methods, as it is made with the goal of being used for running higher order simulations. It consist, then, of 3710 quadrilateral elements. There is not boundary layer related refinement as the this mesh is made to run inviscid cases.

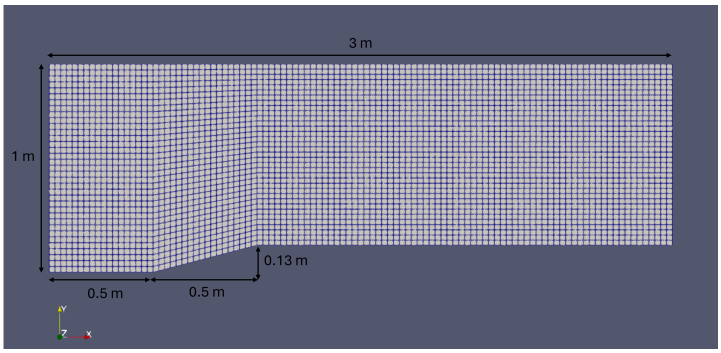


Figure 3.10: Wedge mesh with physical dimensions.

The boundary conditions may be found in figure 3.11. Black lines indicate the faces of the mesh on which each boundary conditions is applied. The inlet is supersonic for this case as the flows enters supersonically in the domain. A slip wall is used on the bottom part of the mesh, enforcing the existence of the wedge. The wall has no slip due to the flow being inviscid. The outflow is considered supersonic as, for this method, it is expected that the flow remain supersonics at the exit. This assumption is in accordance with the work of Vandenhoeck [2]. The top part of the boundary of the mesh is a slip wall, allowing for a shock reflection in this boundary. This will lead to a stronger shock near this boundary as well as a reflection of this shock in order to ensure that the velocity remains parallel to the boundary.

The initial conditions are given as the same for every inner element of the domain. These are: $\rho = 1$ kg m^{-3}, $\rho u = 2.366431913$ kg m^{-2} s^{-1}, $\rho v = 0$ kg m^{-2} s^{-1} and $\rho E = 5.3$ J kg m^{-3}, with ρ being the density, u the horizontal velocity, v the vertical velocity and E the energy of the fluid.

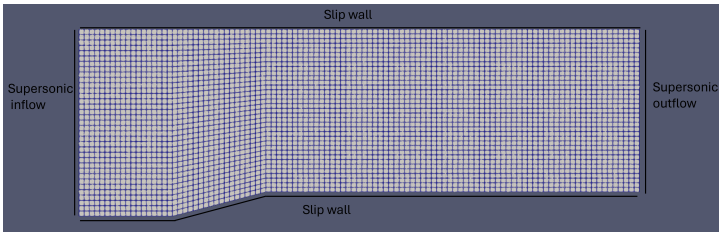


Figure 3.11: Boundary conditions for the Wedge case.

3.7.3. Simulation Conditions

This section details the setup of the simulations ran. All simulations were run with the base setup described in table 3.1. It concerns the spacial and temporal discretisations used, as well as the integrators. It also concerns the physical model and variable types. The boundary and initial conditions were already previously given and are, hence, omitted.

An implicit method for integrating time is used in order to achieve a stable evolution of the solution in time, due to the higher order nature of the FR simulation. A backwards Euler is system is chosen, which is then solved with the PETSc library. An additive Schwarz method is used as preconditioner. This follows the typical in house procedure for implicit simulations, already tested in several test cases for Euler and Navier Stokes physical models. These test cases may be found in the testcase folder for Navier Stokes in the code repository [4]. It has also been typically used in FR simulations for Navier Stokes [2]. The AUSM+ Riemann solver was chosen based on the fact that it is suitable for supersonic cases with shocks at high orders [2].

The conditions for the positivity methods may be found in table 3.2. These were chosen such that the minimum values are several orders bellow the typical values found in the simulation. The bracketing is also several orders of magnitude bellow the typical strength of a filter (between 0 and 1).

Table 3.1: Simulation base setup for the Wedge case simulations. Boundary and initial conditions were omitted here (previously described in section 3.7.2) as well as positivity method due to it being the parameter being studied.

Parameter	Choice
Physical Model	Euler2D
Stop Condition	Relative norm: -6
Time Integrator	Backwards Euler
Preconditioner	PCASM
KSPT	KSPGMRES
Convective Fluxes	ConvRHSJacob
Time Contribution on Jacobian	StdTimeRHSJacob
Artificial Viscosity	LLAVJacob
Update Variable	Conservative
Solution Variable	Conservative
Riemann Solver	AUSMPLUSFlux2D
VCJH Factor	0.3
CFL	function: $\min\left(0.8, 0.05 \cdot 1.2^{\max(\text{iteration}/100\text{-}1,0)}\right)$

The runs are presented in table 3.3. The artificial viscosity is directly tied to the order of the simulation, requiring higher values of strength and activation parameter S0 at higher orders [2]. Two runs are made in P2, making use of different AV paraments. Run 2 makes use of a stronger AV with a wider range of influence, leading to an increased effect in the solution.

Table 3.2: Parameters chosen for the filters an the previous in use positivity method.

Previous Positivity		Filter	
Parameter	Choice	Parameter	Choice
Minimum Density	$1 \cdot 10^{-5}$	Minimum Density	$1 \cdot 10^{-8}$
Minimum Pressure	$1 \cdot 10^{-5}$	Minimum Pressure	$1 \cdot 10^{-8}$
		Minimum Entropy	$1 \cdot 10^{-4}$
		Bracketing Method Tolerance	$1 \cdot 10^{-8}$
		Quantities to be checked	pressure, density and entropy

Run 3 is run from the initial conditions (no restart) with higher AV to compensate for the additional instability of being P3. Relations for the estimation of the S0 required may be found in the work of Vandenhoeck [2].

Table 3.3: Artificial viscosity parameters and polynomial order of the FR method for the simulations ran.

Run number	Order	AV Péclet	AV S0	AV kappa
1	P2	20	-2	1
2	P2	15	-2	1.5
3	P3	10	-3	1.5

3.7.4. Analysis of the Results

This section concerns the analysis of the results for the wedge case. All runs reported had the positivity methods active during the simulation, and, in case of convergence, did not have them active at convergence.

Table 3.4: Convergence results of the simulation. Converged means that the simulation stopped due to the target residuals being achieved. All residuals in the converged simulations were bellow either -6 or -7 (log scale).

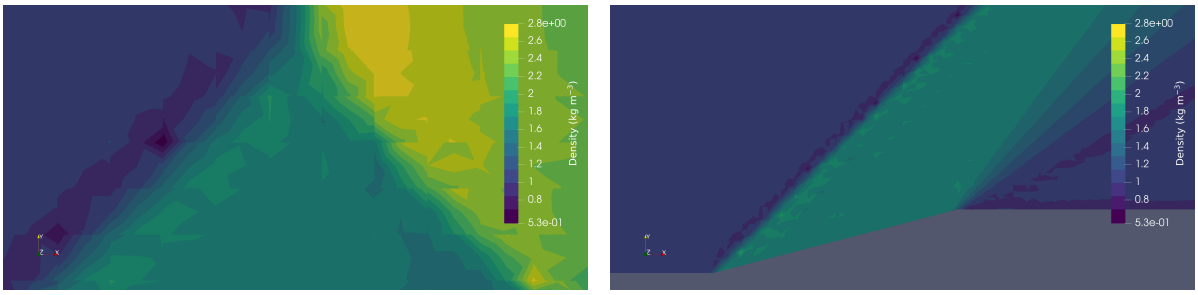
Run number	previous in use method	Exponential Filter	Fejér
1	Converged	Converged	Diverged
2	Converged	Converged	Converged
3	Converged	Converged	Converged

3.7.4.1. P2 case

For P2, convergence was found for all methods in run 2, but only for the exponential filter and the previous in use positivity for run 1. Run 1, having lower AV, may have more unstable tendencies, since instabilities are not as damped by AV as in run 2. It can be seen, then, that the Fejér filter may lead to a more unstable solution than the previous in use method, possibly by destabilizing it more than the previous in use method during its activation. Only run 2 is here reported further, as to allow further comparison between all of the filters.

The exponential filter density may be found in figure 3.12. Numerical artifacts can be seen along the shock wave. An attached oblique shock wave is seen on the wedge, followed by two reflections. Density increases after each shock wave. An expansion fan is clearly seen at top of the wedge, lowering the density. These qualitative results are in line with the ones found by Vandenhoeck [2]. Details of the numerical artifacts may be seen in figures 3.13a and 3.13b. These do not seem to be aligned with the mesh and are possibly related with the oscillations inherent to higher order methods – which also explains their appearance near the shock wave regions.

Figure 3.12: Run 2: wedge case with the exponential filter for run 2.



(a) Run 2: wedge case with the exponential filter for run 2. Detail of the reflection of the shock at the top boundary.

(b) Run 2: wedge case with the exponential filter for run 2. Detail of the attached shock on the wedge.

Figure 3.13: Run 2: wedge case with the exponential filter for run 2 – details.

The density plots for the Fejér and the previous in use method are presented, respectively, in figures 3.14 and 3.15. They show exactly the same features as the case for the exponential filter.

Figure 3.14: Run 2: wedge case with the Fejér filter for run 2. Detail of the attached shock on the wedge.

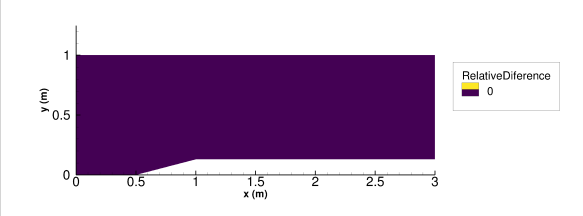
The relative change in density to the previous in use method may be found by taking:

$$\text{relative change} = \frac{\rho_{\text{filter}} - \rho_{\text{previous in use method}}}{\rho_{\text{previous in use method}}} \tag{3.7.1}$$

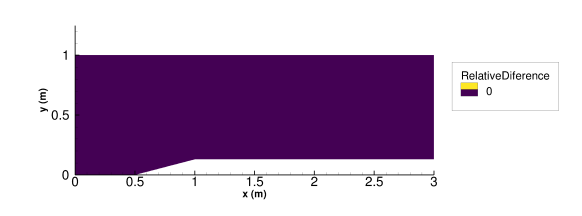
for each point. These are plotted in figures 3.16a and 3.16b. The methods all converged to exactly the same values, which is to be expected since the simulations were run under the same conditions and the positivity methods were not active at convergence.

For the P2 case, based on runs 1 and 2, it can be seen that the Fejér filter has a more destabilising effect than the other methods. The relative difference found between the plots was 0. The result for density

Figure 3.15: Run 2: wedge case with the previous in use method for run 2.



(a) Run 2: relative difference between the case with the exponential filter and the case with the previous in use method.



(b) Run 2: relative difference between the case with the Fejér filter and the case with the previous in use method.

Figure 3.16: Relative difference plots between the new method and the previous in use method.

is presented in figures 3.16a and 3.16b. The same result was found for all the other variables. It can be seen that the positivity methods do not affect the converged result as long as they are not active at convergence – at least for the these particular runs. If the steady state solution exists and is unique, however, it seems that the path to it should not matter for a well converged enough solution, which would suggest that this would be the general case. However, this is not proven here in general.

3.7.4.2. P3 case

The density plots for the P3 cases may be found in figures 3.17, 3.18 and 3.19. All simulations converged with respect to the stop condition of a relative norm of the residuals of -6. As such, all residuals had a norm of bellow -6 or -7. The same structures are found.

The first reflected shock wave is at a different angle relative to P2, leading to a shock further away from the inlet than before. This might be related to not having achieved spatial convergence with the P2 case.

In the same way as the P2 case, the end result was not affected by the use different positivity methods. As such, the relative difference found was 0 everywhere.

The difference in angle and in density between P2 and P3, as has already been addressed, suggests to a under-converged simulation. As such, this motivates a convergence study to find that the new positivity methods can substitute the previous in use positivity ones for fully converged ones. This is dealt with in the section 3.7.5, which follows from this one. Only the P2 case is used, as further refining the mesh of a P3 method will not lead to better results than a converged mesh for P2. It should, however, converge spatially faster than P2. However, it is sufficient to spatially converge one higher order simulation for the Wedge case to show the feasibility of the substitution of this method.

Figure 3.17: Run 3: density plot for the exponential filter method.

Figure 3.18: Run 3: density plot for the Fejér filter method.

3.7.5. P2 Convergence Study

Spatial convergence study is required in order to see whether fully converged solutions can be achieved with the new methods in the same way as with the previous in use method. As such, the mesh was refined three times, producing meshes with double, quadruple and octuple the elements. These simulations were run for the P2 case. These can be found in table 3.5.

The simulations were run with the same conditions as run 2. All simulations were found to converge with all residuals bellow -6 and -7. A slice was taken along the horizontal at y coordinate 0.5 m – passing over all shocks and the expansion fan – showing a clear convergence of the shock position and jumps for run 6 and 7. This can be seen in figures 3.20, 3.21, 3.22 and 3.23. The overshoot and undershoot are clearly visible near the shocks. All simulations follow the same path better the shocks, with the

Figure 3.19: Run 3: density plot for the previous in use method.

Table 3.5: Mesh refinement P2 runs.

Run number	previous in use method	Exponential Filter	Fejér	Mesh Refinement
5	Converged	Converged	Converged	x2 elements
6	Converged	Converged	Converged	x4 elements
7	Converged	Converged	Converged	x8 elements

shocks being the places of greater disagreement. The simulation without refinement (run 2) was found to agree much better on the first shock with the other simulations than for subsequent shocks. This is due to the angle of the reflected shock being wrong, leading to a shock nearer to the inlet than the others. This is agreement with the comparison done between P3 and P2 in the previous sections.

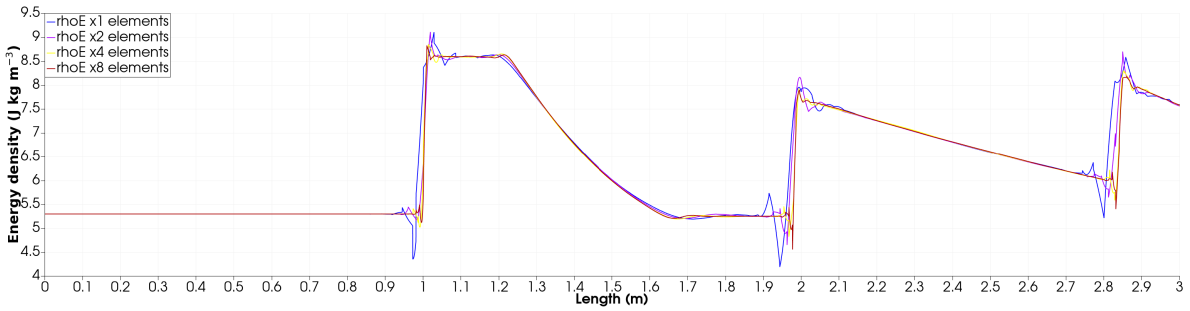


Figure 3.20: Energy density slice for P2 runs with exponential filter.

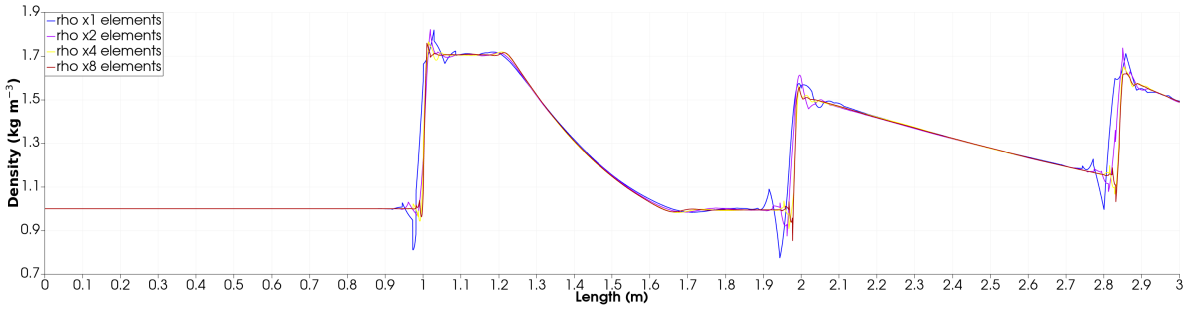


Figure 3.21: Density slice for P2 runs with exponential filter.

The analysis for the difference between the exponential filter and the old filter was repeated. No discrepancy was found, with the same exact values being reach at the end of the solution, in all ways analogous with the results for run 2. The obtained difference plots are identical to figure 3.16b. As such, they are not repeated here. The same was found for the Fejér case. As such, figures 3.20, 3.21, 3.22 and 3.23 concern not only concern the exponential method run, but also all other method runs, being identical.

For comparison with run 2, figures 3.24 and 3.25 show the result for the mesh with 8 times as many elements. The shock waves are thin and well defined. Gibbs phenomena related effects can be seen near the shock wave, but are not propagating through the domain. There is a clear difference in the placement of the first shock reflection, being more towards the left, relative to the unrefined P2 simulation. These clear changes further strengthen the necessity of refining the mesh.

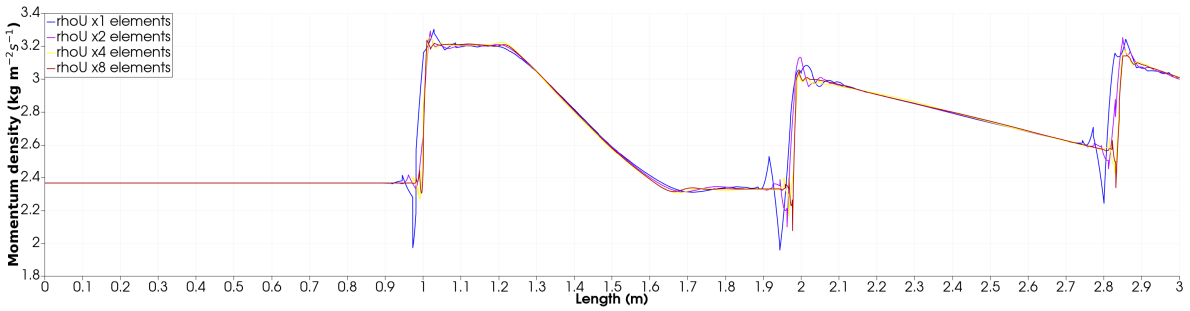


Figure 3.22: Momentum density in the x direction slice for P2 runs with exponential filter.

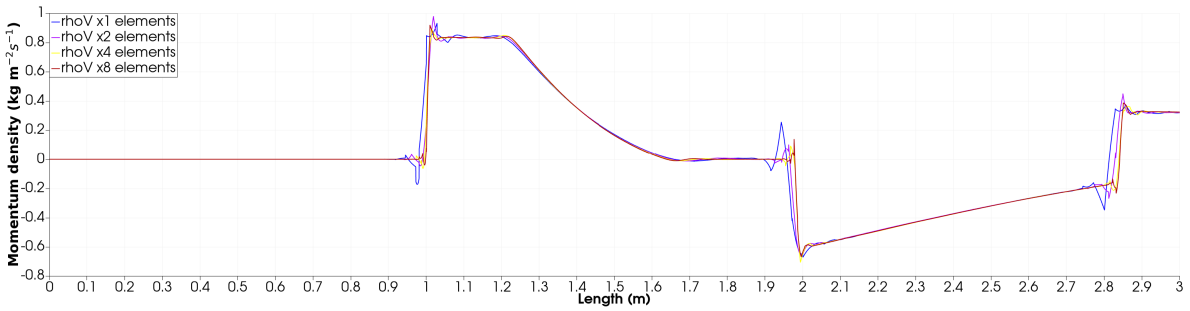


Figure 3.23: Momentum density in the y direction slice for P2 runs with exponential filter.

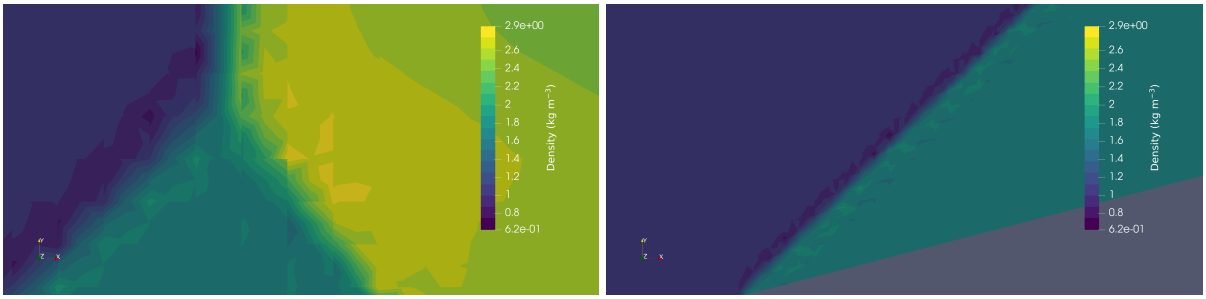
3.7.6. No Artificial Viscosity Case

In order to see the direct influence of the methods on the solution, unaltered by AV, P2 runs were executed without AV. These runs are not expected to converge, but rather to stall the residuals. Giving that the order is merely P2, it is expected that these simulations should not necessarily blow up.

The goal is to see if the residuals evolve all in the same way for all simulations and if the end solution is the same. Given the diverging behaviour with the Fejér filter in the previous sections, it is expected that some change could occur for this filter.

These simulations have a stop criterion of 30000 iterations, more than ten thousand above the required to converge with AV. While this is a conjecture that it will be enough for the residuals to stall, it will be confirmed during the analysis of the results. The CFL was changed to abide by the function $\min(4, 0.05 \cdot 1.2^{\max(\text{iteration}/150\text{-}1,0)})$.

Figure 3.24: Run 7 density plot.



(a) Run 7 density plot: detail of the shock reflection.

(b) Run 7 density plot: detail of the attached shock on the wedge.

Figure 3.25: Run 7 density plot: details

3.7.6.1. Results

Table 3.6 concerns the end results for the cases with no AV. The residuals stalled for the runs with the previous in use method and the exponential filter, while the one with the Fejér filter diverged.

Table 3.6: Convergence results of the simulations with no AV. Converged means that the simulation stopped due to the target residuals being achieved. All residuals in the converged simulations were bellow either -6 or -7 (log scale).

Run number	previous in use method	Exponential Filter	Fejér
4	Stalled	Stalled	Diverged

Figures 3.26a, 3.26b and 3.26c show the residuals of the simulations with no AV. The residuals for the cases with the exponential filter and with the previous in use method show a similar evolution, increasing until the CFL stops increasing, and then lowering to a plateau. The exponential filter run shows lower residuals overall relative to the previous in use method one. The plateau occurs for both simulations around the 5000 iteration mark. The residuals in the figures concern the residuals for the mass, the momentum equation in x, the momentum equation in y and the energy equation, in this order.

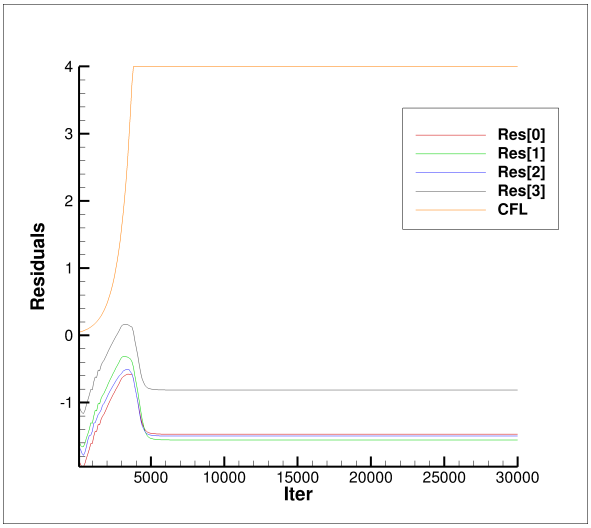
Figure 3.27 concerns the wedge with the exponential method. Well defined shock waves are found. Like in P2, the same three shock and one expansion fan are found also. The range of values is very similar to the one found in P2.

3.7.7. Wedge Case Conclusion

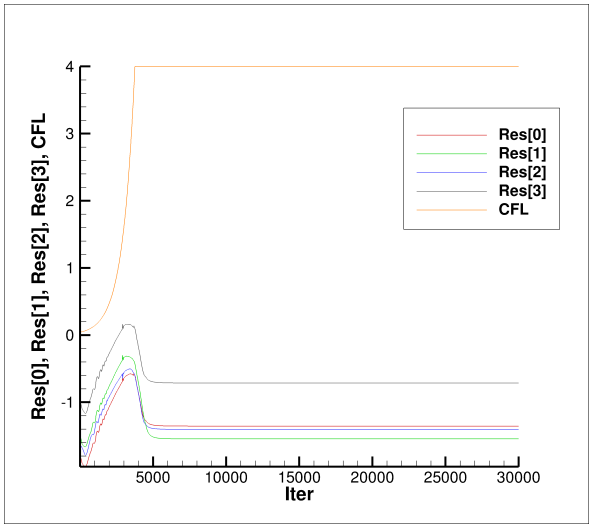
The Wedge case was selected as a testing case for the substitution of the previous in use positivity method for a filtering based one on the accounting of having several reflected shock wave, causing positivity issues. Two possible filters were tested for this new approach: the exponential and the Fejér filters.

Simulations were first run for order P2 and for order P3 using the same mesh. P2 simulations revealed that the Fejér filter diverged in the same conditions that the previous in use method converged. A second P2 simulation run with higher AV allowed all simulations with all methods to converge. This showed a greater dependency on AV tuning for the Fejér filter. A possibly negative effect on convergence, relative to the previous in use method, for the Fejér filter was, hence, detected.

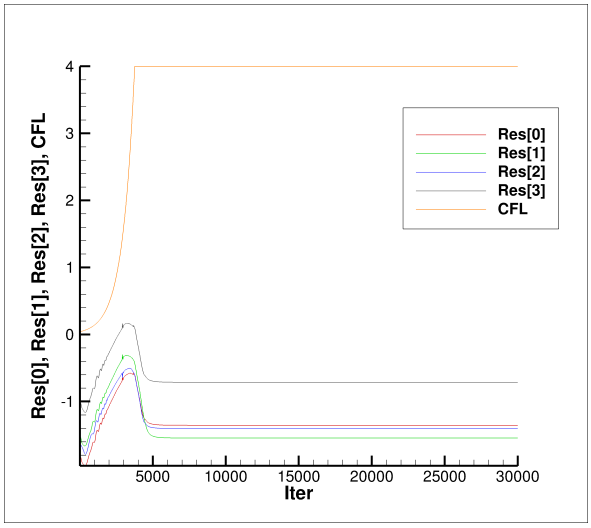
Given residual convergence (relative residual norm of -6), no change between the solutions was found. This seems to suggest that the positivity method, as long as its not active at convergence, allows for the simulation to converge to the same solution. No positivity method was ever found to be active at convergence, which means that the solution was positive in the entire domain at convergence.



(a) Run 4: Residuals of the P2 with previous in use method and no AV.



(b) Run 4: Residuals of the P2 with previous in use method and no AV.



(c) Run 4: Residuals of the P2 with Fejér filter method and no AV.

Figure 3.26: Residuals for Wedge case runs with no AV.

A discrepancy in the maximum density and the shock wave position was found between the P2 and P3 runs, suggesting that the simulations were not fully spatially converged. It was elected to converge the P2 simulation through mesh refinement. The mesh was refined by doubling the elements each iteration,. An horizontal slice across the midle of the domain was used to compare shock wave position and jumps. The simulations with four times and eight times as many elements seem to agree on the shock wave position. They plots for these two simulations do not overlap completely due to Gibbs effect related phenomena.

All methods were successful for the refined mesh runs. The density plot for the case with the mesh refined to eight times as many elements was presented, showing a thin and well defined shock wave and a clear expansion fan. The methods were found to allow for full convergence of the Wedge case.

This first study was supplemented by a study of running the P2 case in the original mesh without AV in order to study the methods without the influence of AV. The cases with the previous in use method and with the exponential filter run until the residuals stalled, acquiring the same end solution. The case

Figure 3.27: Run with the exponential filter P2 and no AV.

with the Fejér filter diverged.

The Fejér filter was found to have negative consequences on the run, while the exponential filter was found to work as well as the previous in use method for all tested cases. The substitution for the exponential filter was found to work for the tested cases.

3.8. Research Question Answer

In section 3.2, the research question for this chapter was presented:

Can a filter be used to guarantee positivity in a FR simulation for the inviscid non reacting case?

This research question sought to focus on the implementation and use of a filter positivity scheme for the Euler cases. It was tested through the use of the Wedge case in a higher order setting. The answer was answered positively for this particular case, with the exponential filter guaranteeing positivity. While this case is not an exhaustive test, it contains drastic enough conditions in terms of shocks to test whether it should work. It should be noted that it was also found that not all filters do work, with the Fejér filter leading to divergence of the solution in particular runs, which will be here interpreted as not guaranteeing positivity, in the sense that a positive number should be bounded.

While further tests could strengthen the answer to the question, no further were done or deemed required. It is known from the construction of the filter that, if the mean mode is negative, then the filter cannot be applied to retrieve a positive solution. Same applies for the previous in use method. Hence, it is expected that, based on the Wedge case, if the no mean mode is negative, that a filter could guarantee positivity. In contrast, if the mean mode is negative, it was found that, much like the previous method used, it can not.

Hence, for all cases for which the previous method works, it is expected that the exponential filter could guarantee positivity.

A second research question was posed:

Does the implemented filter converge to the same solution as the previous method for the inviscid non reacting case in FR?

This question was answered positively for all the converged cases for the Wedge case. This allows for greater trust in that the positivity method will not be altering the converged solution.

Both research questions were answered positively for the particular case of the Wedge case. While

generalisations are not guaranteed, it is expected that, for cases as drastic as these in terms of positivity violations due to the Gibbs effect, the exponential filter should be able to guarantee positivity and allow for the convergence to the correct solution.

3.9. Conclusion

A positive method for the thermochemical nonequilibrium cases is required. Filters have been a found in literature to be an alternative to the previous approach of limiting the states. Given their more holistic approach to the elements and the fact that they deal with high frequency oscillations directly, it was chosen to substitute the previous approach for a filter based one.

Two filters were selected: the Fejér filter and the exponential filter. Furthermore, research questions were posed about the use of these filters, namely in regards to them altering the end solution and guaranteeing positivity.

A presentation of the filters was given, together with their derivation and implementation. The Fejér filter's derivation was given in a general enough form to be applicable to any convolution filter for structured elements in FR. The filters' implementation was given both in abstract – emphasising the goals of the code and the ideas behind its functioning – and in concrete, emphasising its implementation into the wider framework of COOLFluiD and the algorithms developed from the abstract goals. The latter included not only the algorithms, but also the object oriented design executed based on the framework of the code base and on the user given control through parameter definition.

This implementation was tested against the Wedge case. This case consisted of inviscid air impacting a wedge, thereby producing a shock, which is then reflected twice on the boundaries. This case was run in a higher order formulation (P2 and P3). The exponential filter and the previous method both obtained the same results, having been used in fully converged simulations. The Fejér filter showed divergence in low AV cases, and the same result as the previous method in higher AV ones. A mesh refinement study was executed for the P2 case, showing a spatially converged case after two refinements for all methods.

A no AV case was used to isolate the effects of the method. While the previous method and the exponential filter method both resulted in the same solution with stalled residuals, the Fejér filter led to divergence. Based on these runs and on the P2 runs, the Fejér filter was deemed to be less reliable than the previous method.

The research question were answered positively in particular for the Wedge case for the use of the exponential filter. This case seems to suggest that the filter should hold in other cases with shocks where the positivity is threatened.

The development and implementation of a filter based positivity was successfully created, implemented into the code and tested against a relevant case. This, thus, allows the development of a filter based positivity with greater trust and insight into the method itself as well as its possible implementation for other models beyond the Euler model.

4

Positivity Method for Multiple Species

Following the findings of the literature review, it can be concluded that the positivity in higher order methods is an ongoing problem for which there is no current form implemented in the context of FR with thermochemical non equilibrium. As such, this chapters focuses on deriving a method to guarantee positivity.

A high level implementation is first given in section 4.1. This is the driver for the type of method that is sought to be designed, and will constrain the method. Then, an exponential filter based on literature is given and applied to our method. Within the exponential filter section (section 4.2), an entropy derivation is given in section 4.2.3. This is required for checks related to the filter. A least squares method is given in section 4.3, with the intuition of being used with the exponential filter. An energy based correction is derived in section 4.4. The full method is presented in section 4.5.

4.1. High Level Implementation and Objectives

The positivity method must be implemented within the scheme of the simulation. As such, depending on the exact goal and function, different places and dependencies are possible. The main objective for this method is:

- The method must guarantee that the thermodynamic primitive variables are positive.

Simple guidelines would be:

- The method should aim to be, relative to the main FR loop, fast;
- The method should be easy to implement in the context of COOLFluiD.

These guidelines are ill defined, and as such cannot be taken as requirements. However, they allow for comparison between available options. Two methods can be compared for which should be the fastest, or which one should take the least amount of time to implement.

Regarding the first guideline, emphasis should be put onto species densities, as they are one of the major differences between Navier Stokes cases and thermochemical nonequilibrium ones, and due to

their possibly zero value in their absence.

A further requirement is added:

- The method should not be active at convergence.

This requirement derives from the idea that LLAV should be the one responsible for enabling a stable marching and for eventually correcting the solution to have positive values. That is, the role of robustness and correcting numerical oscillations falls on LLAV. The positivity method should guarantee that the solution remains positive such that the method can march the solution without crashing the simulation. Artificial viscosity is still active at convergence.

With this framework, it was chosen to include the positivity method as a stand-alone method in between time marching steps – that is, immediately after each time marching step. This will allow each loop of the overall FR method (which starts with the fluxes and source terms calculation) to start with positive values.

This is on par with the previous method used with FR in COOLFluiD: the state limiter [2], which acts in between time marching steps. This gives better guarantees that the method can start with positive values, than if it the method was integrated in the middle of the step. If such were the case, then it would have guarantee that the after the time marching it would have positive values to begin the next iteration. Given the highly nonlinearity of the system, it is unlikely that this solution could be used.

A stand-alone method highly benefits from an implementation standpoint based on two key aspects:

- Being stand-alone, it does not require altering an already existing part of the FR method but simply adding it to the main loop.
- Stand-alone methods in between time marching steps have previously been implemented, and, as such, the code frame work is already in place. Changing the class being used currently to a new class for the new method is enough for it to be implemented.

Based on literature, the entropy based exponential filter [73] was chosen. It is a stand-alone method that can be ran after each time marching step, and it has been successfully used in the context of CFD. However, it fails in guaranteeing a positive solution. If the mean mode of a species in the element is negative, then the filter is unable to recover a positive solution. As such, a new method is proposed in section 4.3. This new method will only fail in the case of the sum of the densities of species being negative, which is a harder condition to break – as the sum of the densities being negative already seems to presuppose at least one negative mean mode – and it would reveal that the simulation is at a highly unstable stage.

A further method is proposed in section 4.4 to work with this new positivity method, and the full method is proposed in section 4.5. This latter one is to be implemented in between time marching steps.

4.2. Exponential Filter

The exponential filter follows the form given in the work by Dzanic and Witherden [73]. This form was already briefly discussed in section 2.3. From an intuitive perspective, the idea behind using the filter centres on filtering the states in an element until all states are positive. If all the mean mode values for the states are positive, then this solution works – as the filter will at max strength, by definition, return a mean mode.

The definition of the filter is firstly given in section 3.5.1, while its implementation is discussed in 4.2.2.

4.2.1. Defining the Exponential Filter

The definition of the exponential filter was already given in section 3.5.1. The description is valid for both the case with multiple species and for the Navier Stokes case. As such, it not redefined here.

4.2.2. Implementing the Exponential Filter

Given the filter definition in equation 3.5.8, the filtering of the variables can be achieved in the code simply by directly applying the filter directly. However, the filter strength has not yet been found. Furthermore, it has not been clarified how the filter should be used to achieve positivity. The implementation strategy and details are given in this section.

Regarding the filter strength, assuming that the mean mode is positive, there could be several strengths that could achieve a desired positivity. This is trivial to notice if one notices that if a filtered solution is positive for some ξ_{strength}, it is also positive for the case of $\xi_{\text{strength}} = 0$. Hence, is makes sense to expect more than one solution.

It is expected that this filter retrieves a positive solution for some set $[0,\xi_{\text{min}}]$, assuming the mean mode is positive. As such, the approach by [73] can be used. Given a set of conditions $\Gamma = \{\Gamma_0, ...\Gamma_{N_\Gamma}\}$, these can be achieved through a bracketing method, assuming that the conditions are made in a way that the assumption that there is a ξ_{min} here described holds. For the case of positive densities, it makes sense to expect that, as the densities are primitive variables.

Since several filter strengths could, then, work, it makes sense to choose the least amount of strength possible, as to impact the solution as little as possible. As such, the bracketing method can be used to find ξ_{min}, and this value can be selected to be used as the filter strength. A schematic of the application of the bracketing method, for a user defined tolerance, can be found in figure 4.1.

The filter is applied to the states using equation 3.5.8. The basis transformation matrices $\mathcal{V}$ and $\mathcal{V}^{-1}$ are constant (for a mesh element type), and are given outside of the method, thus they are not computed during the method. During setup of the simulation, based on the mesh elements, these matrices are computed and made available. Based on the current element type (2D quadrilateral, 2D simplex,...), these matrices, are, then, accessed to be used in equation 3.5.8. The diagonal matrix with the filter strength in equation 3.5.8 is computed each time the filter needs to be applied, as the filter strength is not a constant.

Regarding which variables must be checked, positivity of each thermodynamic variable is the trivial choice. A more elaborated choice can be taken from the work by Dzanic and Witherden [73], which includes density and pressure positivity (and, hence, thermodynamic variables positivity) as well as an non-decrease in entropy. Entropy rising is quite an expected phenomena, and clearly something to be expected during a shock moving in the domain. However, given the possible (and desired at times) non physical evolution in implicit methods, this method could be too stringent. Furthermore, the entropy check is applied to each element and its neighbours and not to the full domain, and certain phenomena, such as heat transfer across boundary (such as a wall) could decrease entropy in the element. In the work of Dzanic and Witherden [73], an explicit method is assumed as to assume that the entropy only has to be larger in the next time step than the previous values in the current and adjacent elements. This is due to the hyperbolic nature of the equations, where the finite wave propagation implies that

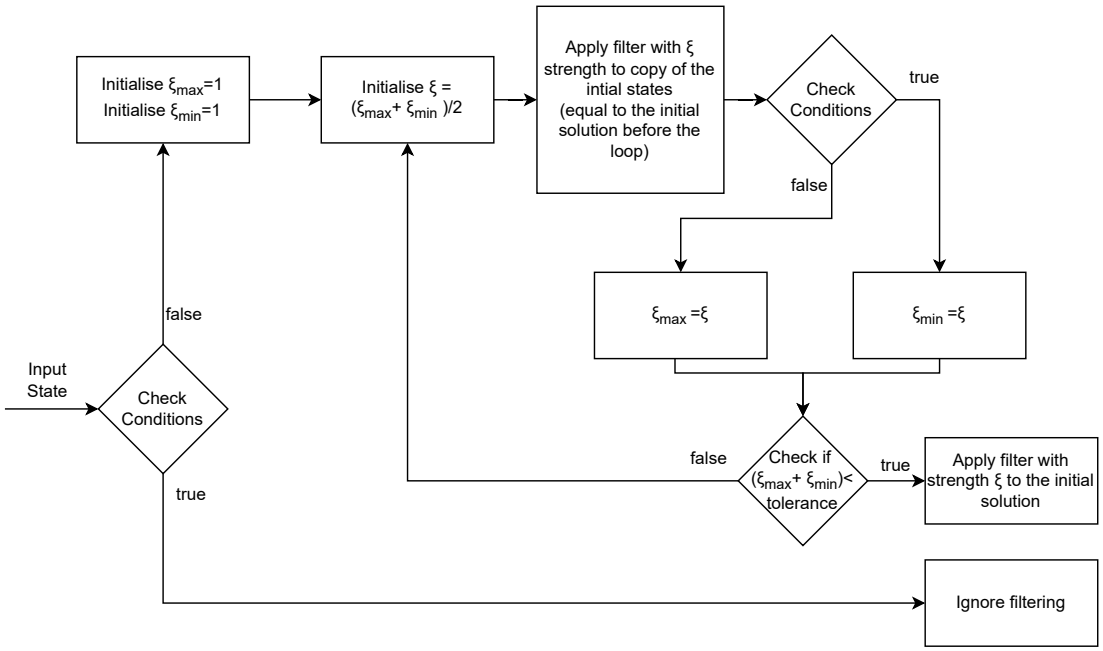


Figure 4.1: Bracketing method for finding and applying an exponential filter.

the entire domain does not have to be checked. Furthermore, the explicit methods with their strict CFL imply that the waves should not propagate more than one element over, explaining the choice of domain to be checked. Given the success of this constraint in the work of Dzanic and Witherden [73], and the clear problems associated with it, this check was introduced but kept as optional. These choices mirror the ones done for the Euler case in the previous chapter.

As such, three checks for the filter were introduced:

$$\Gamma_0 = \rho_s > 0, \text{for any species } s \tag{4.2.1}$$

$$\Gamma_1 = p > 0 \tag{4.2.2}$$

$$\Gamma_2 = s > s_{\min}, \tag{4.2.3}$$

$$\tag{4.2.4}$$

where $s_{\min}$ is the maximum entropy in the element and its adjacent elements in the previous iteration. Pressure p is given by Dalton's law – that is, the partial pressure of each species is calculated through an ideal gas law, and then summed together:

$$p = \sum_{i}^{N_s} p_i = \sum_{i}^{N_s} \rho_i R_i T, \tag{4.2.5}$$

where R_i is the ideal gas constant of species i, for the single temperature case. For two temperature, see equation 4.2.6 can possibly be taken:

$$p = \sum_{s \neq e} (\rho_s R_s T) + \rho_e R_e T_v, \tag{4.2.6}$$

where, $_e$ denotes electron, and T_v is the roto-vibrational temperature.

The entropy s is taken as:

$$s = \sum_{i}^{N_s} s_i. \tag{4.2.7}$$

However, s_i is not well defined. As such, it must be derived. This derivation is tackled in section 4.2.3.

4.2.3. Entropy derivation

The entropy of the species involved is required for the entropy check. This chapter derives the entropy for a single species. The species index is omitted here as the derivation is valid for all species. However, this entropy is not yet well defined. Assuming ideal gas, the entropy can be calculated using:

$$s = \int_{T_{\text{ref}}}^{T} \frac{c_v}{\tau} \mathrm{d}\tau + R_s \ln(\frac{1}{\rho_s}), \tag{4.2.8}$$

with

$$c_v = \frac{\mathrm{d}e}{\mathrm{d}T} = \frac{\partial e_{tr}}{\partial T_{tr}} + \frac{\partial e_v}{\partial T_v}, \tag{4.2.9}$$

in the for the two temperature case. For the three temperature case, electron temperature T_e is usually taken as well. For the current implementation with PLATO, the energy expressions for 3D are [21]:

$$e_h = \frac{5}{2}\frac{k_B}{m_s}T, \text{ where } e_h \text{ denotes the translational energy and internal energy of a heavy species } h, \text{ that is, all species except for the electron;} \tag{4.2.10}$$

$$e_e = \frac{3}{2}\frac{k_B}{m_e}T_v, \text{ where } e_e \text{ denotes the electron energy;} \tag{4.2.11}$$

$$e_{v_s} = \frac{k_B}{m_s}\frac{\Theta_s^v}{e^{\frac{\Theta_s^v}{T_v}} - 1} \text{ where } e_{v_s} \text{ denotes the vibrational energy of diatomic molecules.} \tag{4.2.12}$$

In these expression, Θ_s^v is the characteristic vibrational temperature of species $_s$. For 2D and 1D cases, the number of degrees of freedom is corrected in the e_s and e_e equations. For 2D, then, $\frac{3}{2}$ would be used instead of $\frac{5}{2}$, and $\frac{2}{2}$ instead of $\frac{3}{2}$.

Substituting equations 4.2.10, 4.2.11 and 4.2.12 into equation 4.2.8 and solving analytically using u substitution, it can be shown that:

$$s = -\frac{5}{2}\frac{k_b}{m_s}\ln(T_{tr}) - \frac{k_\mathrm{B}}{m_s}\left(\frac{e^{\frac{\Theta_v^s}{T_v}}\frac{\Theta_v^s}{T_v}}{1 - e^{\frac{\Theta_v^s}{T_v}}} + \ln(1 - e^{\frac{\Theta_v^s}{T_v}})\right) + s_{\text{ref}} - R_e\ln(\rho_s) \text{ for diatomic species} \tag{4.2.13}$$

$$s = -\frac{5}{2}\frac{k_b}{m_s}\ln(T_{tr}) + s_{\text{ref}} - R_e\ln(\rho_s) \text{ for monoatomic species} \tag{4.2.14}$$

$s = -\frac{3}{2}\frac{k_b}{m_e}\ln(T_e) + s_{\text{ref}} - R_e\ln(\rho_e)$ for the electron. T_e is usually taken as T_v for two temperature models. (4.2.15)

4.2.3.0.1 Derivation The derivation is here given. First, the required partial derivatives must be calculated:

$$\frac{\partial e_e}{\partial T_v} = \frac{3}{2}\frac{k_\text{B}}{m_e}; \tag{4.2.16}$$

$$\frac{\partial e_e}{\partial T_h} = 0; \tag{4.2.17}$$

$$\frac{\partial e_h}{\partial T_v} = \frac{k_\text{B}}{m_e}\frac{(\Theta_v^s)^2}{T_v^2}\frac{\exp\left(\frac{\Theta_v^s}{Tv}\right)}{\left(\exp\left(\frac{\Theta_v^s}{Tv}\right)-1\right)^2}; \tag{4.2.18}$$

$$\frac{\partial e_h}{\partial T} = \frac{5}{2}\frac{k_\text{B}}{m_e}. \tag{4.2.19}$$

Substituting equation 4.2.16 into equation 4.2.8, the entropy for electrons can be calculated:

$$s_e = \int_{T_{e\text{ref}}}^{T_e}\frac{3}{2}\frac{k_\text{B}}{m_e}\frac{1}{\tau}\mathrm{d}\tau - R_e\ln(\rho_e) = \frac{3}{2}\frac{k_\text{B}}{m_e}(\ln(T_e)-\ln(T_{e\text{ref}})) - R_e\ln(\rho_e) = \frac{3}{2}\frac{k_\text{B}}{m_e}\ln(T_e) - R_e\ln(\rho_e) + s_{e\text{ref}} \tag{4.2.20}$$

For a heavy species, doing the same procedure (with slight abuse of notation):

$$s = \int_{T_\text{ref}}^{T}\frac{\partial e_h}{\partial T}\frac{1}{T}\mathrm{d}T + \int_{T_{v\text{ref}}}^{T_v}\frac{\partial e_h}{\partial Tv}\frac{1}{T_v}\mathrm{d}T_v - R_s\ln(\rho_s). \tag{4.2.21}$$

Isolating the first integral:

$$\int_{T_\text{ref}}^{T}\frac{\partial e_h}{\partial T}\frac{1}{T}\mathrm{d}T = \int_{T_\text{ref}}^{T}\frac{5}{2}\frac{k_\text{B}}{m_e}\frac{1}{T}\mathrm{d}T = -\frac{5}{2}\frac{k_\text{B}}{m_s}(\ln(T_s)-\ln(T_{s\text{ref}})). \tag{4.2.22}$$

For the second integral:

$$\int_{T_{v\text{ref}}}^{Tv}\frac{\partial e_h}{\partial Tv}\frac{1}{T_v}\mathrm{d}T_v = \frac{k_\text{B}}{m_s}(\Theta_v^s)^2\int_{T_{v\text{ref}}}^{Tv}\frac{1}{T_v^3}\frac{\exp\left(\frac{\Theta_v^s}{Tv}\right)}{\left(\exp\left(\frac{\Theta_v^s}{Tv}\right)-1\right)^2}\mathrm{d}T_v. \tag{4.2.23}$$

Using $\alpha = \frac{\Theta_v^s}{Tv}$:

$$\frac{k_B}{m_s}(\Theta_v^s)^2 \int_{\alpha_{\alpha\text{ref}}}^{\alpha} \frac{\alpha^3}{(\Theta_v^s)^3} \frac{\exp(\alpha)}{\left(\exp(\alpha)-1\right)^2}(-1)\frac{\Theta_v^s}{\alpha^2}\mathrm{d}\alpha \tag{4.2.24}$$

$$= -\frac{k_B}{m_s}\int_{\alpha_{\text{ref}}}^{\alpha} \alpha \frac{\exp(\alpha)}{\left(\exp(\alpha)-1\right)^2}\mathrm{d}\alpha \tag{4.2.25}$$

$$= -\frac{k_B}{m_s}\left(\frac{e^\alpha \alpha}{1-e^\alpha} + \ln(1-e^\alpha)\right) - s_{\text{ref}_{\text{2nd integral}}} \tag{4.2.26}$$

$$= -\frac{k_B}{m_s}\left(\frac{e^{\frac{\Theta_v^s}{Tv}}\frac{\Theta_v^s}{Tv}}{1-e^{\frac{\Theta_v^s}{Tv}}} + \ln(1-e^{\frac{\Theta_v^s}{Tv}})\right) - s_{\text{ref}_{\text{2nd integral}}}. \tag{4.2.27}$$

Plugging these results, the entropy for a species is retrieved. This concludes the derivation.

4.3. Least Squares Method

The exponential filter described will not always achieve positivity. If the mean mode is negative, the filter cannot recover positivity. A method for recovering density positivity is, then, required. For such, a method was developed that would guarantee density positivity. The method here presented is applied to a single solution point inside an element.

Take some species density ρ_i at a solution point. A mass fraction could be defined as:

$$c_s = \frac{\rho_i}{\rho}, \tag{4.3.1}$$

where ρ is the density of the fluid, and assumed to be positive:

$$\rho = \sum_s^{N_s} \rho_s. \tag{4.3.2}$$

One could then imagine adding a small quantity ϵ_s to this mass fraction, as to make it positive:

$$\tilde{c}_s = c_s + \epsilon_s \Leftrightarrow \epsilon_s = (\tilde{c}_s - c_s). \tag{4.3.3}$$

Given this formulation, it would make sense to try to add as little as possible, as to distort the chemistry by small amounts. Consequentially, ϵ_s should be minimised. However, this operation should be done to all species. While it could be possible to try to make each species positive separately, a combined approach will be taken. This approach is justified by a desire for consistency and by the fact that a single minimisation problem is much simple than N_s minimisation problems. It would not make sense to change the density by so much that it deviates greatly from the original value ρ. In order to insure this, a possible constraint would be:

$$\sum_s^{N_s} c_s = \sum_s^{N_s} \tilde{c}_s = 1. \tag{4.3.4}$$

This is equivalence as to saying that the density of the fluid does not change during this method, as is named here the consistency condition.

Having established the goal of finding the positivity in a combined approached, instead of minimising

ϵ_s, one attempts to calculate:

$$\min \sum_{s}^{N_s} (\epsilon_s)^2 = \min \sum_{s}^{N_s} (\tilde{c}_s - c_s)^2, \tag{4.3.5}$$

which is a typical least squares formulation. As it can be seen, this formulation deals well with the constrain in equation 4.3.4, as it deals with exactly the same optimisation variables in a single equation fashion.

However, the species densities range in order of magnitude, and, as such, weights w_{c_s} can be added in order to try to balance the correction to the species

$$\min \sum_{s}^{N_s} (\epsilon_s)^2 = \min \sum_{s}^{N_s} \frac{(\tilde{c}_s - c_s)^2}{w_{c_s}}. \tag{4.3.6}$$

Given this introduction, the minimisation problem to acquire density positivity can be defined for each solution point:

$$\begin{aligned} &\min \sum_{s} \frac{(\tilde{c}_s - c_s)^2}{w_{c_s}} \\ &\text{s.t} \sum_{i}^{N_s} \tilde{c}_s = 1, \\ &-\tilde{c}_s \leq 0, \text{for all species } s. \end{aligned} \tag{4.3.7}$$

This constrained least squares minimisation problem can be solved using the standard procedure of Lagrangean multipliers with Karush–Kuhn–Tucker conditions:

$$L_s = \sum_{s} \frac{(\tilde{c}_s - c_s)^2}{w_{c_s}} + \sum_{s} \mu_s(\tilde{c}_s) + \lambda \sum_{s} \tilde{c}_s = 0. \tag{4.3.8}$$

To solve this Lagragian, the partial derivatives are set to 0:

$$\begin{aligned} &\frac{\partial L_s}{\partial \tilde{c}_s} = 0 \Leftrightarrow \tilde{c}_s = c_s + \frac{\lambda_s + \mu_s}{2}, \\ &\frac{\partial L_s}{\partial \mu_s} = 0, \Leftrightarrow -\tilde{c}_s = 0 \vee \mu_s = 0 \\ &\frac{\partial L_s}{\partial \lambda_s} = 0 \Leftrightarrow \sum_{j}^{N_s} \tilde{c}_{sj} = 1. \end{aligned} \tag{4.3.9}$$

The last equation in system 4.3.9 corresponds to the equality constraint. The second equation has two possible solution for each mass fraction. In case the positivity constraint is not required (the solution is already positive), the value of μ_s is set to 0; if the solution is negative, the inequality constraint becomes a positivity constraint with $\mu_s \geq 0$ and $\tilde{c}_s = 0$. That is, if the mass fraction was to be negative, it is set to zero by this constraint.

The solution to the system of equation 4.3.9 is:

$$\begin{aligned} \tilde{c}_s &= c_s + \left(\frac{1 - \sum_i^{N_s} c_s + \sum_{s,\mu_s \neq 0}^{N_0} (c_s)}{\sum_i w_{c_s}} \cdot w_{c_s} \right) = c_s + \left(\frac{1 - \sum_{s,\mu_s=0}^{N_0} (c_s)}{\sum_i w_{c_s}} \cdot w_{c_s} \right) \\ &\text{or} \\ \tilde{c}_s &= 0, \quad \text{in which case} \quad \mu_s \neq 0. \end{aligned} \tag{4.3.10}$$

This solution corresponds to a summing weighted average to the initial value in which only the mass fractions not set to zero are taken into account. Thus, it distributes the difference between the desired value of 1 and the actual sum of the mass fractions. The sum of the mass fractions will not be 1 as some of them will be set to zero to satisfy the positivity constraint.

Multiplying all by ρ, the following solution is obtained:

$$\begin{aligned} \tilde{\rho}_s &= \rho_s + \left(\frac{\rho - \sum_{s,\mu_s=0}^{N_0} (\rho_s)}{\sum_s w_{c_s}} \cdot w_{c_s} \right) \\ &\text{or} \\ \tilde{\rho}_s &= 0, \quad \text{in which case} \quad \mu_s \neq 0. \end{aligned} \tag{4.3.11}$$

This family of solutions gives necessarily a positive value as long as ρ is positive. This can trivially be seen by noting that setting one species density to ρ and all others to 0 is a possible solution.

However, this family of solutions has several possible correct solutions – that is, there are several local minima. To choose correctly, all possible solutions would need to be sampled and compared in order to find the global minima. Instead, an iterative approach for each solution point is taken:

- 1. Check for negative densities. If none found, exit algorithm.
- 2. Set the most negative density to 0.
- 3. Use the first equation in equations 4.3.11 to update the original densities (always apply this step to the original set of densities, with exception of the ones set to zero).
- 4. Go to step 1.

Regarding the weights, these can be set to ρ_i in order to keep the order of magnitude correct.

4.4. Energy Based Correction

An additional correction for temperature is here given, based on the internal energy. This choice was motivated by the fact that changing the species densities changes the influence of their formation enthalpy as well as of their temperature in the energy of the system.

As such, a method to recovered a desired temperature was developed. Changing the densities without changing the temperatures through the LS method would lead to different energy values and different gradients. Given this change, it was conjectured that these changes could be driving the instabilities. As such, a correction method was proposed. Following the same idea as the density method, quantities which depend on several species are used to estimate new values. While in the LS method the ρ was used, here the internal energy e_t and pressure p will be used to try to retrieve new estimates.

The internal energy is given as [21]:

$$e_t = \sum_{i \in \mathcal{H}} \frac{\rho_i}{\rho} e_{h_i} + \sum_{i \in \mathcal{I}} \frac{\rho_i}{\rho} e_{v_i} + \frac{\rho_e}{\rho} e_e + \sum_i \frac{\rho_i}{\rho} e_{\text{formation}}, \tag{4.4.1}$$

where $\mathcal{H}$ is the set of non electron species (denoted heavy species), and $\mathcal{I}$ is the set of species which have roto-vibrational energy. The expressions for these energies are given in equations 4.2.10,4.2.11 and 4.2.12.

Given equation 4.4.1 for internal energy and equation 4.2.6 for pressure, a system of equations can be defined:

$$\begin{cases} e_t = \sum_{i \in \mathcal{H}} \frac{\rho_i}{\rho} e_{h_i} + \sum_{i \in \mathcal{I}} \frac{\rho_i}{\rho} e_{v_i} + \frac{\rho_e}{\rho} e_e + \sum_i \frac{\rho_i}{\rho} e_{\text{formation}} \\ p = \sum_{s \neq e} (\rho_s R_s T) + \rho_e R_e T_v \end{cases} . \tag{4.4.2}$$

Given densities ρ_i, system 4.4.2 can be solved for T and T_v. For a single temperature version, either a single energy equation for T is taken, or, if breaking the energy modes into multiple parts, such as in equation 4.4.1, T is taken to be equal to T_v – that is, in thermal equilibrium, as expected from a single temperature formulation. It is important not to set T_v to zero if the energy is being broken into translational and vibrational components when using a single temperature, as this will disregard those components. The equilibrium assumption is the physical one.

To solve this system, one can write in terms of T and T_v, and note that it gives a linear equation for T and a non-linear equation for T_v. Given that T_v is physically bounded – cannot be smaller than 0, and, for our cases, is usually bellow 30000 K – its equation can be solved through simple bracketing with a user defined tolerance.

The energy and pressure taken in system of equations 4.4.2 are the ones before the LS method for densities. This is what allows the temperature to be corrected for the densities.

4.5. Full Positivity Method

Given the strategies above presented, the full method for positivity can now be defined. The method is broken into two large parts: compute the filter and apply the filter.

From a high level perspective, the high level implementation is as described in figure 4.2. The physicality module is called after the time marching and before the start of the next loop. As such, it prepares the values to start the FR method with positive values.

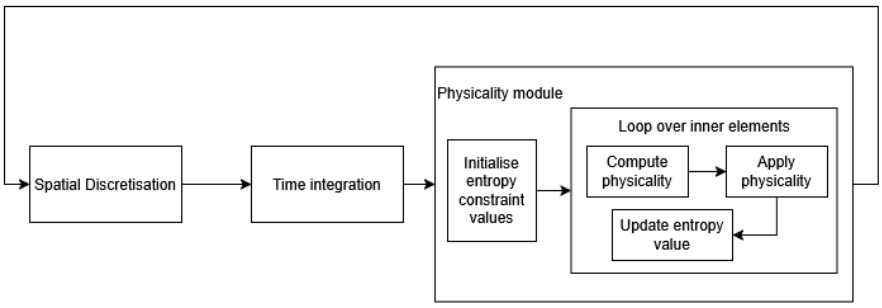


Figure 4.2: Schematic of the physicality method within the context of the FR method.

As seen in figure 4.2, the physicality module first initialises the entropy for each element. This is

required as the entropy condition checks not only the value of the current element, but also the entropy values of the adjacent elements. As such, it makes sense to calculate all values first, and then merely check them, instead of calculating the values for the adjacent cells repeatedly during the entropy check, which trivially would increase the number of computations (the same element would have its entropy calculated several times).

After this initialisation, the physicality module loops over the inner elements and applies physicality. For such, the two main parts are executed: computation of the required physicality, and its application. The computation concerns mainly the finding the strength of the exponential filter required. Given this context, the computation of the physicality and its application can now be defined. A possible – and the chosen – implementation is given in figure 4.3. This choice in particular calculates the filter strength without using the least squares method for positivity nor the energy based correction. This is important, as using them during the computation of the filter will lead to the density check always being achieved. This, then, leads to the filter not being used for that check, but merely for the others (pressure and, possibly, entropy). However, the filter is the choice found in literature that was shown to work well for the positivity problem. Furthermore, by reducing oscillations, it increases the robustness of the solution. As such, the choice here made is equivalent to choosing the smallest strength such that the exponential filter checks all constraints.

The least square method and the energy based method, then, only serve to correct the solution in case the filter fails – which is possible. As such, they are used in the apply physicality part as to prevent the failure of the physicality module.

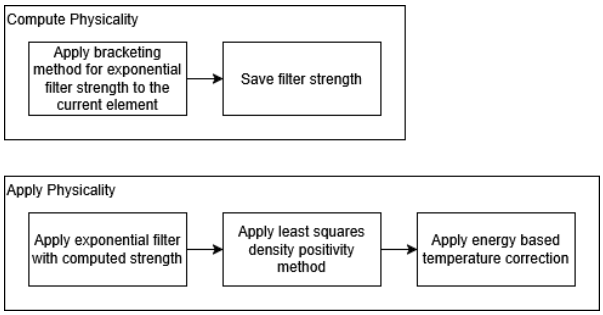


Figure 4.3: Schematic of the application of the compute physicality and apply physicality modules. Importantly, these differ in the sense that the least squares positivity and associated energy based correction are not present in the computation part. This is valid for all methods of order P1 and higher

A special case is defined for the P0 case (order zero solution, where a single solution point is used inside the element). In this case, the filter cannot be applied, as the solution is defined by a constant value. As such, only the least squares method and the energy correction are applied. In this particular case, only the density constraint is, then, imposed.

The bracketing method used during the compute physicality part can be found in figure 4.1. The bracketing method chosen was a simple bisection due to being easy to implement. A more detailed description of the apply physicality part is given in figure 4.4. In it, it is emphasised how the energy and pressure have to be saved prior to application of the least squares method for densities, in order for them to be used in the energy based correction.

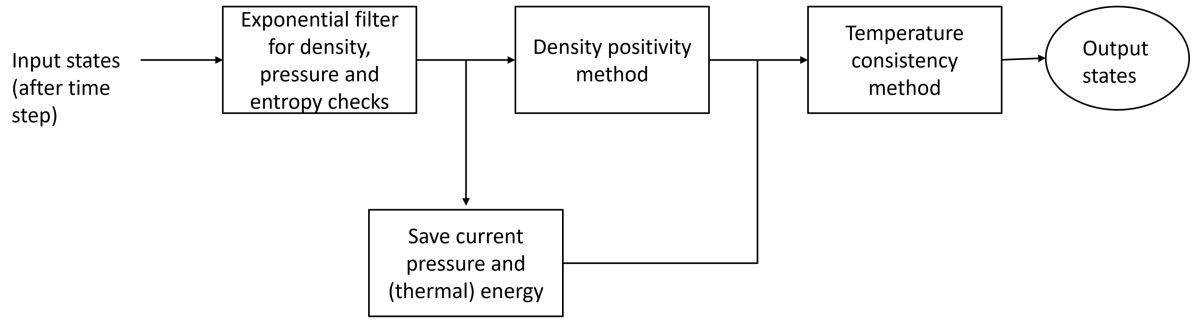


Figure 4.4: Schematic of the apply physicality part

4.6. Physicality Proof of Concept

The implementation of the physicality method was tested for positivity and consistency of the species through a small proof of concept. This proof of concept had the objective of being a succinct demonstration that the method would guarantee consistency of the mass fractions as well as positivity for a noisy solution while qualitatively maintaining properties of the initial curves.

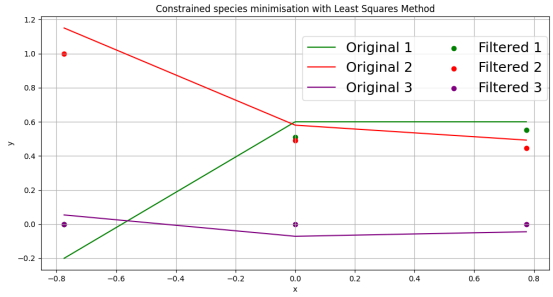
As such, an implementation in python was executed. This allowed isolating the positivity method as well as easily creating artificial data to test it. An implementation for three species and an arbitrary number of solution points in 1D was made. Given a set of mass fractions, the method is applied in order to generate a set of mass fraction within the constraints.

For this proof of concept, the energy based temperature method was omitted. This had three justifications. First, given that the proof of concept focused primarily on the positivity of the density, only this constraint was tested. Second, the model is non physical, but based on user input artificial data, and, hence, only the densities were chosen to be created, as using other values would require comparison with literature to use reasonable results (pairs of temperature density) which could skew the results. Third and foremost, at the time of creating of this proof of concept, the need for the energy based method was not evident, and, hence, such had not yet been created.

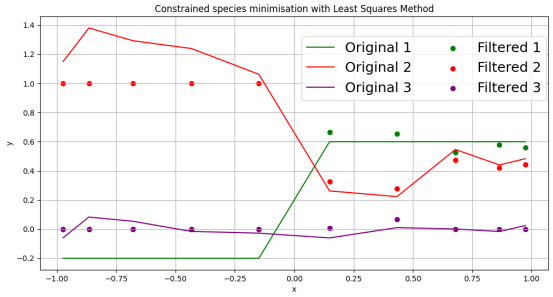
The implementation in python applies successively for artificial mass fraction first the exponential filter and then the least squares method. The results were plotted in a space-density plot, including the original and new values.

For testing, it was chosen to have a species near and bellow zero, representing a very low density species which could even be supposed to be zero. It was also chosen to have a species with a step function in space, as to see the effect across a shock when a species jumps in mass fraction due to formation during a chemical process. A third species with high values was also added, as to represent the opposing case to the low density species. Gaussian noise was added to the initial solution as to more easily represent a solution with oscillations. The seed chosen was 42.

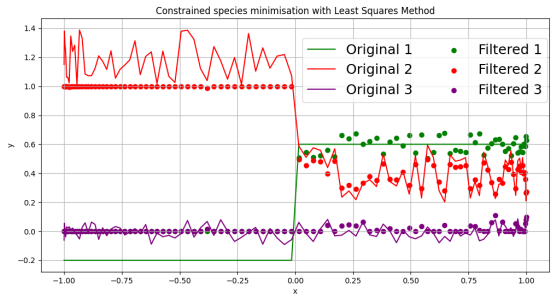
Figure 4.5a shows a case for a 1D spacial domain with 3 solution points. Species 1 has a step from a negative value to 0.6 (mass fraction values). Species 2, to compensate, decreases its value. In this example chosen, the mass fractions do not add to zero due to the noise added, and by choice. This case is harder than the starting case for the actual implementation in COOLFluiD where, by construction of the method, the initial mass fractions would add to 1. As such, this should be understood as the case where one of the species (a fourth species not here represented) has already been set to 0, and the three



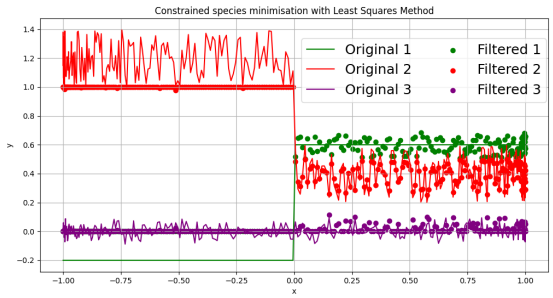
(a) Species variation over a 1D domain for 3 solution points before and after the application of the positivity method. The y axis represents the mass fractions, while the x axis represents the 1D physical domain (in arbitrary length units)



(b) Species variation over a 1D domain for 10 solution points before and after the application of the positivity method. The y axis represents the mass fractions, while the x axis represents the 1D physical domain (in arbitrary length units)



(c) Species variation over a 1D domain for 100 solution points before and after the application of the positivity method. The y axis represents the mass fractions, while the x axis represents the 1D physical domain (in arbitrary length units)



(d) Species variation over a 1D domain for 300 solution points before and after the application of the positivity method. The y axis represents the mass fractions, while the x axis represents the 1D physical domain (in arbitrary length units)

Figure 4.5: Positivity applied to synthetic data points representing three species in a 1D domain. The y axis represents the mass fractions, while the x axis represents the 1D domain in arbitrary length units.

species remaining are being corrected for. The third species oscillates around zero.

The new solutions clearly follow the outline of the original ones, but keep values between 0 and 1, add to 1 and are all positive, showing a successful implementation of the method. Following the same idea, plots with 10, 100 and 300 points were generated, corresponding respectively, to figures 4.5b, 4.5c and 4.5d.

All of these examples also show a successive trial in using the positivity method, as for all solution points tested, the values were between 0 and 1, added to 1 and were all positive.

Given that the method is applied point wise, the shape of the graph dos not influence the solution points themselves. The shape of the graph mere influences its post filtered shape. As seen in figures 4.5a, 4.5b, 4.5c and 4.5d, the resulting values seem to be near the original graph in shape, qualitatively speaking. Furthermore, the cross species influence is well seen in figure 4.5c, where species 1, despite originally a constant line, now oscillates together with species 2. This is reasonable, as the consistency constraint dictates that the species are related and should, indeed, vary together as a whole. A quantitative study regarding the shape of the graph was not made as the original species were highly distorted and this case was fabricated based on artificially creating purposely noisy data. The excessive nature of the artificial data shows well the power of the scheme, but it also renders quantitative comparisons as useful as qualitative visual analysis of the plots.

A statistical analysis of the working of the method was not done as all cases attempted returned successful results. A stochastic simulation to test a wide range of data was not attempted as the method

is mathematically proven to work point wise (for initial positive sum of the mass fractions) and would be pointless. This proof of concept quickly showed how much the graphs are distorted, as well as helping confirm the working of the method in a application standpoint. Furthermore, given its isolated implementation, it helped develop the chosen and presented implementation of the current method and, hence, its final implementation in COOLFluiD.

4.7. Physicality Implementation in COOLFluiD

The positivity method was given in section 4.5, where upon an implementation was given. However, this implementation still requires being translated into code in the code base of COOLFluiD. Following the same approach as for the filter in chapter 3, the class FilteringTQNE2D is created. It is a subclass of the BaseFiltering class (already tackled also in chapter 3). This class must allow – as it was done for FilteringEuler2D – for the calculation of the physical quantities entropy, density and pressure from the variables in the solution points, as well as define a function to check the constraints, and another function to choose the filter. In addition to this, it must allow the computation and use of the least squares method defined in section 4.3 and the energy based correction defined in section 4.4.

The physicality here presented is implemented in the same place of the code as the one presented in figure 3.3, in accordance with the integration method developed in chapter 3.

4.7.1. FilteringTQNE2D class

The FilteringTQNE2D class defines in essence the way to compute the required physical variables for the filter, the filter, the least square method, the energy correction method, and allows for the choice of filter and species being used. These functions are supported by auxiliary functions defined in the code. These characteristics may be found in schematic form in figure 4.6.

The function to choose the filter is analogous to the one for the FilteringEuler2D class, allowing the user to choose from a hardcoded variety of filters. These are presented in figure 4.7, and consist of variations on using the least squares and the exponential filter. The first option is simply the exponential filter, allowing for the use of this method with a single species. The second option is based on using the exponential filter and the least squares method with the energy based correction together for the computation and application of the filter. However, the use of the least squares method in the computation of the filter will make it so that a lower filter strength might be used, as the density positivity will be guaranteed. As such, the deformation of the solution to a positive one will be based on the least squares method heavily, rather than on the exponential filter. While this option is given, it will make heavy use of a point by point correction rather than by the filter which takes the whole element into account, and which is based on a more tested method in literature. Furthermore, it could also lead to longer computations due to driving the filter strength in the bracketing method to effectively disregard the filter for cases where only the density of the species are negative and not the pressure, making it a more elaborate and computationally expensive method than just using just the least squares while having the same effect for these cases.

The third option given is just the least squares method together with the temperature correction. This option substitutes option two in the aforementioned cases, while also being the only option that targets the P0 case in specific, since a filter cannot filter a mean mode further (which is the case for P0). This option, hence, should be the default for P0 cases from the ones given here. The compute filter option in this case is ignored, effectively forcing the program to skip that step.

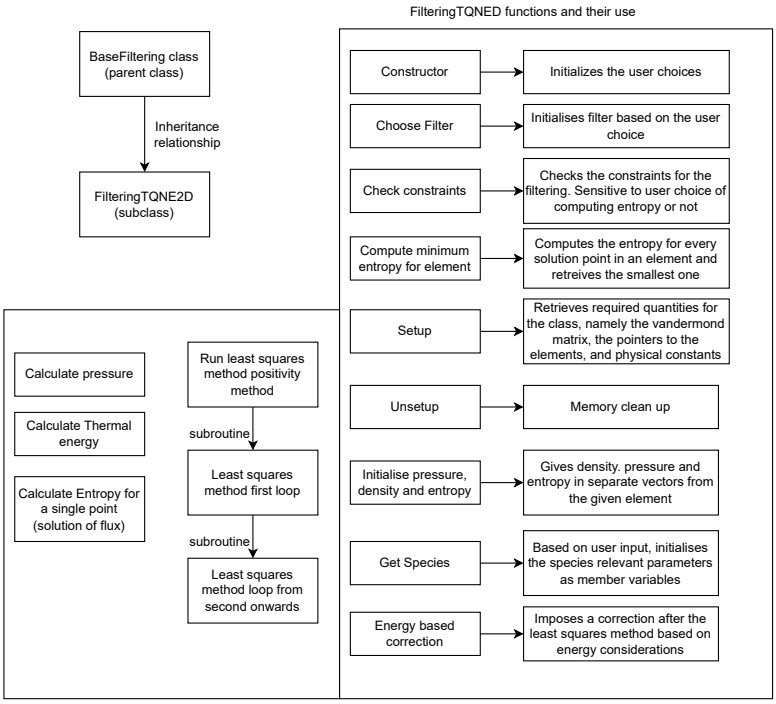


Figure 4.6: FilteringTQNED2D class.

The fourth option comes to remedy the second one by using only the exponential filter during the computation of the filter while using the filter and the least squares method with the energy based correction for the application of the case. This choice is a reiteration of the defended approach in figure 4.3, and the elected one for P1 or higher cases.

The choice of the species is given in figure 4.8. The user can choose from three different options currently, which are relevant to the tested cases, and common choices in hypersonics within the Earth's atmosphere. Both the 5 species and 11 species models for air (the latter including ions and electrons) are included given their importance for simulations within the the Earth's atmosphere. The 3 species argon – included also in PLATO and corresponding to argon, ionised argon Ar^+ and electron – is included for test case related purposes. After the choice, the constants related to the method described in this chapter

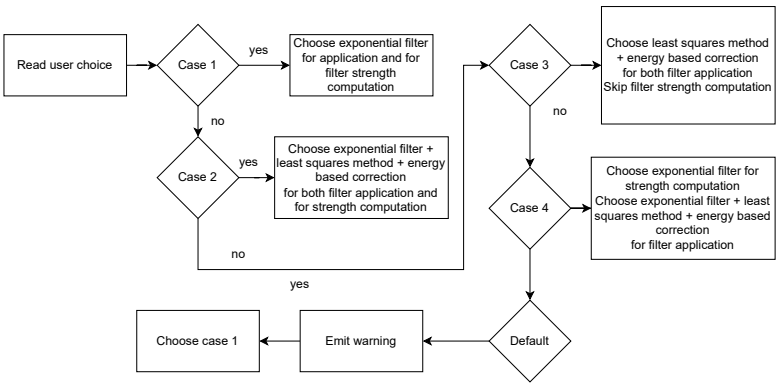


Figure 4.7: Filter choice function for the FilteringTQNEQ2D class.

are loaded, corresponding to the number of species, the mass and formation energies of the species, and which species have roto-vibrational energy and which do not. The information about the inclusion of electrons in the species is also included given the required used for the energy based correction.

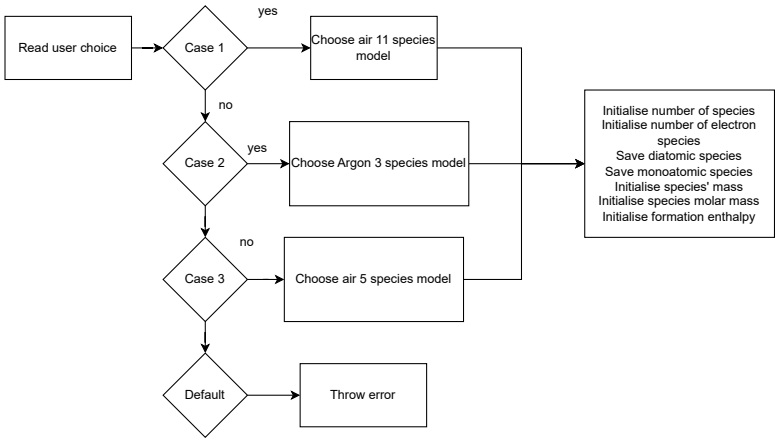


Figure 4.8: Species choice function for the FilteringTQNEQ2D class.

4.7.2. Least Squares implementation

The least squares method was included in the FilteringTQNE2D class. As emphasised in the figure 4.6, the method is separated into three functions. The main loop is given in figure 4.9. It consists of looping over all the solution points and applying the method described in section 4.3 to each one. It initialises a variable to track which species are being used in the weighted average step before the application to each solution point. This variable will allow to differentiate between the species for which $\mu_i = 0$ and which ones have $\mu_i \neq 0$.

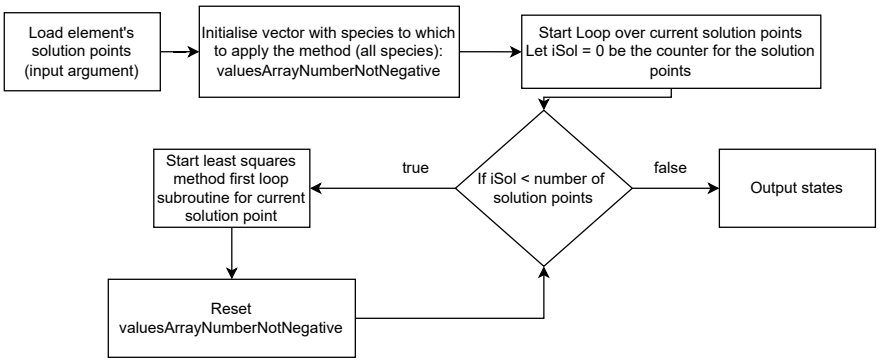


Figure 4.9: Main loop for the least squares method for density positivity.

The first subroutine of the method is given in figure 4.10. This routine calculates the reference density to be used in the weighted average step and checks whether the method is required. If the method is required, it set the most negative density to a minimum density value, stores that species has $\mu_i \neq 0$ (that is, it stores that the weighted average is not applied to this species), and then proceeds to the next loop of the method.

The next loop is seen in figure 4.11, and consists of applying the weighted average step to the species which require it and testing if there are no negative species. If there are, it stores that species has $\mu_i \neq 0$,

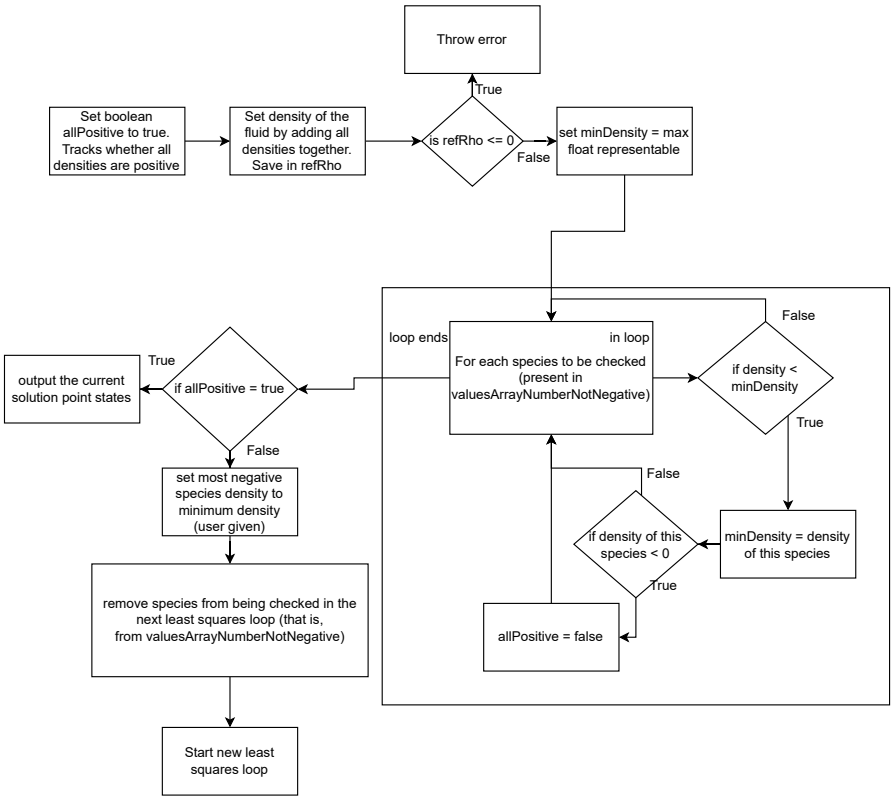


Figure 4.10: First subroutine for the least squares method for density positivity.

sets it to a minimum value and proceeds to run itself again. The weighted average is always applied to the original density values, and not the ones acquired by the weighted average in between loops. Once the a solution with all positive densities is found, it exits the function.

4.7.3. Energy based correction Implementation

The implementation of the energy based correction can be found in figure 4.12. It consists in a bracketing method to find the T_v temperature, followed by solving a linear equation for T, in consistency with equation 4.4.2. If there is a single temperature, then T_v and T are in equilibrium, and the method can be simplified to only solving the linear equation. This justification was given in section 4.4. This allows the extension of the method to the single temperature case as well, thereby increasing the range of its applicability through a simple check.

4.8. Physical Quantities Related Functions

Auxiliary functions had to be created for the calculations of the physical quantities required for the methods. Functions for the calculation of pressure and density for single solution points were employed. These correspond to the implementation of the equation 4.2.6 for pressure and equations 4.2.13, 4.2.14 and 4.2.15 for entropy.

The function to calculate the minimum entropy for an element follows the same implementation that for the FilteringEuler2D class. The check constraints also follows the same implementation. Both make heavy use of the functions described in the previous paragraph.

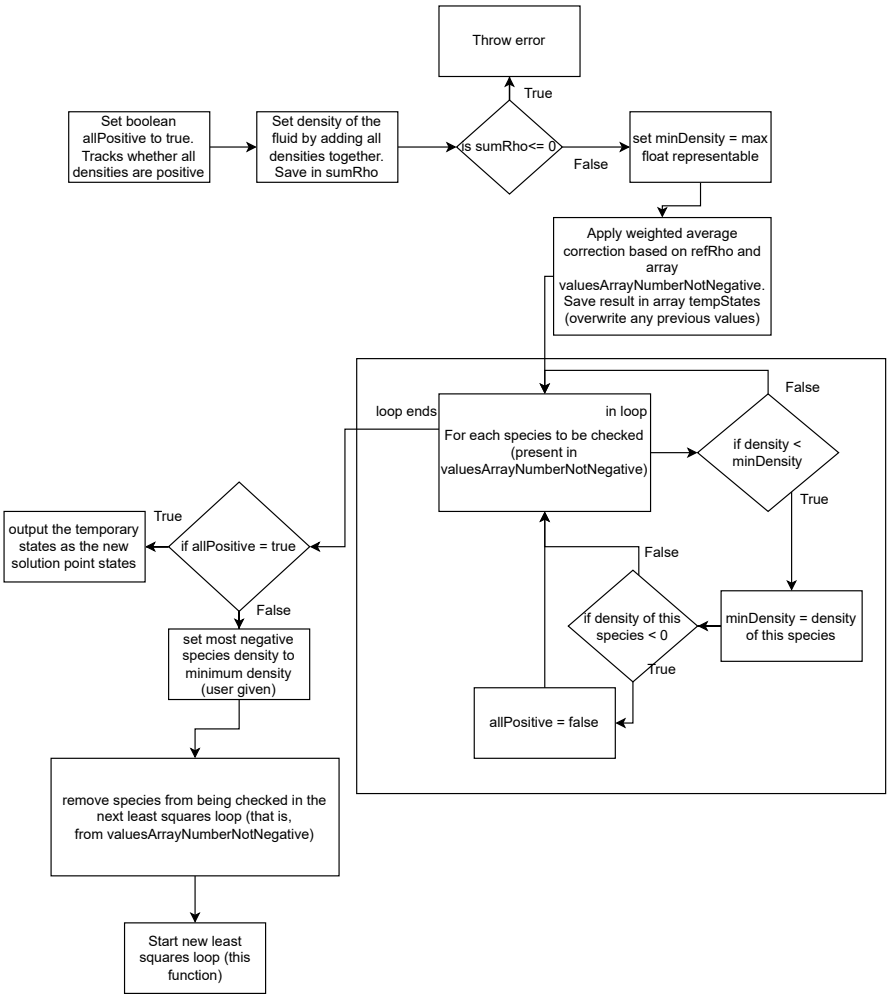


Figure 4.11: Second subroutine for the least squares method for density positivity.

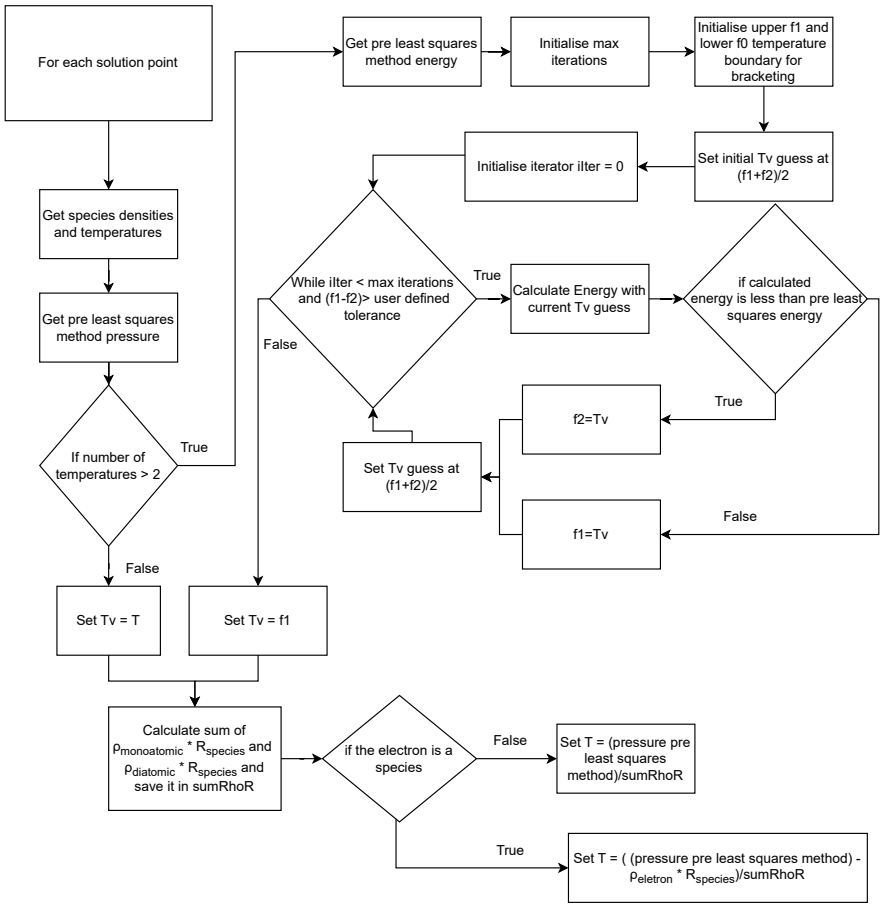


Figure 4.12: Temperature correction method based on energy routine.

5

Application of the Positivity Method

To test the new physicality method, an ensemble of cases were tested based on available literature. These tests concern cases in which thermal and/or chemical nonequilibrium is found, and which can be reproduced using 2D simulations with COOLFluiD and PLATO. The cases were also chosen to be those in which the nonequilibrium conditions are driven by a detached shock wave. The main goal of this section is to verify that positivity is maintained and that the positivity method works. Further remarks are made on the validity of the simulations and their accuracy, as it is important not to compromise the simulation solution. The first focus is put on P0 simulations, as these compute with the most ease – as their low order prevents higher order driven instabilities – while making use of a positivity method.

The first case described is the high enthalpy shock tunnel (HEG) cylinder [27], as it includes both thermal and chemical non equilibrium, together with both experimental measurements and an array of validated simulations [27]. The second on corresponds to the Kranc case

5.1. Research Questions

Given the new positivity method developed in the last chapter, the questions presented in section 3.2 can be recast for the thermochemical nonequilibrium cases.

Regarding the method, the main research question is:
Can the least squares method guarantee positivity of the densities during FR simulations for the thermochemical nonequilibrium case?

Regarding the the extension of the filter to the multiple species and multitemperature cases, the following question applies:
Can the exponential filter together with the least squares method guarantee positivity of the densities and temperatures during FR simulations for the thermochemical nonequilibrium case?

5.2. Method

In order to address the research questions, two test cases were selected: the HEG and the Kranc case. The HEG case is introduced in section 5.3, while the Kranc case is introduced in 5.4. The HEG case concerns a thermochemical nonequilibrium case with 11 species, while the Kranc case focuses on chemical nonequilibrium with 3 species. This choice of cases manages to tackle a two different species sets and well as two different nonequilibrium conditions. Further motivations for these choices are given in sections 5.3 and 5.4.

In order to verify that the positivity works, the cases are first run in order P0. This allows the testing of the LS method for the most robust FR order option. It also isolates it for the filter method by necessity, since the filter cannot act on a P0 solution, allowing the isolating test of this method. Furthermore, since no P0 simulations with PLATO in COOLFluiD were possible, a successful run would entail already an improvement on the previous state of the program.

A successful run will also provide a good initial solution for subsequent P1 or higher runs, where the shock wave will be much closer to the converged positions, and the distribution of the variables over the domain will also be closer to the converged ones in P1 or higher. Starting from such an initial solution in P1 or higher will, then, help skip the transient motion of the shock wave from the body surface to position which can be more unstable than just running from a previous converged P0 solution.

A P0 solution, then, will enable tackling the positivity question for the LS method while giving a solution for P1 or higher simulations. Given that FR code is not made to run thin FV like meshes, the mesh chosen will be purposely coarse for P0, with the goal of testing it further at higher orders.

All cases are to be run to convergence residual wise, which is taken as a relative norm of the residuals bellow -6. The CFL is heuristically taken with a low initial start in order to not destabilize the solution. The setups for the P0 simulations may be further found and motivated in sections 5.3 and 5.4.

As a follow up from the P0 simulations, P1 simulations are taken. These are introduced and addressed in section 5.5. The P1 cases are restarted from P0, as to make use of an initial solution that is closer to the converged solution. The cases follow the higher order setup used for the Wedge case, with the required alterations to work with the new physical model. The filter with the LS method is the focus, and, thus, employed in every run. The cases are run with a range of AV strength, given that it is a tunable parameter which can impact both the run and the end solution. The S0 and κ for the AV are fixed to specific values based on the considerations in the work of Vandenhoeck [2].

The validation of the runs is done against results in literature, all presented in the respective sections of the cases. The HEG case is validated against wall quantities in Knight, Longo, Drikakis, *et al.* [27], and also compared to the converged and validated simulated solutions of Sharma, Giangaspero, Munafò, *et al.* [6], while the Kranc case is validated based on the shock standoff distance measured in Kranc, Yuen, and Cumbel [82]. The P0 simulations are not expected to be fully validated, as already addressed in this section, due to the mesh use, but are expected to be qualitatively similar. P1 simulations are expected to be closer to the validation values.

The specifics of the method for each run may be found in section 5.3 for the HEG P0 case, in section 5.4 for the Kranc P0 case, and in section 5.5 for the P1 cases.

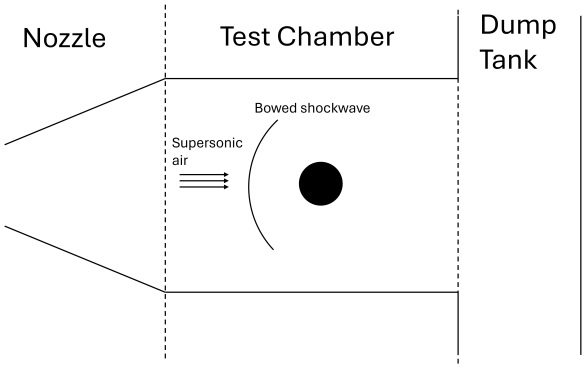


Figure 5.1: Simplified test setup schematic (not to scale) for the HEG cylinder, based on the setup presented by Karl, Schramm, and Hannemann [79]. Air exits the nozzle at supersonic speed and hits the cylinder, leading to the formation of the shock wave. Not depicted, pressure traducers on the surface of the cylinder measure pressure. Also not depicted, the density in the shock layer is measured optically. The rest of the shock tunnel is not depicted as it is not relevant, but can be found in [79].

5.3. Application to the High Enthalpy Cylinder

The case for an high enthalpy shock tunnel (HEG) cylinder present in the work of Knight, Longo, Drikakis, *et al.* [27] was considered – in particular, the test conditions denoted in [27] as I. This chapter describes the setup of the simulation, the simulation and its results when compared to literature.

A P0 simulation will be the first target of the studies. This is the most stable of the possible orders, and does not make use of the exponential filter, isolating the LS based positivity method.

Importantly, the P0 FR case with PLATO, prior to the implementation of the positivity method, could run 0 iterations and not more. This previous result was run with the same setup to be described – to the author's knowledge – but with the previous in use positivity. As such, the success of a P0 simulation will be considered a step forward in the use of PLATO with FR in COOLFluiD.

5.3.1. The Physical Case

The HEG cylinder case regards an experiment done at high enthalpy shock tunnel of the German Aerospace Center (DLR) [27], [79] and which was studied numerically by Knight, Longo, Drikakis, *et al.* [27]. An introduction to this case and an argument for its use in validation procedures has already been given in section 2.6.1.

A cylinder model with 45 mm radius and a 380 mm span was aligned transversally with the nozzle of the high enthalpy shock tunnel. The test conditions here considered are found in table 5.1, and correspond to the conditions I in Knight, Longo, Drikakis, *et al.* [27]. The working fluid was air. A detached shock wave occurred in front of the cylinder, behind which non-equilibrium chemical relaxation processes took place, leading to alteration of the thermodynamic properties behind the shock wave and, as a consequence, of the shock wave stand-off distance [27]. A schematic of this setup can be found in figure 5.1.

The large stand-off distance allowed experimental density measurements with optical techniques. Surface pressure measurements were enabled with traducers mounted on the cylinders [27]. This enabled this case to be used for validation for CFD codes with chemistry, according to Knight, Longo, Drikakis, *et al.* [27].

Assuming that the test bed did not influence the test, the size dimensions of the cylinder together

Table 5.1: Test conditions for the cylinder, adapted from [27].

Quantity	Value	Quantity	Value
H_0 (MJ/kg)	22.4	$Y[N_2]_\infty$	0.7543
P_0 (MPa)	35.0	$Y[O_2]_\infty$	0.00713
T_0 (K)	9200	$Y[NO]_\infty$	0.01026
U_∞ (m/s)	5956	$Y[N]_\infty$	$6.5 \cdot 10^{-7}$
p_∞ (Pa)	476	$Y[O]_\infty$	0.2283
ρ_∞ (kg/m^3)	$1.547 \cdot 10^{-3}$		
M_∞	8.98		

with the freestream data given in table 5.1 are enough to set up a 2D numerical fluid simulation with chemistry.

5.3.2. Simulation setup

This section concerns the setup of the simulation using an FR scheme in COOLFluiD for spatial discretisation, together with a chemical source term provided by PLATO. The details of this setup are given here. These concern the mesh, the boundary and initial conditions and the case parameters used.

5.3.2.1. HEG Cylinder Mesh

A mesh in Gmsh [85] was created. The mesh can be found in figure 5.3, and its geometry can be found in figure 5.2, while the physical regions can be found in figure 5.4.

The initial mesh consists of 70 nodes along the wall of the cylinder, together with 70 opposite ones along the inlet. On the outlet, 110 nodes are put on both sides – that is, above and bellow the cylinder. Given the supersonic nature of the flow at the outlet, only the front half of the cylinder is simulated, as it is sufficient to capture the shock wave and the surface temperature and pressure on the front part of the cylinder. This approach is in accordance with the mesh choices used by Knight, Longo, Drikakis, *et al.* [27].

While Gmsh does not work with physical units, the nodes were chosen as if the unit of distance were meters. The progression from the cylinder up was chosen as to keep the the first node at circa 10^{-6} distance (implicitly, meters) from the cylinder.

The mesh was generated with Q2 elements, as these were found to be needed to obtain accurate results with high-order simulations [2]. While the first case of interest is P0, a mesh for higher order allows the extension of the same case for higher orders with ease without having to change the mesh.

This mesh might appear coarse, namely in the shock region, when compared to a mesh for FV simulations (see [6] for an example). However, it is expected that, at high enough orders, this mesh should be sufficient. This is based on the meshes presented in [2] for the hypersonic cylinder. It is evident, from those meshes – and from the results for the simulations by Vandenhoeck [2] – that one of the advantages present in FR methods is not needing to refine the mesh in the shock wave position with as much care as in FV simulations.

5.3.3. P0 HEG Cylinder Simulation Case Setup

The setup of the simulation for the HEG cylinder with order P0 is described here. This consists in defining initial and boundary conditions, as well as the spatial and time discretisation and required methods. COOLFluiD specific inputs are required in the case file, but will be omitted here, as they are

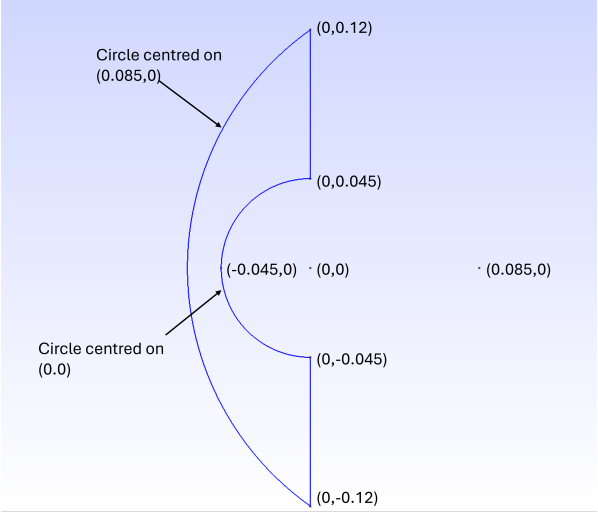


Figure 5.2: Geometry generated in Gmsh for HEG case.

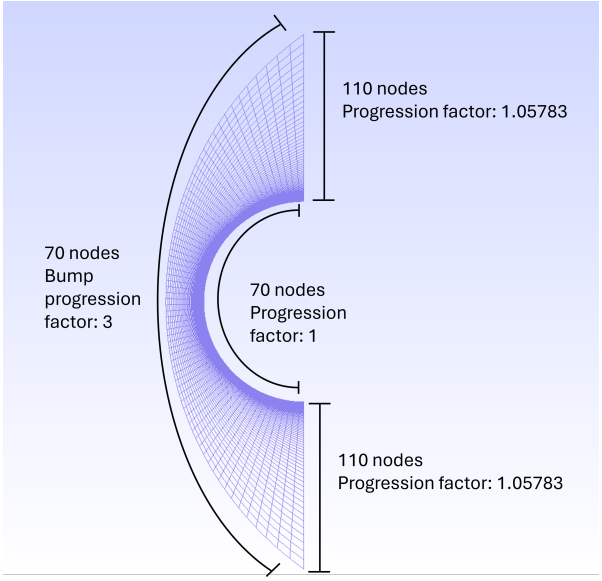


Figure 5.3: Mesh generated in Gmsh for HEG case.

not required for the understanding of the simulation.

A pre-setup step is required: that of changing the Gmsh mesh file to a CFmesh file (readable by COOLFluiD) with the desired order for the flux reconstruction. It is at this step that the P0 order is enforced. For higher order, this change of mesh format also includes an upgrade in order to PN, where N is the order. This is done in COOLFluiD through the use of MeshUpgrade builder. Thus, the order of the FR method is in the CFmesh file and is not declared as a parameter in the case file. The mesh used for this case was the one described in section 5.3.2.1.

The setup choices for this case can be found mainly in table 5.2, with the boundary conditions being present in table 5.3, and the initial conditions in table 5.4.

The physical model chosen corresponds to the Navier Stokes 2D Non Equilibrium with a chemical source term, in the form already presented and discussed in section 2.2.2. The chemical source terms in

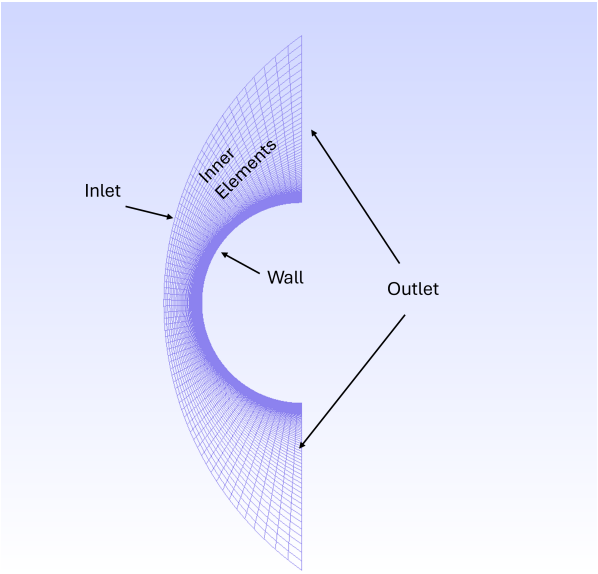


Figure 5.4: Physical regions generated in Gmsh for HEG case.

the model are given by PLATO [21]. Details on the coupling of Plato with COOLFluiD may be found in the work of Sharma, Giangaspero, Munafò, *et al.* [6], as already discussed in section 2.2.2. This physical model is run with a two temperature formulation: Temperature: the translational-rotational (T) and vibrational-electron-electronic (T_v), allowing for thermal nonequilibrium. This formulation was already used by Sharma, Giangaspero, Munafò, *et al.* [6] for this case in a FV simulation, where it was found that thermal non-equilibrium was quite evident, namely near the shock wave, justifying its use here. Details regarding the choice of these two temperatures in a standard two temperature formulation may be found in the work of Gupta, Yos, Thompson, *et al.* [8].

The Riemann Solver used was AUSMPlusFluxMS2D, available in COOLFluiD. The family of AUSM+ Riemann solvers are a flux splitting method commonly used in hypersonics, tailored to not deform the shock wave (avoid the carbuncle effect) and resolving the large range of speeds found in these cases (from the hypersonic flow to the subsonic one) [86], [87],. The choice of A12 was based on previous experience, and should not affect the converged result.

The correction function for the flux reconstruction step is the VCJH correction function [54], as they are easily extended from 1D to 2D quadrilateral elements, and have been used previously in FR simulations with success [2]. The constant was heuristically chosen as 0.33.

The physicality method chosen was the least squares method for density with the energy based correction which was derived and described in chapter 4. The exponential filter – also described in that chapter – is not used as a P0 simulation has only one solution point per element and, consequentially, the mean mode is already the value at that solution – that is, elements cannot be filtered further in P0 simulations. The only checks done during the positivity method is that of positive density, which is part of the derivation of the least squares method. Entropy and pressure checks are not executed.

The forward Euler integration was chosen due to its fast computation time. Despite the expected stiffness of the source term – given that certain chemical reactions could occur at a much faster rate than that of the flow motion – which could hamper the use of the forward Euler, it was found here that, at least for this P0 simulation, it was appropriate as it allowed the use of CFL in the range of the decimals.

The initial conditions were the same for all elements and are acquired from table 5.1 – that is, they are set to the free stream values. The boundary conditions for the supersonic inlet were also acquired from table 5.1, which is consistent with these being the free stream values. A supersonic outlet was chosen in agreement with the simulations by Knight, Longo, Drikakis, *et al.* [27], as the flow is expected to be supersonic at the outlet. The wall was chosen to be isothermal agreement with the simulations by Knight, Longo, Drikakis, *et al.* [27]. This is in agreement with it given the worse case scenario for heat transfer at the surface, as well as the speed at which the shock wave develops and leads to a steady state. The main alternative – the adiabatic wall – would be unphysical, leading to much higher temperatures near the surface, and not heat transfer to the surface. Thus, such option would only have merit in a restart method, where starting with adiabatic wall conditions would make it easier to advance the simulation (due to smaller gradients), and not for obtaining the final result. The no slip condition on the cylinder corresponds to the standard wall condition for a viscous flow.

The update variables concern the variables stored during the simulation for each solution point. These were chosen to be the primitive variables regarding the densities, the velocities and the temperatures. Such allows to derive all the other quantities of interest for the simulation, in particular the conserved variables. For these thermochemical simulations with PLATO, the primitive variables must be used in order to derive the other quantities, namely the ones given by PLATO, which require the temperatures. These would include the viscosity and diffusivity coefficients, as well as the internal energy and vibrational energy. Since these energies are given by PLATO, there is no explicit formula to invert the calculation and get the primitive variables from the conserved ones, unlike for the Euler formulation used for the Wedge case, where the ideal gas relationships could be readily used.

The solution variables concern the ones used for the formulation of the discretised equation that is forwarded in time. These concern the conserved variables for thermochemical reacting case, much like for the Wedge case. This information is required in order to process the adequate variable transformation inside COOLFluiD. Both of these choices are considered standard for the setup of a thermochemical nonequilibrium simulation with COOLFluiD to the author's knowledge.

Table 5.2: Setup for P0 HEG cylinder case (excluding boundary and initial conditions).

Parameter	Choice
Chemical model	PLATO
Physical System	Navier Stokes 2D Non Equilibrium with chemical source term
Spatial Method	Flux Reconstuction
Temperature Model	Two Temperature: translational-rotational (T) and vibrational-electron-electronic (T_v)
Riemann Solver	AUSMPlusFluxMS2D
AUSMPlusFluxMS2D A12	3
Correction Function FR	VCJH
VCJH constant	0.33
Time Integration	Forward Euler
CFL	Interactive (from 0.1 to 1)
Update Variables	Species densities, velocity in x, velocity in y, T T_v
Solution Variables	Conserved variables
Physicality	Least squares density correction with energy based correction
Finalisation	StdFinalize

Table 5.3: Boundary conditions of the simulation setup for P0 HEG cylinder case.

Parameter	Choice
Inlet BC	Supersonic Inlet
Inlet u BC (m/s)	5956
Inlet v BC (m/s)	0
Inlet T BC (K)	901
Inlet T_v BC (K)	901
Inlet ρ_{e^-} BC (kg/m^3)	0
Inlet ρ_{N_2} BC (kg/m^3)	$0.7543 \cdot 1.547 \cdot 10^{-3}$
Inlet ρ_{O_2} BC (kg/m^3)	$0.00713 \cdot 1.547 \cdot 10^{-3}$
Inlet ρ_{NO} BC (kg/m^3)	$0.0.01026 \cdot 1.547 \cdot 10^{-3}$
Inlet ρ_{N} BC (kg/m^3)	$6.5 \cdot 10^{-7} \cdot 1.547 \cdot 10^{-3}$
Inlet ρ_{O} BC (kg/m^3)	$0.2283 \cdot 1.547 \cdot 10^{-3}$
Inlet $\rho_{N_2^+}$ BC (kg/m^3)	0
Inlet $\rho_{O_2^+}$ BC (kg/m^3)	0
Inlet ρ_{NO^+} BC (kg/m^3)	0
Inlet ρ_{N^+} BC (kg/m^3)	0
Inlet ρ_{O^+} BC (kg/m^3)	0
Outlet BC	Supersonic Outlet
Wall BC	Isothermal and No Slip
Wall T BC (K)	300
Wall T_v BC (K)	300

5.3.4. Results of the P0 HEG Cylinder Simulation

Given the mesh described in section 5.3.2.1 and the setup given in section 5.3.3, a simulation in COOLFluiD was run. The simulation was restarted in order to further decrease the residuals after it had stopped once due to convergence criteria, as the higher residues had not yet converged. The residual plots and the relevant plots regarding the primitive variables are given.

Two different restarts were executed to fully converge the solution. The two last simulations were ran based on maximum number of iterations. The residuals of the simulation are found in figures 5.5, 5.6, and 5.7. The order of the residuals is: density equations, followed by momentum in x equation (Res[11]), momentum in y(Res[12]), energy equation (Res[13]) and T_v conservation equation (Res[14]). The densities are in the following order in the residuals (0 through 10): electron, N,O,N_2,NO,O_2,N_2^+,NO^+,O_2^2, O^2. A transient period is always found at the beginning of each restart in the residuals where they jump to a much higher value before descending rapidly. This is an expected restart behaviour. The residuals

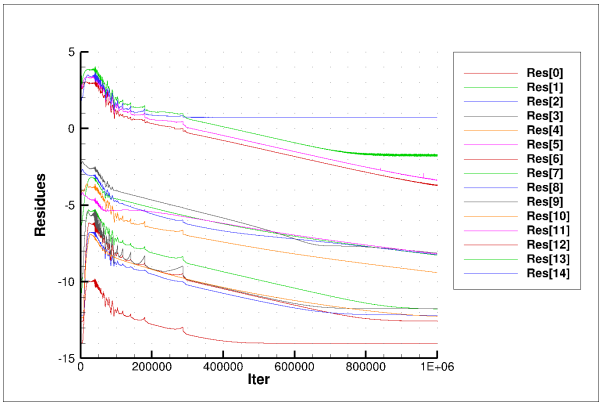


Figure 5.5: Residues of the first simulation for the P0 HEG cylinder

Table 5.4: Initial conditions of the simulation setup for P0 HEG cylinder case.

Initial Condition	Choice
T (K)	901
T_v (K)	901
u (m/s)	5956
v (m/s)	0
ρ_{e^-} (kg/m^3)	0
ρ_{N_2} (kg/m^3)	$0.7543 \cdot 1.547 \cdot 10^{-3}$
ρ_{O_2} (kg/m^3)	$0.00713 \cdot 1.547 \cdot 10^{-3}$
ρ_{NO} (kg/m^3)	$0.0.01026 \cdot 1.547 \cdot 10^{-3}$
ρ_N (kg/m^3)	$6.5 \cdot 10^{-7} \cdot 1.547 \cdot 10^{-3}$
ρ_O (kg/m^3)	$0.2283 \cdot 1.547 \cdot 10^{-3}$
$\rho_{N_2^+}$ (kg/m^3)	0
$\rho_{O_2^+}$ (kg/m^3)	0
ρ_{NO^+} (kg/m^3)	0
ρ_{N^+} (kg/m^3)	0
ρ_{O^+} (kg/m^3)	0

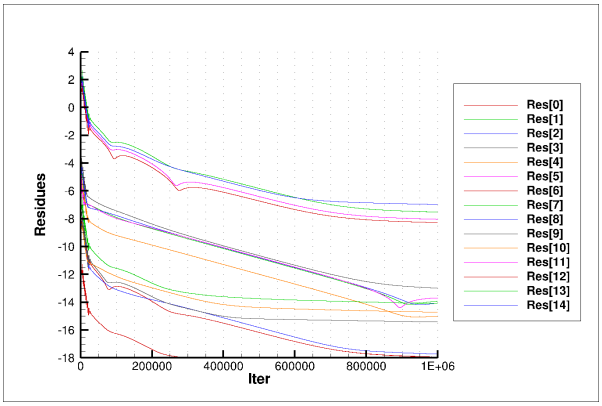


Figure 5.6: Residues of the first restart for the P0 HEG cylinder

are found to have converged in figure 5.7. Over the course of 2 million iterations, barely any change was found. The residuals also have small values, bellow -7 (log scale).

Primitive variables were plotted over the space. Figure 5.8 concerns the temperatures acquired, while figure 5.9 concerns the electron and NO^+ densities. Figure 5.10 shows the horizontal velocity. All figures show a clear detached shock wave where the quantities plotted follow a drastic jump. Figure 5.10 shows a jump from supersonic to subsonic flow across the normal shock, as expected. The shock wave is found at circa x = 0.065 m along the stagnation line. Given that the wall, on the stagnation line, is at x = 0.045 m, this leads to a shock stand off distance of circa 0.011 m. A rise of temperature of several thousand Kelvin is found in figure 5.8 – namely to circa 12 000 K for T and 8.3 K for T_v. This difference between the two temperatures shows a thermal non-equilibrium. These two temperatures lower the temperature no near wall temperature (300 K) near the wall, as expected of an isothermal wall condition. Chemical effects are clearly seen in figure 5.9, as the formation of electrons and of ions is seen due after the shock wave, with greater production in the normal part of the shock wave, where it is strongest, and where a higher temperature is reached. The formation of ions shows the clear necessity of including chemical models with ionisation in the model. The density of NO^+ is in the order of 10^{-5} kg/m^3, which is a non negligible value. The thermochemical nonequilibrium effects, then, were clearly present in this simulation.

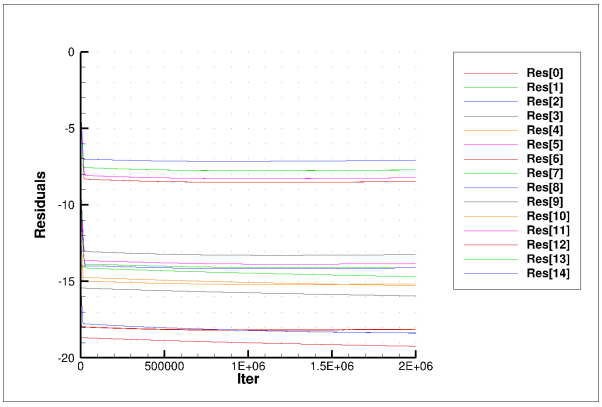
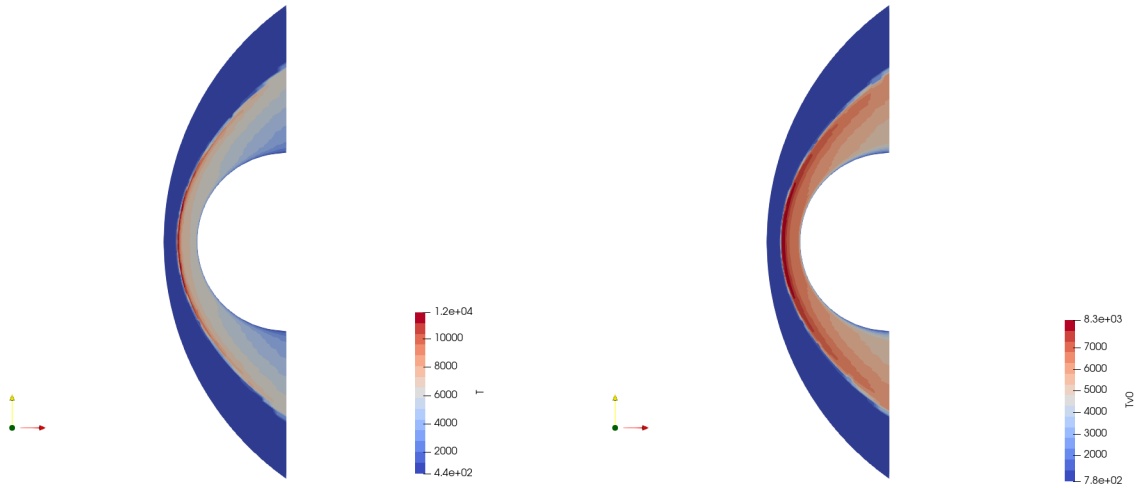


Figure 5.7: Residues of the second restart for the P0 HEG cylinder



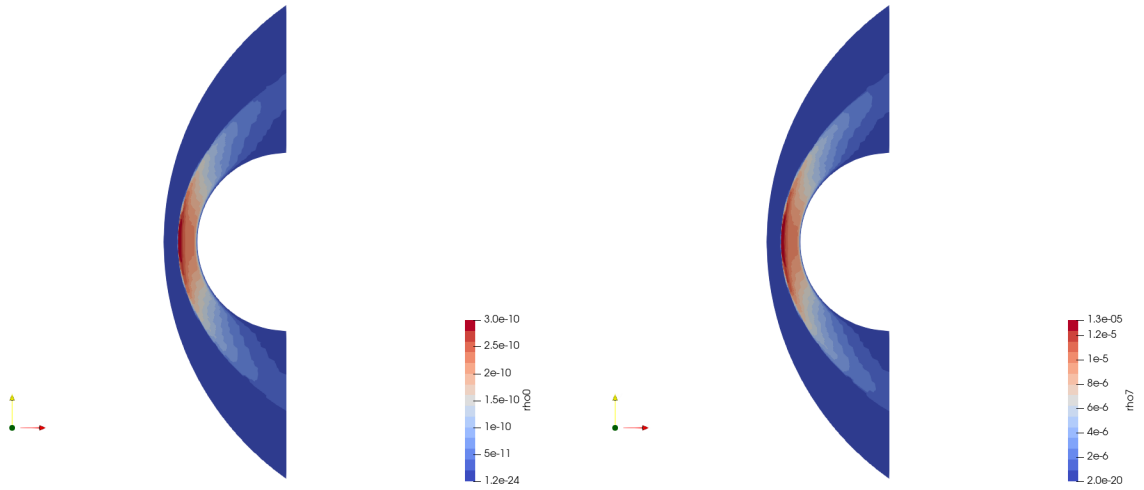
(a) HEG cylinder P0 simulation: T in K

(b) HEG cylinder P0 simulation: T_v in K

Figure 5.8: HEG cylinder P0 simulation temperatures. A clear difference between the range of temperatures found near the shock wave.

The wall quantities were plotted in figure 5.11. The pressure plot shows a curve with a maximum of 53154.5 Pa near the centre, with decreasing pressure towards the edges. This range of pressure and curve are close to the experimental results present in the validation present in the work of Knight, Longo, Drikakis, *et al.* [27]. The heat flux, however, shows a jagged curved with maximum above 12 MW, which overshoots the values presented in Knight, Longo, Drikakis, *et al.* [27] by over 4 MW. The lack of a smooth curve might imply a low resolution tangent to the surface. The wrong estimation is to be expected, since the method, being low order with a coarse mesh likely has the shock wave unconverged, affecting the entire subsonic region. With these characteristics identified, it is possible to affirm that COOLFluiD can now, in a limited sense, run P0 cases with thermochemical nonequilibrium with positive quantities. While the new method targeted mainly species densities, this simulations had both the densities and temperatures positive.

Given the lack of agreement for the wall fluxes as well as the still not well defined enough shock wave shape and position, this case cannot be argued to be fully converged, despite showing the aforementioned increase in capabilities of the code. Thus, this case requires further studies in P1 or higher. This solution is expected, as aforementioned during the explanation for the rationale for the mesh, to be used as new initial condition for a higher order simulation, as a way to avoid the transient section of the shock wave



(a) HEG cylinder P0 simulation: electron density in kg/m^3 **(b)** HEG cylinder P0 simulation: NO^+ density in kg/m^3

Figure 5.9: HEG cylinder P0 simulation: density of electrons and of NO^+. Ionisation is clearly present after the shock wave.

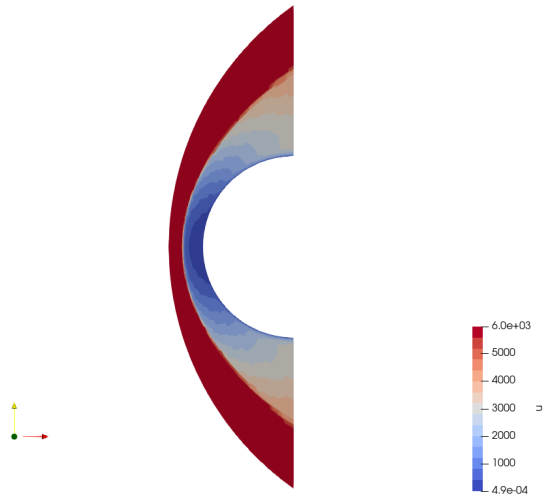
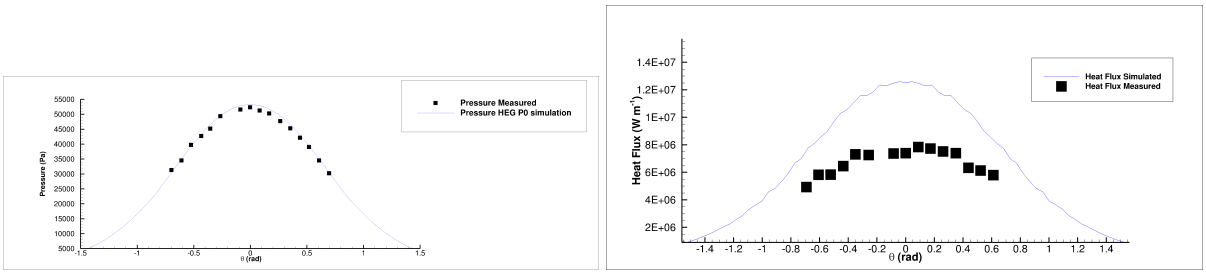


Figure 5.10: HEG cylinder P0 simulation: horizontal velocity of the air. The shock wave is clearly seen in the large jump in velocity.



(a) Pressure at the wall for the HEG P0 simulation. **(b)** Heat flux at the wall for the HEG P0 simulation.

Figure 5.11: Wall quantities for the HEG P0 simulation compared with the measured values presented in the work of Knight, Longo, Drikakis, *et al.* [27].

moving away from the wall, which involves high gradients and can be more unstable.

5.3.5. HEG Case Conclusion

The HEG case introduced and described in this chapter. This case consists of supersonic flow flowing against a cylinder forming a detached shock wave after which thermal and chemical nonequilibrium are found. The physical case was first given, from which a simulation setup was derived. The simulation was modelled as a 2D reacting flow with a two temperature model, and given the physical quantities from the physical case for the initial and inlet boundary conditions.

The mesh for the case was given. It contains the flow facing half of the cylinder in 2D, discretising the flow field in front of it. The mesh was created such that the boundary layer is discretised with the first node a micrometer from the wall and with a progressive spacing. A coarse mesh was given near the shock wave with the intent of using the result of this simulation to be used as the initial conditions of higher order ones. As such, a non smooth shock was expected.

The setup was given for the case in COOLFluiD with emphasis on the physical model used, the time integration used, the spatial discretisation, the boundary and initial conditions, as well as COOLFluiD specific options. The boundary conditions were given, assuming supersonic inflow and outflow, based on the results of the physical case.

The results of the simulation were reported an analysed. Residual wise convergence was found, converging all residuals bellow -6. The shock wave was found to be at circa 0.011 m away from the cylinder along the stagnation line. Formation of new species was found after the shock wave, including that of electrons and ions, justifying the chemical nonequilibrium assumption. A different jump between the two temperatures was also found at the shock wave, showing a clear separation and, hence, thermal non equilibrium. The temperatures converge to 300 K as the flow near the wall, nearing also each other.

Wall quantities were plotted in figure 5.11. An increase of heat flux was found relative to the physical case. This shows an over estimation of the heat flux. The pressure was along the wall and was found to show good agreement with the measured physical values.

The P0 case showed an improvement in the capabilities of COOLFLuiD with the ability to run a thermochemical nonequilibrium simulation in P0 without diverging. This study is expected to be completed with higher order simulations in future sections, better converging the solution. The follow up goals from this section were identified as converging a P1 or higher order simulation using this result as the initial solution, thereby achieving better agreement for the wall quantities and shock stand off distance and shape.

5.4. Application to the Kranc Case

The Kranc case consists of of a blunt body being impacted by supersonic argon plasma. The argon is expected to dissociate behind the shock wave, originating Ar^+ and electrons.

The goal of this simulation is to see if the positivity method can guarantee the positivity for it, as well as give rise to a residual wise converged solution which can be used to restart in a higher order simulation. As such, the mesh is not created with the goal of discretising the shock wave, much like for the HEG case. It should be expected that reruning this case from this solution at a higher order would give a better discretisation for the shock wave. Furthermore, starting a simulation from scratch involves much more drastic conditions than from a restart, given that the shock wave forms at the wall and travels part of the domain to its position. The P0 simulation helps, essentially, jump that step which might be

destabilizing for higher order simulations.

The simulation here described, hence, should converge residual wise while having a detached shock wave. The setup of the mesh and the case file are here given in section 5.4.2, while the results are given in section 5.4.3.

5.4.1. Introduction

The dubbed here as the "Krance case" concerns a simulation based on the study present in the work of Kranc, Yuen, and Cumbel [82]. In this case, Argon plasma is directed at a blunt probe, thereby creating a shock away from it. Measurements regarding drag were taken by the probe, while optical measurements were used to measure the shock stand off distances as well as the species in the shock layer. The main focus of the work present in Kranc, Yuen, and Cumbel [82] is that of studying the influence of a magnetic field in affecting the shock position, the drag on the body and the different drag related forces (viscous forces, pressure and Lorentz force). The study of the influence of magnetic field is, however, not the goal of the current research here presented. Furthermore, the current physical model does not take such fields into account.

The case without magnetic field can still be used as test case. This case allows the test of a chemically reacting flow that is different from air, as it concerns only argon, ionised argon and electrons. It, then, improves variety in the test cases, while also demonstrating the modularity for different species of the developed code.

The flow conditions present for this case are given in chapter 3 of the report by Kranc, Yuen, and Cumbel [82]. The flow is steady and supersonic. The flow quantities were measured experimentally or derived from these measurements. They require being measured due to the experimental facility not allowing for the direct input flow conditions such as velocity, but rather indirect conditions, such as the pressure of the tank that holds the argon to be used.

The flow conditions given can be used to justify supersonic boundary inlet boundary conditions, given that the air is moving at Mach 4.75 [82]. Furthermore, the temperature and velocity of the flow are also given in chapter 3 of the report by Kranc, Yuen, and Cumbel [82], where they are derived from measured quantities. The density of the flow is also calculated, from which, based on the given degree of ionisation, the electron number density is derived. Assuming neutrality, all densities are, thus, trivially available. These inlet conditions may be found in the setup regarding the inlet in table 5.6.

The wall temperature is, based on the report, to be at ambient temperature in the lab. It can be taken as approximately 300 K.

Regarding the wall quantities, the report largely focuses on the increase in them due to the magnetic field. However, it also mentioned that the total aerodynamic drag and pressure drag coefficients are 1.5 and 1.0 respectively for the probe used, however such values are estimated rather than measured and, hence, will not be used for validation [82]. The other quantity of interest is that of the shock stand off distance, which is given as 0.35 inches which is converted to 0,00889 cm.

5.4.2. Simulation Setup

The simulation was setup with with an FR scheme in COOLFluiD for spatial discretisation, and with PLATO for the chemical source term. This section provides the details for the setup of the simulation.

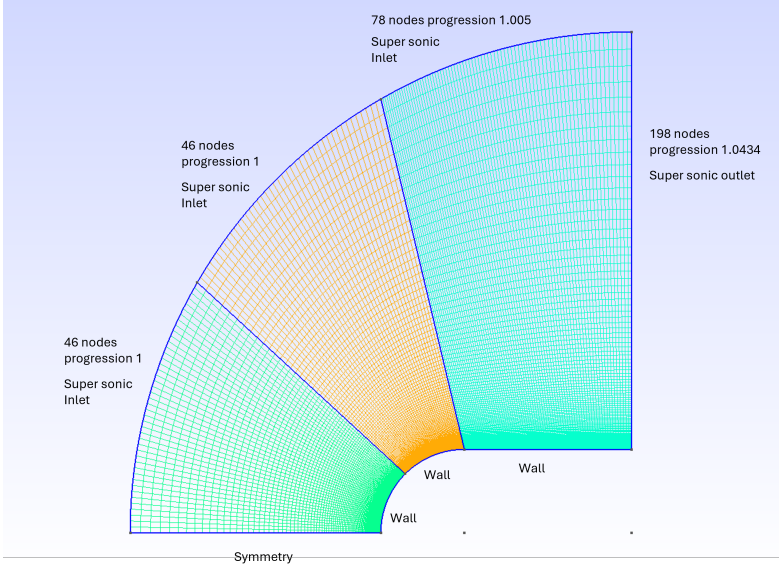


Figure 5.12: Kranc case mesh for P0 simulation

5.4.2.1. Mesh

The mesh for the Kranc case is given in this section. The mesh is made with the goal of giving rise to a simulation result that may be restarted in higher order. The mesh may be found in figure 5.12. The mesh was created such that the first node from the wall is at circa 10^{-6} m from the wall, as a first estimate. The physical lines are given in figure 5.12, and show the location of the inlet, walls and of the outlet. Both the inlet and the outlet are supersonics, as the air in this test case enters and exits supersonically. The walls impose a no slip and impose that no flow goes through it. They also discretise the geometry of the probe in Kranc's case. The symmetry allows the simulation of only the upper part of the domain, as the result is symmetric given the symmetric conditions and the presence of a final steady state solution.

5.4.2.2. Case setup

The setup of the case file is given in this section. This case is for a P0 simulation. The spatial and temporal discretisation choices, as well the choice for the fluxes and chemical source term are given in table 5.5. The boundary conditions are given in table 5.6, while the initial conditions are given in table 5.7.

In contrast to the HEG case, this case makes use of a single temperature formulation, focusing on the chemical nonequilibrium only. In all other choices, it is analogous to the P0 HEG case, given in section 5.3.3. In section 5.3.3, the details for these choices were already given, with sole exception for the single temperature formulation. This formulation is given directly as an option PLATO, to the positivity method, as well as passed to the spatial discretisation. Furthermore, the T_v is no longer present in the update variables.

The given formulation allows to define a case in COOLFluiD.

5.4.3. Results

The results of the Kranc case are here given. The residuals of the last restart are given in figure 5.13. The case converged all the residuals to bellow -5. The residuals are given in order of the equations: three density equations – one for each species – two momentum equations and one energy equation,

Table 5.5: Setup for P0 Kranc case (excluding boundary and initial conditions).

Parameter	Choice
Chemical model	PLATO
Physical System	Navier Stokes 2D Non Equilibrium with chemical source term
Spatial Method	Flux Reconstuction
Temperature Model	Single Temperature
Riemann Solver	AUSMPlusFluxMS2D
AUSMPlusFluxMS2D A12	3
Correction Function FR	VCJH
VCJH constant	0.33
Time Integration	Forward Euler
CFL	Interactive (up to 0.5)
Update Variables	Species densities, velocity in x, velocity in y, T
Solution Variables	Conserved variables
Physicality	Least squares density correction with energy based correction
Finalisation	StdFinalize

Table 5.6: Boundary conditions of the simulation setup for P0 HEG cylinder case.

Parameter	Choice
Inlet BC	Supersonic Inlet
Inlet u BC (m/s)	3000
Inlet v BC (m/s)	0
Inlet T BC (K)	1100
Inlet ρ_{e^-} BC (kg/m^3)	$9.41 \cdot 10^{-12}$
Inlet ρ_{Ar} BC (kg/m^3)	$1.09 \cdot 10^{-4}$
Inlet ρ_{A^+} BC (kg/m^3)	$6.86 \cdot 10^{-7}$
Outlet BC	Supersonic Outlet
Wall BC	Isothermal and No Slip
Wall T BC (K)	300

respectively. The species, in order, are electron, argon and ionised argon. This is a sufficiently converged case residuals wise. The temperature residual was the largest one, followed by the two momentum ones.

The temperature and horizontal velocity are given in figure 5.16. The shock wave is clearly visible in both of these plots, with the velocity plot showing a clear jump to subsonic velocities near the vertical part of the shock (normal to the flow). It, stands, however, diffused over several elements, showing a lack of good convergence to its location. It stands between x=0.036 and x=0.048, making the distance to the wall along the stagnation line to be between 0.02115 m and 0.00915 m. The lower boundary give an error of ≈9.29% relative to the reference quantity of 0.00889 m. The shock wave is also thinner at the bottom, near the symmetry, diffusing over more elements near the outlet. This could and should be related to the coarse mesh chosen, which would influence the numerical diffusivity as well as impede the shock from moving close enough to the correct shock stand off distance to converge to the steady state.

The species densities show a a rise after the shock wave, and a larger and steeper rise near the wall. Per consequence, given the wide range of orders of magnitude, with a much larger magnitude at the wall, making it hard to differentiate rises in colour maps – as seen in figure 5.15. A rise after the shock wave is seen, but orders of magnitude smaller than that at the wall. An horizontal slice was taken at the height of 0.0061 m, cutting the lower part of the domain and intersecting the wall of the probe. The species densities along this slice is given in figure 5.16. A clear rise is seen starting at after the 0.04 m

Table 5.7: Initial conditions of the simulation setup for P0 HEG cylinder case.

Initial Condition	Choice
T (K)	1100
u (m/s)	3000
v (m/s)	0
Inlet ρ_{e^-} BC (kg/m^3)	$9.41{\cdot}10^{-12}$
Inlet ρ_{Ar} BC (kg/m^3)	$1.09{\cdot}10^{-4}$
Inlet ρ_{A^+} BC (kg/m^3)	$6.86 \cdot 10^{-7}$

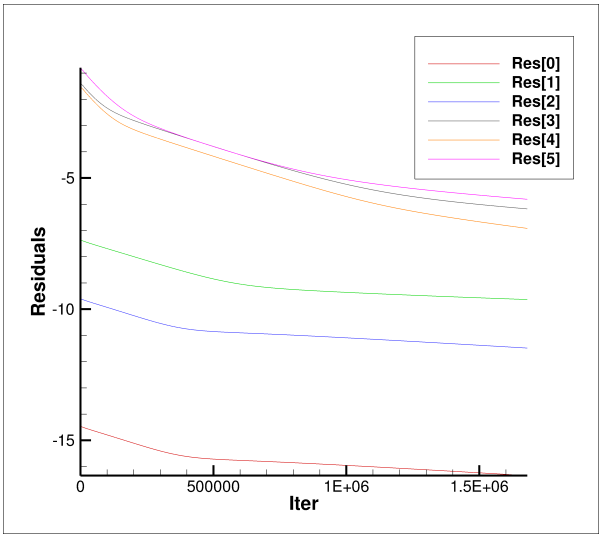
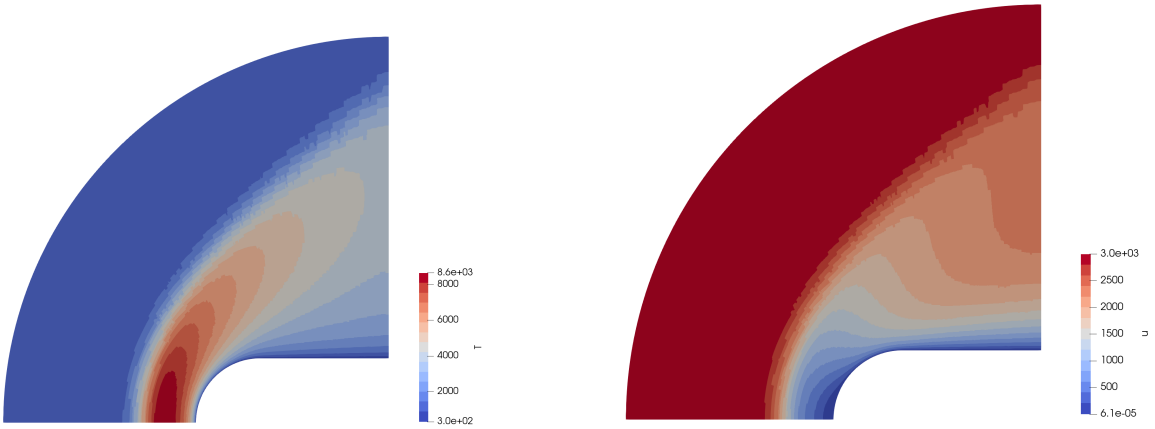


Figure 5.13: Residuals of the last restart of the Kranc P0 simulation simulation.



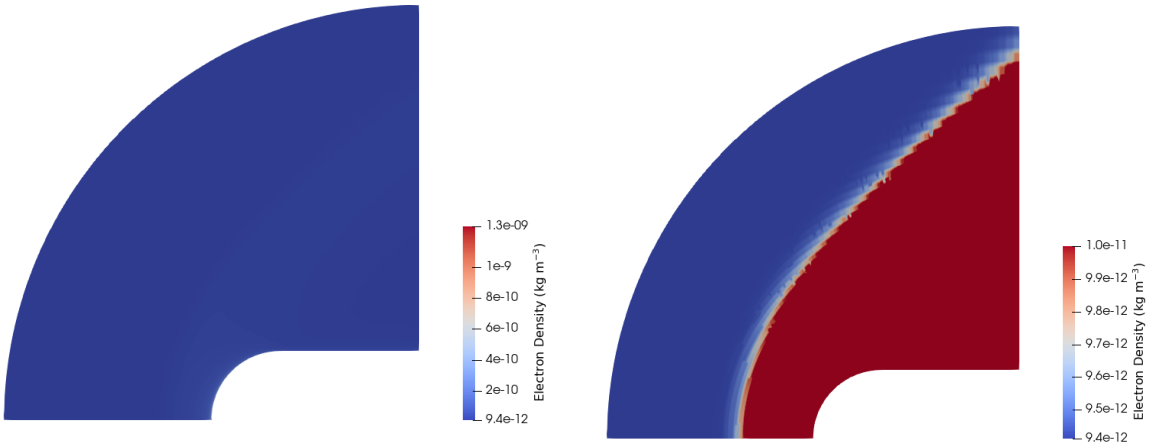
(a) Temperature in the P0 Kranc case.

(b) Horizontal velocity in the P0 Kranc case.

Figure 5.14: Temperature and horizontal velocity in the Kranc case. A shock wave is clearly visible. However, it spans several elements. Mesh artifacts on the shock wave are visible. Spatial convergence is not possible for P0 with the current mesh, but this result might serve as a future restart.

mark, with a steep increase near the wall for all species. The rise at the wall eclipses the rise at the shock wave by one or more orders of magnitude. The values before the shock wave are those of the freestream, already given in table 5.6.

This solution showed another case where COOLFluiD successfully ran a P0 case without diverging



(a) Electron density. The colour map uses the maximum and minimum values. The rise in electron density is barely perceptible, indicating a big rise in a confined region of the domain. A glow near the wall indicates higher density there.

(b) Electron density. The colour map values were capped at 10^{-11} kg m^{-3}. There is a clear rise after the shock wave.

Figure 5.15: Electron density for the Kranc P0 case.

or crashing, showing an increase in capability from before the implementation of the new positivity method. However, given the clear diffused shock wave, this case, much like the HEG one, is not yet spatially converged, and is to be restarted in P1 or higher for future studies.

5.4.4. Kranc Case Conclusion

The Kranc case was presented and described. It consisted of Argon being directed at supersonic speed against a blunt object representing a probe. The mesh was created by outlining the shape of the blunt object and discretising the flow field surrounding it. The symmetry of the case was used in order to only discretise the upper part of the domain, making use of a symmetric boundary. The rationale of the mesh was the same as for the HEG case, discretising mainly the boundary layer, with the mesh purposely coarse as to serve for future P1 or higher simulations.

The setup of the case was given, thereby giving the choices for the spatial discretisation, the temporal integration, the boundary and initial conditions, the positivity method, as well as COOLFluiD specific choices. This physical model was chosen to capture the chemical nonequilibrium nature of the flow, including chemical source terms and multiple species. A single temperature model was adopted. The other choices were similar to those for HEG, following the standard setup for a supersonic simulation with FR.

The simulation was converged residual wise. A diffused shock wave was found detaches at a distance between 0.02115 m and 0.00915 m from the wall along the stagnation line. Increase in the density of all species was seen after the shock wave, with a further increase in one or more orders of magnitude next to the wall.

This case further reinforced the increased capabilities of COOLFluiD in simulating P0 cases without divergence. The new positivity schemed worked, guaranteeing positivity all across the domain, despite the large gradients at the wall and at the shock. Given the non spatial convergence of this solution, this case is deemed to be restarted in future studies in P1 or higher order, with the expectation of being a more stable and closer to the final solution initial condition.

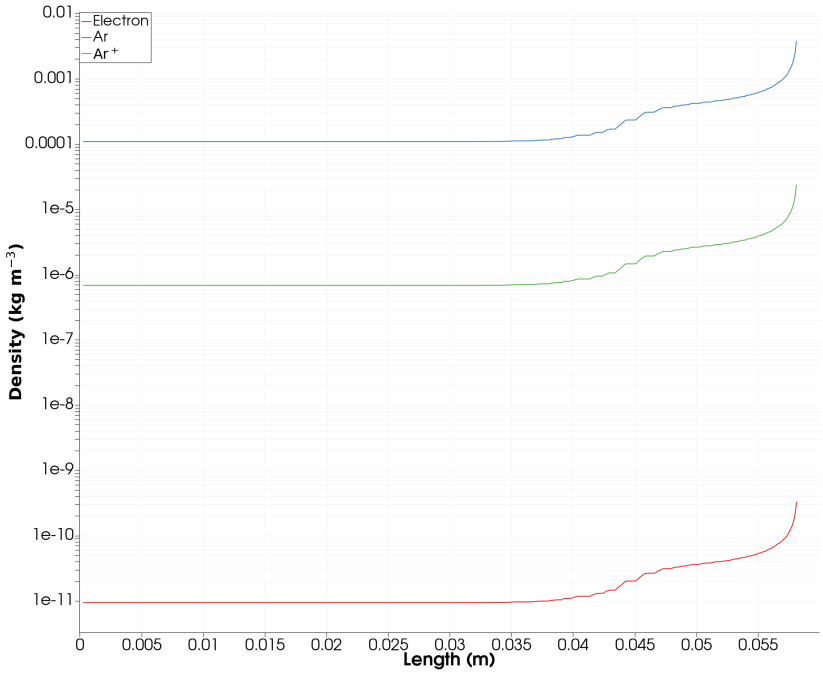


Figure 5.16: Species densities along an horizontal slice at y=0.006119 m. A clear rise in density starts between the 0.4 and 0.45 m mark, which is in the region immediately after the shock wave. An additional rise in several orders of magnitude is seen across all species near the wall.

5.5. P1 Cases

The P0 cases described in this chapter showed successful use of the LS method for positivity. However, they require being restarted in at a higher order than P0 in order to converging the shock wave position and the wall quantities, as well as the flow field in general. A finer mesh could be used, instead, but such would go against the purpose of using a flux reconstruction instead of a low order method. As seen for the wedge case, in section 3.7, a reasonably coarse mesh is fine enough to converge a shock wave given a higher enough order.

The setup for P1 cases are here described, followed by their results. As it will be seen, these cases will diverge. A series of test cases to identify the reason for divergence will be given, based not only on the results, but also on the knowledge of functioning cases for the code. It will be argued, at the end of this section, the need to inspect the code, which will be subject of chapter 6.

The setup of the cases will be given in section 5.5.1, while the results will be presented in section 5.5.3. A further batch of tests will be presented in section 5.5.4, with their results given in the same section.

5.5.1. Setup

The setup of the cases is given in this section. They follow the setup of the cases in P0 already given in this chapter, with some exceptions. First, an implicit scheme is preferred in most runs, as the P1 nature of the run is more unstable than P0. Second, P1 runs already allow for the filter to be used. As such, the entropy based filter is used together with the least squares method with the energy based correction. The calculation of the strength of the filter is done without the use of the LS scheme, as already discussed in section 4.5. Some runs chosen – as it will be seen in section 5.5.2 – will make use of the explicit scheme forward Euler. These will make use of the same setup as the one described for P0,

with the exception of the addition of the entropy based filter to the positivity scheme. The choices for the implicit ones is given in table 5.8.

The additional setup for the HEG and the Kranc cases is given in tables 5.9 and 5.10. This is related to the fact that these cases do not both use two temperature modes.

Not mentioned in the setup tables, all fluxes and source terms were chosen with the option for the Jacobian to be calculated and taken into account. Such comment requires being made due to the fact that there are two versions of the same fluxes in COOLFluiD, one of which does not compute the Jacobian, and that the source term requires having the Jacobian set to true.

Table 5.8: Setup for implicit P1 cases.

Parameter	Choice
Chemical model	PLATO
Physical System	Navier Stokes 2D Non Equilibrium with chemical source term
Spatial Method	Flux Reconstruction
Riemann Solver	AUSMPlusFluxMS2D
AUSMPlusFluxMS2D A12	3
Correction Function FR	VCJH
VCJH constant	0.33
Time Integration	Backwards Euler
Time contribution to Jacobian	StdTimeRHSJacob
CFL	Interactive
Solution Variables	Conserved variables
Physicality	Entropy based Exponential filter and least squares density correction with energy based correction
Finalisation	StdFinalize

Table 5.9: Additional setup for the HEG P1 cases.

Parameter	Choice
Temperature Model	Two Temperatures
Update Variables	Species densities, velocity in x, velocity in y, T and T_v

Table 5.10: Additional setups for the Kranc P1 cases.

Parameter	Choice
Temperature Model	Single Temperature
Update Variables	Species densities, velocity in x, velocity in y, T

5.5.2. Runs chosen

This section regards the runs chosen for the case. In order to test, a batch of test was developed for HEG and Kranc with the intent of going over a wide range of AV strength. The AV parameters chosen were based on the suggestions present in the work of Vandenhoeck [2] for the P1 case. The CFL was chosen such that the starting residuals were bellow 3, with only the last two residuals – related to the energy equation and the vibrational temperature – allowed to be above 2. This choice was based on experience with P0, where such was common or even larger at startup. Species related residuals (the first 11 for HEG or the first 3 for Kranc) were always found bellow 0 (log scale) at startup.

The runs chosen are presented in table 5.11. The suffix "A" was added to the run numbering in order to differentiate it from the P0 related runs. The runs 2A to 14A were run with backwards Euler as the integration method. They survey a wide range of AV strength by changing the Péclet number (tuning

factor for the AV; the bigger, the smaller the weaker the AV) between 75 and 0.5. The S0 and the κ were set to -1.5 and 1, respectively, following the suggestions present in the work of Vandenhoeck [2].

A single forward Euler run was executed with high AV and a small CFL of 10^{-6} in order to check whether forward Euler could have any merit in being used for P1 simulations.

Table 5.11: Runs chosen for the P1 case. The suffix A was given to the run numbering to differentiate them from other batches. The runs focused on tuning the AV to the case.

Run	Case	Time Integration	S0	κ	Péclet	CFL	Convergence
1A	HEG	Forward Euler	-0.5	1.5	0.5	10^{-6}	Diverged
2A	HEG	Backwards Euler	-1.5	1	0.5	0.01	Diverged
3A	HEG	Backwards Euler	-1.5	1	2	0.01	Diverged
4A	HEG	Backwards Euler	-1.5	1	3	0.01	Diverged
5A	HEG	Backwards Euler	-1.5	1	10	0.01	Diverged
6A	HEG	Backwards Euler	-1.5	1	15	0.01	Diverged
7A	HEG	Backwards Euler	-1.5	1	20	0.01	Diverged
8A	HEG	Backwards Euler	-1.5	1	40	0.01	Diverged
9A	HEG	Backwards Euler	-1.5	1	75	0.01	Diverged
10A	Kranc	Backwards Euler	-1.5	1	0.5	0.1	Diverged
11A	Kranc	Backwards Euler	-1.5	1	5	0.1	Diverged
12A	Kranc	Backwards Euler	-1.5	1	10	0.1	Diverged
13A	Kranc	Backwards Euler	-1.5	1	20	0.1	Diverged
14A	Kranc	Backwards Euler	-1.5	1	75	0.1	Diverged

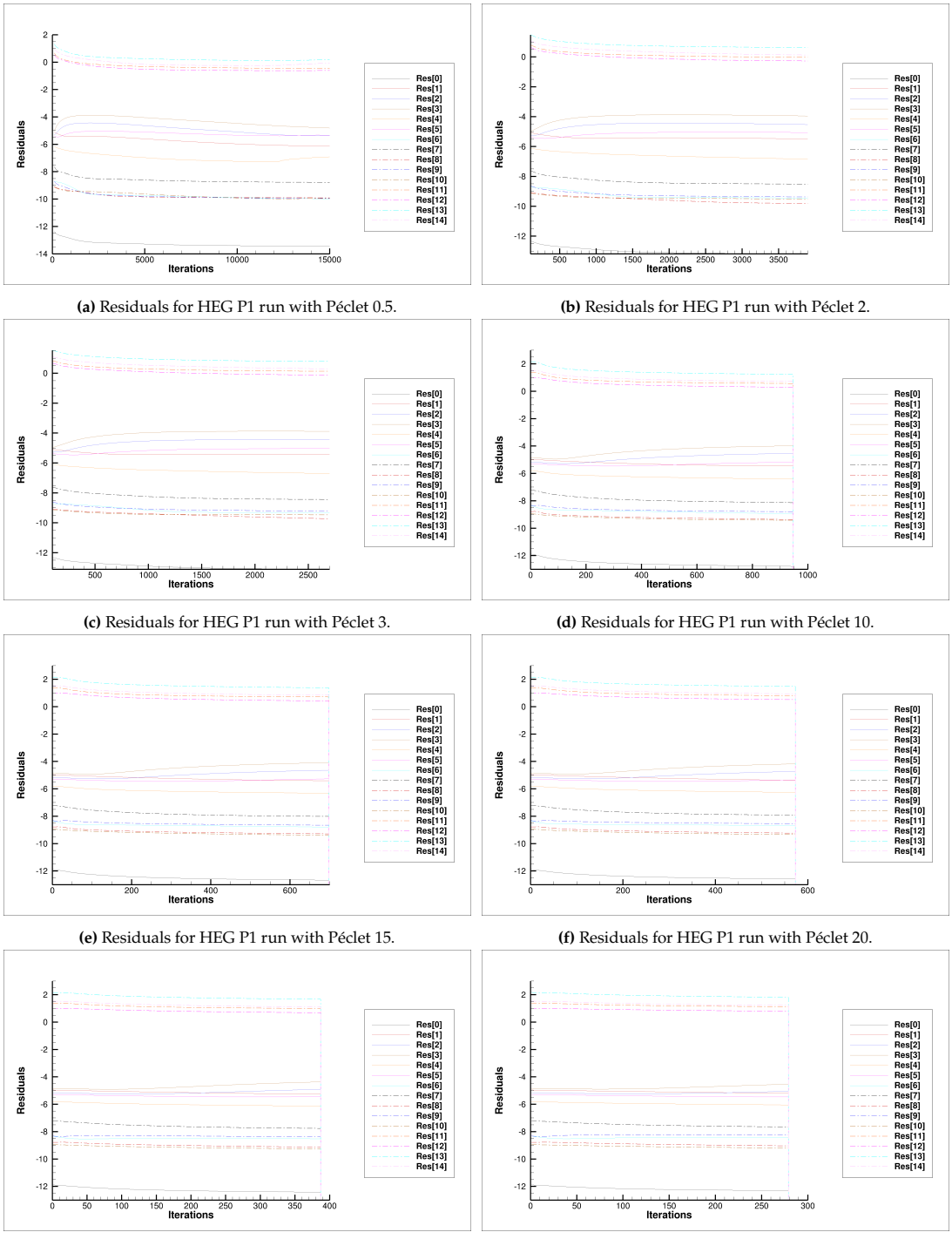
5.5.3. Result Analysis

As seen in table 5.11, all runs diverged. The analysis of the results and possible reasons for divergence are given here. The common aspects are first addressed, followed by analysis of sample cases in order to analyse the likely faults.

Regarding these general aspects, no negative densities nor temperatures were found during run time in any case, showing that the positivity method did not fail. Negative temperatures were found in the final simulation files, as a result of the last time step where the simulation diverged. These were the result, thus, of this diverging time step, and not of the failure of the positivity scheme as this scheme was not activated after the final time step (due to divergence). Thus, any negative temperatures found did not lead to the divergence of the simulations as much as they were the consequence of said divergence.

Regarding residuals, as stated in the previous section, the initial residuals were bellow 3 for the energy and temperature related residuals as well as for momentum ones, and bellow 0 for the density equation ones. This may be seen in figures 5.17 and 5.18. The residuals were steady for all the HEG simulations until blowup. The Kranc ones had mostly steady residuals with the exception of the simulations with the least AV. For these, a sharp increase in residuals may be seen before blowup. The HEG cases with the most AV lasted the longer amount of iterations, showing a positive effect of the AV. The Kranc cases, on the contrary, lasted the longer for the least amount of AV. Both of these assertions may be seen in figures 5.17 and 5.18 by noting the iterations axis.

All HEG cases showed the artificial viscosity on top of the shock wave, with the smoothness factor being also higher there – meaning a more oscillatory behaviour (see section 2.3.3). The result for run 4A is shown in figures 5.19 and 5.20. All HEG results are analogous to this one, with the amount of AV changing between them. As such, this run is used for further analysis. Figure 5.19 details the temperature T over the domain. The shock wave, despite appearing smoother than before, does not



(a) Residuals for HEG P1 run with Péclet 0.5.

(b) Residuals for HEG P1 run with Péclet 2.

(c) Residuals for HEG P1 run with Péclet 3.

(d) Residuals for HEG P1 run with Péclet 10.

(e) Residuals for HEG P1 run with Péclet 15.

(f) Residuals for HEG P1 run with Péclet 20.

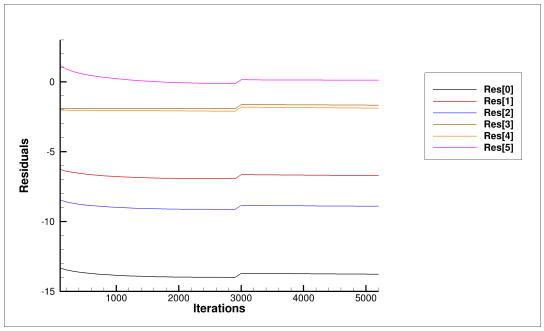
(g) Residuals for HEG P1 run with Péclet 40.

(h) Residuals for HEG P1 run with Péclet 75.

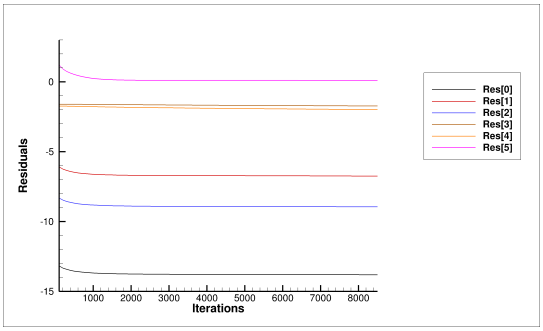
Figure 5.17: HEG runs' residuals.

show the influence of AV to an expected extent for such a high value as a Péclet of 3. For comparison, the Wedge case runs seen in section 3.7.5 made use of a Péclet of 15, while having a much smoother shock wave. The detail in figure 5.19b shows variations along the shock wave and still a jagged look along it. Several peaks to 13000 K are seen in this figure, with gaps along the shock wave. As such, it could be argued that the AV is not diffusing as much as expected, or working as intended.

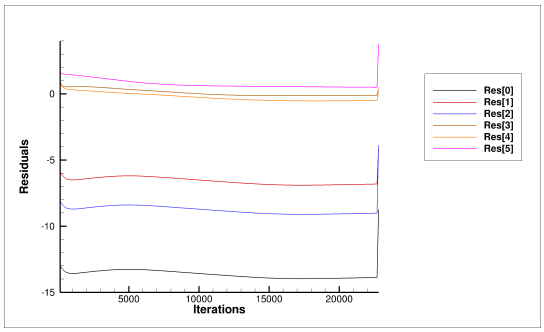
Figure 5.20 regards the smoothness of the flow, the AV placement and the filter placement. The



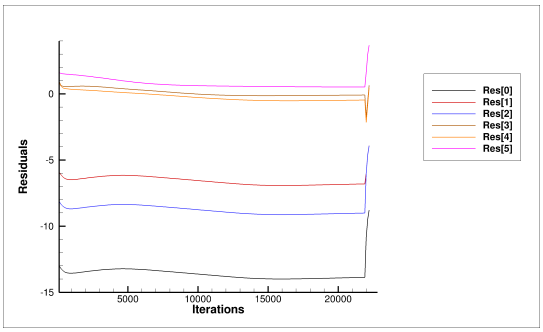
(a) Residuals for Kranc P1 run with Péclet 0.5.



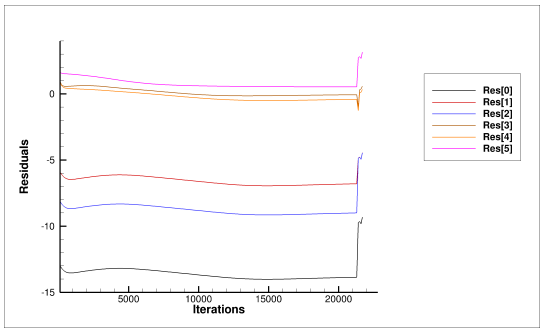
(b) Residuals for Kranc P1 run with Péclet 5.



(c) Residuals for Kranc P1 run with Péclet 10.

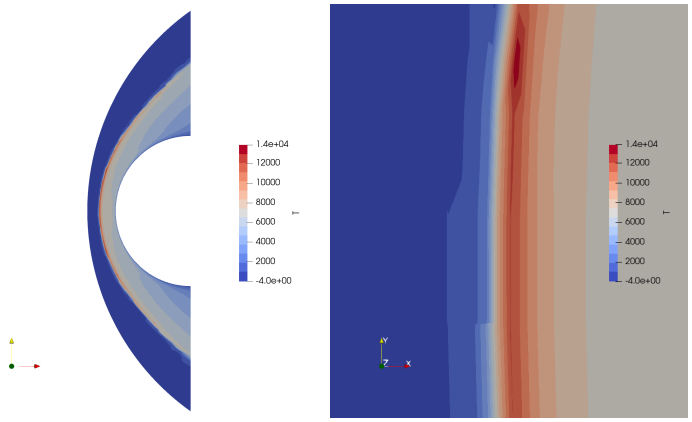


(d) Residuals for Kranc P1 run with Péclet 20.



(e) Residuals for Kranc P1 run with Péclet 75.

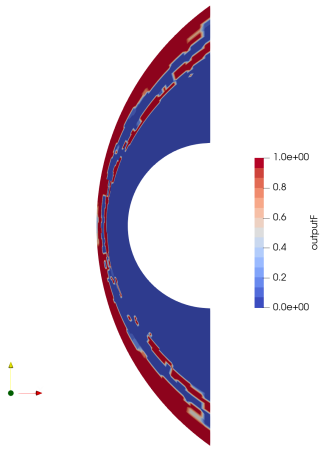
Figure 5.18: Kranc runs' residuals.



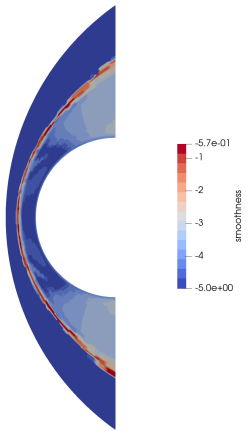
(a) Temperature T.

(b) Detail of the temperature along the front of the shock wave.

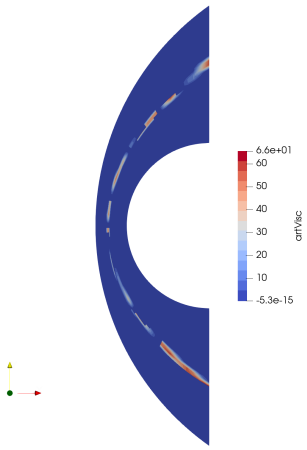
Figure 5.19: Run 4A: HEG P1 case with AV Péclet 3 – temperature T in K.



(a) Exponential filter strength. The maximum strength is 1 and the smallest possible (not active) is 0. This is the symmetric case to the strength described in the theoretical description, but it is done so to help visualising.



(b) Smoothness measurement. The minimum value was set to -5 in order to allow for visualisation by allowing distinguishing values between -5 and 0. Actual minimum value is -100.



(c) Artificial viscosity strength.

Figure 5.20: Run 4A: HEG P1 case with AV Péclet 3 – Smoothness related quantities. The AV is clearly on top of the shock wave, but parted instead of continuously. The smoothness clearly is higher on the shock wave. The filter is active in front of the shockwaves likely due to density positivity violations for the species due to numerical oscillations upstream of the shock.

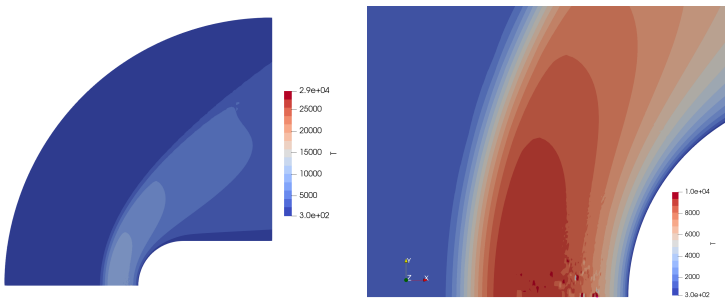
smoothness is indicated by the smoothness factor, which is detailed in the work of Vandenhoeck [2], and in section 2.3.3. The solution is smoother the lower the value of the smoothness factor, which is capped at -100 in log scale. The smoothness in figure 5.20b is shown with the minimum value of -5 to avoid obfuscation of the variation at higher values due to the smoother part of the domain having a very low smoothness factor. The smoothness seems to outline the shock wave, with values near -0.57. These values are high, and, hence, might indicate that the simulation diverged due to the damping of spurious oscillations not being appropriate.

The exponential filter strength is shown in figure 5.20a shows to areas of damping: over the shock wave and in front of it. Filtering the shock wave might indicate that there are positivity issues there, or that the entropy constraint is being violated possibly due to spurious oscillations lowering the temperature. Given that the temperature seems to rise progressively, as seen in figure 5.19b and that there are species in the inflow that are not present, it is more likely that it is avoiding negative densities there. However, this indicates that the AV is not acting to avoid these negative values.

The AV is clearly being sparse on top of the shock wave, showing possible issues with the AV application. This sparse pattern follows mesh lines. It could also be a visualisation problem related to the recording of the AV in a parallelised manner in the file, given that it seems to follow mesh partitions. Such patterns have previously been seen by the author in COOLFluiD for Euler cases. However, it could also be produt of some problem with the code regarding the AV placement.

The filtering is occurring in front of the shock, near the inlet. This seems to indicate that shock related numerical errors are propagating upstream and not being mitigated, implying the necessity of the filter to correct and keep positivity. This further suggests that the shock is not being appropriately dampened.

Several explanations are possible for the ineffectiveness of the AV. It could be possible that the simulation did not run long enough to allow the diffusion of the AV. Alternatively, it could also be that the AV is too low, not allowing for enough diffusion. Given that the smoothness and the AV are in the correct place, it can be discarded that the AV is not being applied in the correct place. An alternative option



(a) Temperature T.

(b) Detail of the temperature along the front of the shock wave.

Figure 5.21: Run 23A: Kranc P1 case with AV Péclet 10 – temperature T in K detail. The colour map was capped at 10000 K to permit the visualisation of the numerical artifacts, as well as of the shock wave.

would be that the code is not applying the AV fluxes properly, either due to failing to construct them correctly or due to estimating their Jacobian incorrectly, thereby evolving them in time incorrectly. This could be due to a bug in the program or a wrong setup of the case file. The bug in the program can be related to either a typo – missing or wrong calculations – or due to inconsistencies, as before the cases were being run with different variable set: conserved instead of primitive ones. Furthermore, a different model is being used – multispecies with chemistry and chemical non equilibrium instead of Navier Stokes – whose implementation could have issues. The different variable sets make use of different change of variable functions as well as have different calculations for thermodynamic variables (such as pressure). These inconsistencies would undoubtedly influence the flux calculations.

Regarding the Kranc case, the case with Péclet 10 – which is closer to an advisable AV for P1, while still being high – is here taken for further study. Figure 5.21 shows the temperature of the Kranc case for P1. The temperature grew to a much larger value than reasonable behind the shock wave, creating patchy numerical artifacts. It also created negative temperatures in the same area. These problems are not identified near the shock wave, reflecting that the divergence is unlikely due to the AV or shock related instabilities. Furthermore, the shock appears smoother than for P0, showing that the AV seems to be working as intended or that is not required for this level of smoothness.

Following the same thought process as for the analysis of the HEG case, the implicit method seems to be at fault. Alternatively, the CFL could be too large for this case, leading to an diverging behaviour.

All cases were, thus, found to diverge regardless of AV. The CFL was small enough to have small residuals. The cases, however, showed small influence on the shock wave due to the AV, despite the high values and the presence of the AV over the shock. This suggests that the AV is not working as intended.

Regarding the changes between the P0 and P1 cases, besides the increase in order, it is possible to identify that these are: the use of Backwards Euler as an integration method, the need of the computation of the Jacobian, the use of AV and the use of the filter for positivity. Given the results presented, the positivity does not seem to be the origin of the problem, unless some unidentified interaction with the AV is present. Implicit methods have been in use in COOLFluiD and predate the FR code, making them unlikely to be wrong in themselves. However, their related parameters, such as the preconditioner, could be inappropriate for these cases. Still regarding the use of an implicit method, the computation of the Jacobian could also be affecting the solution. Given the smaller values involved, namely for the densities, the perturbations for the numerical derivatives could be inadequate. Furthermore, given the

fact that the code for FR with chemical nonequilibrium – and, hence, specific for this variable set and for several species –was not being tested until now in these conditions, an undetected error or inconsistency could be present in the code. The same applies for the computation of the AV.

Another possible cause for divergence regards the variables used. For the Wedge case, conserved variables were used for the computation of the case, while here primitive variables are being used. Any missing variable transformation or inconsistency could lead to problems in the spatial discretisation and, hence, on the convergence of the solution. Given that the P0 cases converged, it is not expected that these problems should be present in the same functions as the ones used for the P0 case. This restricts any future analysis to the aforementioned changes between the P0 and P1 case.

5.5.4. Test Runs

Given the diverged cases analysed in the previous section, further runs were elected to better narrow down the problem. Given that one of the differences with the P0 case was the use of backwards Euler, it could be that a simple lack of numerical accuracy was present. As such, the run 5A was repeated with different available preconditioners, with different number of Krylov spaces and with different perturbations for the calculations of the numerical Jacobian. The results are given in table 5.12. With exception of the numerical error associated with too small of a perturbation for numerical derivatives, all other runs gave the same divergent behaviour and the same end result, showing that the accuracy of the implicit was not the root cause of the problem.

Given that the Lax-Friedrichs flux scheme is more diffusive and has successful been used in hypersonic cases [2], it was also attempted. However, it showed divergent behaviour in fewer iterations without improvement in the shock wave diffusion problem. The cases were also run with the filter without the entropy considerations in order to see if the filter was having a negative effect on the shock wave by filtering more than just positivity. This also lead to divergent behaviour.

These results seem to indicate that the implicit method is not at fault in terms of lack of accuracy due to incorrect parameters. It should be noted that while increasing the number of Krylov spaces or increasing the numerical perturbation for the derivative (within reason) are expected to lead to a more accurate solution, the use of a Jacobi preconditioner (PCJACOBI) is not, and was used purely for checking if the Schwarz method one (PCASM) was not misbehaving for this particular case for an unknown reason. The name of the preconditioners in the table 5.12 follow the names of the implemented scheme in PETSc.

In order to further test the implicit scheme, a case with the backwards Euler as the time integrator was used to further converge the solution presented in section 5.3.4. The same setup as for the previous P0 case was used with the exception of the the Backwards Euler option instead of the Forward Euler one, and the additional computation of the Jacobian.

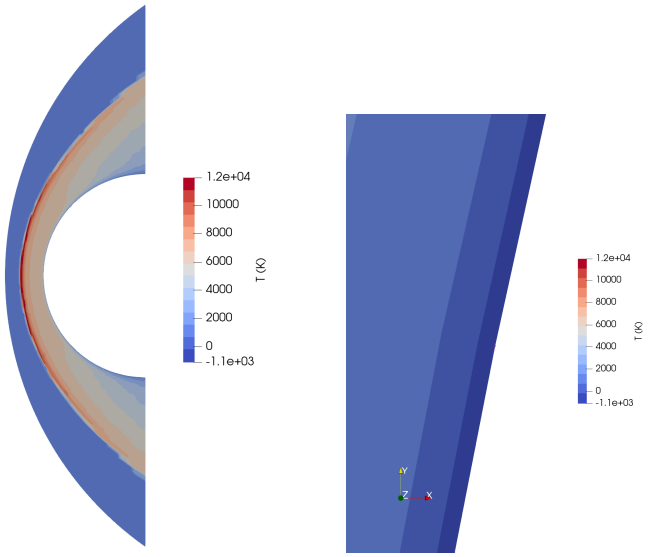
Restarting a solution for that cases always led to a momentary jump in residuals between restarts, so it is not expected that the initial residuals should be as low as the ones presented in section 5.3.4. As such, the case was also started with a lower CFL in order to avoid immediately diverging the case due to an overly large CFL. However, this case with the HEG in P0 diverged after the CFL being increased from 0.05 to 0.1, as seen in figure 5.23. The forward Euler solution previously presented had made use of a CFL of 0.5 without issue. This CFL sensitivity is, thereby, unexpected. Furthermore, the backwards Euler scheme is expected to be more stable than the forward Euler one, possibly allowing for CFL values larger than 0.5. This further suggests that the implicit method does not work for the current setup.

Table 5.12: Alterations tested for the HEG case. These alterations were run for cases with Péclet 3. No difference was found in the result, all diverging.

Run	Case	Preconditioner	Number of Krylov Spaces	Numerical Jacobian Perturbation	Convergence
1B	HEG	PCASM	30	10^{-8}	Divergence
2B	HEG	PCASM	30	10^{-10}	Divergence
3B	HEG	PCASM	30	10^{-13}	Numerical Error
4B	HEG	PCASM	80	10^{-7}	Divergence
5B	HEG	PCASM	150	10^{-7}	Divergence
6B	HEG	PCJACOBI	30	10^{-7}	Divergence
Run	Case	Riemann Solver	Convergence		
7B	HEG	LaxFriedrichsFlux	Divergence		
8B	HEG	AUSMPlusFlux2D	Divergence		
Run	Case	Physicality	Convergence		
9B	HEG	LS method	Divergence		
10B	HEG	Filter without entropy considerations and LS method	Divergence		
Run	Case	Relative Norm for Linear System Solver	Convergence		
11B	HEG	10^{-15}	Divergence		
Run	Case	Time Integrator	Convergence		
12B	HEG	Newton Iterator	Divergence		

Table 5.13: P0 HEG simulations with backwards Euler

Run	Case	Order	Integrator	CFL 13B	Restart	Convergence
	HEG	P0	Backwards Euler	0.05 (658 iterations); 0.1	Yes, from P0 case	Diverged



(a) Temperature plot

(b) Temperature detail at the wall: lowest temperature in the domain.

Figure 5.22: Run 13B: HEG P0 with backwards Euler as the time integrator – temperature plot

Inspecting the temperature for this run in figure 5.22, the shock wave looks barely changed, while the wall now has a negative temperature. The element with negative temperature is seen in figure 5.22b. This is an abnormal failure in positivity never seen with the explicit method. It could be explained by a

large transport of energy to the wall in a single integration, thereby estimating the temperature to be negative. This would imply that the error is not of a spurious oscillation (as it could be in higher order ones), but rather due to a wrong calculation either of the fluxes or of the Jacobian. While the backwards Euler implementation in itself could be at fault, having been in use for successfully for the Wedge case, it is unlikely that it should at fault.

This explanation is in accordance with the residuals (see figure 5.23). Before the divergence of the residuals, the energy equation related residual had grown to 3.1821465. This, being in log scale, is excessively high and already a product of an upwards trend in the residuals. Such a high residual would be related with large variations in the temperature, which wold be in accordance with the negative temperature at the wall.

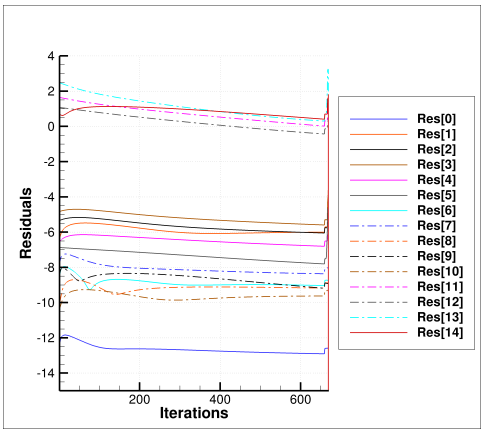


Figure 5.23: Residuals of the HEG P0 case with backwards Euler.

As such, this case suggests that the implicit method, in its current setup, cannot be used to simulate thermochemical nonequilibrium simulations. However, the reason is yet unknown from this analysis. Given that the Backwards Euler method has been seen to work for the Euler case, it could be conjectured that it is related to either: the new physical model (and associated functions) or the variables used (primitive instead of conserved ones). Another possible cause would be that the thermochemical nonequilibrium cases would, for reason yet unknown, behave worse for the implicit case than for the explicit case. Given that the backwards Euler should work better, namely for stiff cases such as those with chemistry, it would be expected for this to outperform its explicit counterpart. As such, such would suggest that the Jacobian was having problems properly characterising the cases – possibly due to inappropriate perturbations (already addressed) – but no reason is yet found to motivate this conjecture.

5.6. Answers to Research Questions

Given the analysis, partial answers to the research questions can be given. Regarding the question
Can the least squares method guarantee positivity of the densities during FR simulations for the thermochemical nonequilibrium case?
it can be argued that the LS method can guarantee density positivity for the P0 and P1 tested cases. Furthermore, the method was not active at convergence, and did not impede the convergence of the P0 test cases, strengthening its applicable use. This answer seems to suggest that the method can guarantee density positivity during FR simulations for the thermochemical nonequilibrium case, but requires further testing for fully converged states in higher orders.

The question :
Can the exponential filter together with the least squares method guarantee positivity of the densities and temperatures during FR simulations for the thermochemical nonequilibrium case?
was answered positively specifically for the cases tested. However, since those cases did not converge, it cannot be ruled out that that the method might fail. However, given that the conditions are most dire during this restart, it is not expected that further problems would be found for these test cases, suggesting that this method should work.

Further testing is required in order to more convincingly answer both of the questions. Such requires, however, a further questioning into the reasons for divergence of these simulations. The possible reasons as to why were already conjectured during the analysis of the results of the simulations. These will be dealt with in chapter 6.

5.7. Conclusion

The research questions regarding the use of the methods implemented for positivity were established. The method to address them was expressed, based on test cases. The test HEG and the Kranc case were selected based on their use of the required physical model and variety in species. The cases were run in P0 and in P1, with their setup motivated by literature and based on the available options in COOLFluiD.

The cases for P0 was found to converge residual wise. The HEG case converged with the shock wave near the correct position. It showed a correct wall pressure distribution and an overestimation of the heat flux at the wall. The shock wave requires further spatial convergence, which can be done through increasing the order of the method. An increase in electrons and NO^+ was found after the shock wave, showing the presence of ionisation, as expected, and showing the importance of including chemical reactions. Thermal nonequilibrium was also found at the shock wave, with the temperature T being higher than T_v. Thermal equilibrium was, however, found away from the shock wave, with both temperatures matching. The case found no negative densities or temperatures, and the positivity method was turned off at convergence, showing a converged solution with positive values.

The Kranc P0 case was found to also converge. Chemical species are found to change behind the shock wave, with a greater presence near the wall. The shock wave is diffused over several elements, possibly due to the natural diffusivity of the coarse mesh. The simulated shock wave position encompasses the expected shock wave position.

The P0 solutions were restarted in P1. No case was found to converge. All cases ran from hundreds to thousands of iterations before diverging. The shock waves were found to lack diffusivity, suggesting that the AV is not working as intended.

Based on the differences between the P0 case and the P1 case – the integration method being Backwards Euler now and the use of AV – and also on the changes between the Wedge case and the new cases, it was conjectured that either the AV or the Backwards Euler was not active appropriately. Such behaviour was conjectured to possibly come from either the AV implementation for the multispecies case, or from the calculations of the fluxes or Jacobian for the multispecies case; the latter case being possibly tied to the difference between the update and the solution variables, or to the specifics parameters of the linear system solver and Jacobian calculation.

The HEG case with Péclet 3 was further tested with alterations in the setup in regards to the preconditioner

and the convergence norm of the linear system, the numerical perturbation to the Jacobian, the Riemann solver, the positivity method and the integrator. No changes were found.

In order to test if the backwards Euler was appropriate, it was used in a P0 HEG simulation, showing a CFL restriction much lower than for the explicit case, and diverging. This seemed to suggest that the backwards Euler was not appropriate.

These results seem to suggest the inspection of the implementation and use of both the integrator backwards Euler as well as of the AV for the multispecies case before proceeding to new simulations.

6

Code Analysis and Required Implementations

The previous chapter – chapter 5 – concerns itself with the application of the new positivity method in P0 and P1 simulations. However, P1 simulations were unable to converge. Based on the analysis therein presented, it was argued that the AV and the implicit method were likely the causes of such behaviour. Consequentially, this chapter concerns a analysis of the COOLFluiD code – specifically the flux reconstruction discretisation part – with the intent of uncovering the inconsistency or error. Also following the reasoning of the last chapter analysis, the emphasis is given on the differences between the code regarding the multispecies implementation with thermochemical nonequilibrium and the one for Euler, as the latter was seen to be working for the Wedge case.

The code analysed, with the exception of the positivity methods developed, may be found in its GitHub page [4].

This chapter is first presents motivating questions which guide the analysis in section 6.1. This is followed by a short section which presents the main loop of the spatial discretisation in FR (section 6.2). An analysis of the AV fluxes is the primary target of section 6.3. There, not only inconsistencies and missing parts in the AV formulation are described, but also the implementations executed in the face of said problems are tackled. The time integration problems found are linked to the use of the Backwards Euler option in section 6.4. The Newton Iterator option, also available in COOLFluiD, is found to be appropriate in the face of the problems found for the Backwards Euler option. The required care in using it is tackled in section 6.5. The corrections and implementations described in these sections are summarised in section 6.6. The initial motivating questions are answered in section 6.7.

The changed code sections are presented in figure 6.1, which concerns the FR discretisation algorithm. This algorithm was presented in the wider framework of COOLFluiD in figure 3.3.

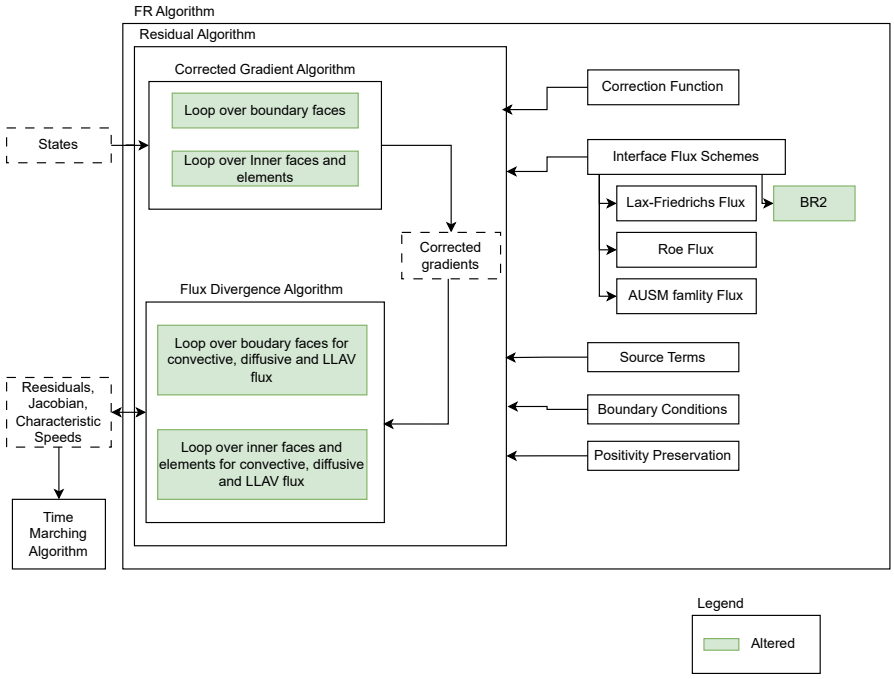


Figure 6.1: Parts of the code altered. The diagram is adapted from the work of Vandenhoeck [2].

6.1. Motivating Questions

Based on the findings in the previous chapter, it as chosen to inspect the code for understanding of the possible failure to converge the simulations as well as the specific behaviour of the implicit method and the AV. This chapter focuses in said analysis and findings. This section presents the motivating questions which are sought to be answered:

- Is there an implementation error in the AV for the multispecies case?
- Is the AV for the multispecies model incomplete?
- Is there an inconsistency in the setup of the case file regarding the integrator?
- Is there a variable set (primitive or conserved variables) inconsistency in the code?
- Is the backwards Euler method correctly defined for the multispecies case?

All of the above questions focus on the AV and the integrator, as per the results of the last chapter. The AV is check for either an error or possibly an incompleteness in being extended to these new cases, since it works for the Euler case.

Given that one of the changes from the Euler case to the thermochemical nonequilibrium one is the use of non conserved variables as the update variables, it is possible that some inconsistency is present in the code. As such, that is also target of inspection.

Given that backwards Euler was found not to work for P0, it is also target of inspection. Special emphasis is put in the calculation of the fluxes and of the Jacobian, possibly due to the afore mentioned change in variables.

The analysis regards only the spatial discretisation part of the code. This is the section of the code where

the flux reconstruction method is used an applied.

6.2. Main Loop

As already addressed in section 3.6.2, the spatial discretisation in COOLFluiD for flux reconstruction is done in sequential form. It follows the scheme presented in figure 3.2. This figure is here presented again in figure 6.2 for clarity.

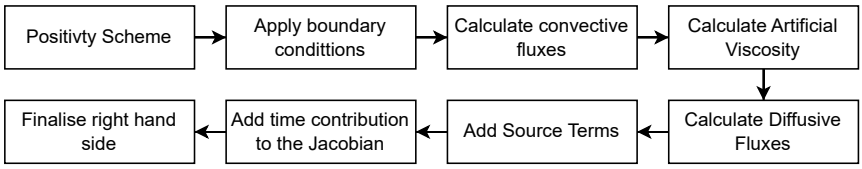


Figure 6.2: Spatial discretisation in COOLFluiD

It is evident from this loop that two main parts require inspection: the computation of the fluxes and the addition of the time contribution to the Jacobian. These two both concern the previous sources of the problem identified.

The positivity was already analysed and implemented in chapter 4 and is consistent with the code. Thus, it is not reanalysed. The source term has already been validated for finite volume in the work of Sharma, Giangaspero, Munafò, *et al.* [6], which makes use of the same variable types as flux reconstruction (the primitive variables). Hence, it is here not target of further analysis.

6.3. Fluxes Computation

In the COOLFluiD implementation of the fluxes, all flux computations follow the same base idea. First, an object oriented approach is used, with further specialisation of the flux computation based on a more general implementation. Second, an execute function is defined which runs the main flux computation loop over all faces. The actions done for each face are presented in figure 6.3. The convective fluxes related execute function differs in looping over the cells for the computation of the discontinuous flux. Figure 6.3 follows the approach for diffusive fluxes and for the AV. The auxiliary functions to this execute function perform the required calculations and allocations of the results, and will be the focus of this analysis. They follow the typical inheritance rules of C++.

The fluxes are divided into three main parts: convective fluxes, diffusive fluxes and artificial viscosity fluxes. The convective fluxes and diffusive fluxes further divide into inner cell fluxes and boundary fluxes. The artificial viscosity does not diffuse across boundaries.

Based on the analysis in the previous chapter, the AV viscosity is addressed in section 6.3.1.

6.3.1. Artificial Viscosity Fluxes

The artificial viscosity used in the nonequilibrium simulations of the last chapter concerns the use of the class LLAVJacobFluxReconstructionNS in the code. The inheritance structure is given in figure 6.4. The function to execute the artificial viscosity belongs to its parent class, and is the same as the one used for the Wedge case. The first part of the analysis focuses on the parts which are overridden by the LLAVJacobFluxReconstructionNS class.

This section concerns the problems found with the AV formulation, including hidden assumptions and

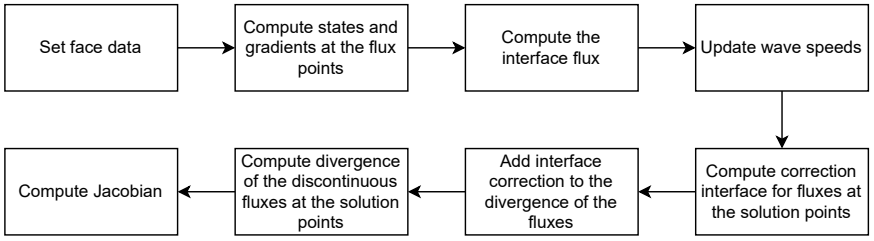


Figure 6.3: Execute function for each face for diffusive fluxes in COOLFluiD. The function for convective fluxes calculates the same using a distinct algorithm

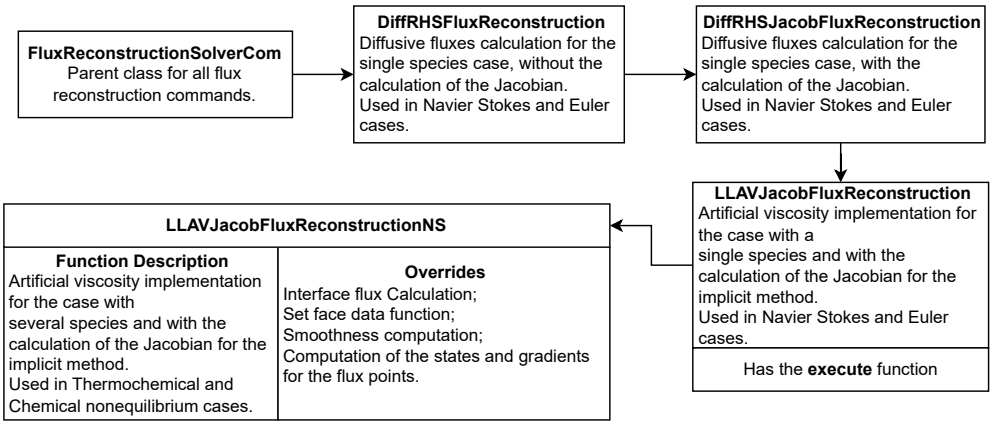


Figure 6.4: Artificial viscosity class and its parent classes.

lack of full extension of the AV implementation for the TCNEQ case. This required the implementation of gradient calculation and perturbed gradient calculation for the TCNEQ case. Changes to the interface flux computation were also required and implemented. The required implementations are motivated and explained in sections 6.3.1.1 and 6.3.1.2.

6.3.1.1. Interface Flux and Gradient Calculation

The implementation for the nonequilibrium cases overrides the functions for the calculation of the interface fluxes. These are the fluxes calculated at the interface between two elements, and which are used in equation 2.3.8 to correct the fluxes at the solution points. This function loops over the flux points at the interface and applies the following equation for each one:

$$F_{\mathrm{AV}} = \epsilon_{\mathrm{AV}} \cdot \left(\frac{\nabla(u^{-}_{\text{cell face}}) + \nabla(u^{+}_{\text{cell face}})}{2} - \frac{\alpha_{\text{damping}}}{l_{\text{characteristic}}} (u^{-}_{\text{cell face}} - u^{+}_{\text{cell face}}) \boldsymbol{n_f} \right), \tag{6.3.1}$$

where $u^{-}_{\text{cell face}}$ and $u^{+}_{\text{cell face}}$ are the value of values stored in the flux point for the left and right cell, respectively; the $\nabla(u^{-}_{\text{cell face}})$ and $\nabla(u^{+}_{\text{cell face}})$ the gradient values stored on the flux point for the left and right cell respectively; and ϵ_{AV} is the artificial viscosity strength, assumed known at the time of the application of this function; α_{damping} is a user given damping value, usually set to 1; and $l_{\text{characteristic}}$ which is a characteristic length equal to the average of the cell volumes divided by the area of the interface. This is the formulation used to calculate AV fluxes as well as diffusive fluxes that are not AV. Importantly, no variable transformations are given in this function. Thus, the function expects an agreement between the variables used for the gradients and the ones used for the states at the flux points. The latter are the primitive variables chosen by the user for nonequilibrium cases: density,

velocities, and temperatures.

The flux computed, hence, is expected to be in primitive variables, according to equation 6.3.1. However, the artificial viscosity is defined for the conserved variables [2]. As such, either a variable transformation is used to calculate transform these fluxes at a later stage, or the gradients are in/supposed to be in conserved variables, requiring an additional variable transformation before the use of the damping term. The gradients are, thus, analysed. The gradients on the flux points interface are extrapolated using the Lagrange polynomial that represents the gradients in the solution points in element. Thus, it has the same variables as the ones in the solution points. The same applies for the states at the interface. Both of these quantities are calculated in the LLAVJacobFluxReconstructionNS class, as seen in figure 6.4. Direct inspection of the code shows that this is done as intended.

The gradients at the solution points are given through the set face data function in LLAVJacobFluxReconstructionNS, among other values also computed by this function. This function connects the socket for gradients for the AV to the correct variable in the code. In particular, it overrides the setting of this same variable with the gradients for the calculation of the diffusive fluxes, which are not in the same variable set, and which is done in in the DiffRHSFluxReconstruction class.

The gradients are not computed in any of the classes shown in figure 6.4. Instead, they are delegated, in the multispecies case, to the class NSJacobGradientComputer. This class is instantiated by choosing the the option for convective fluxes for multiple species in the case file. It concerns both the computation of the gradients for AV, as well as the computation of the gradients for all the other fluxes. It has as its parent class ConvRHSJacobFluxReconstruction, which has the execute function to compute the convective fluxes. Hence, the gradients for the AV, when AV is used with this convective flux choice, are calculated here.

Inspecting the code directly, it is clear that this class computes the gradients for the AV twice: first based on the extrapolation with the Lagrange polynomials, and then correcting them through by applying equation 2.3.8. These are two different functions, invoked during the loop over the faces in the execute function of LLAVJacobFluxReconstruction. Both of these compute these quantities for when the conserved variables are being used, but not when the primitive ones were chosen instead. Comparing these functions with their counterparts for the explicit time integration case – implemented in class NSJacobGradientComputer – it is found that the AV gradients are computed for the primitive variables as well, showing an inconsistency.

The difference between the implementation for the conserved variables and the primitive variables is an additional variable transformation. This transformation transforms the states in the solution points – written with the update variables chosen in the case file which are the primitive variables – to the solution variables, which are also chosen in the case file, and chosen to be the conserved variables.

Analysing the counterpart for the boundary gradients – NSJacobBndGradientComputer – the same inconsistency is found. This leads to the understanding that either the gradients are being calculated elsewhere and should not be changed for the implicit case, or that the both calculations are simply not computing the gradients for AV for the implicit case with primitive variables. No other setting for this gradient socket was found, however, in the code.

Thus, the gradients are not being correctly calculated for the AV, altering both the interface fluxes and the inner fluxes. The AV flux is, however, not zero in neither during runtime since the jump in variables

at the interface seen in equation 6.3.1 can be non zero. It should be recalled that the interface flux affects the fluxes at the solution points through equation 2.3.8.

6.3.1.2. Perturbation of the Gradients

For the calculation of the Jacobian for the AV fluxes – required for implicit methods – the divergence of the fluxes must be perturbed appropriately. The numerical derivative for some function f to some variable p_k (the variable stored in the solution point) is taken by:

$$\frac{\partial f}{\partial p_k} \approx \frac{f(p_k + \epsilon_{\text{Jacob}}) - f(p_k)}{\epsilon_{\text{Jacob}}}, \tag{6.3.2}$$

where ϵ_{Jacob} a numerical the perturbation. Evidently, the function f must be recomputed with $(p_k + \epsilon_{\text{Jacob}})$ as its argument. In the context of AV for FR, this function would represent the divergence of the AV fluxes. For this calculation, the effect on ϵ_{AV} as well as on the gradients must be computed, before the calculation of the AV fluxes and, subsequentially, their divergence. Concerning the computation of the fluxes, this calculation is performed in the code by the function computePerturbedGradientsAnalytical present in the class LLAVJacobFluxReconstruction. This function effectively computed the following:

$$q_{i,d,k,\text{new}} = q_{i,d,k,\text{old}} + \sum_{i}^{N_p} \epsilon_{\text{Jacob}} \frac{\mathrm{d}l_{id}}{\mathrm{d}\xi_d}(\boldsymbol{\xi}_i), \tag{6.3.3}$$

where $q_{i,d,k\text{new}}$ is the new gradient component in the solution point i, in the direction d for the k^{th} variable, and $q_{i,d,k,\text{old}}$ the unperturbed version of the same gradient. This expression trivially holds if the numerical derivative is taken to the same variables as the ones for the gradient. That is, if conserved variables are used in the gradient and are also the variables to which the numerical Jacobian is taken, then the expression holds. This is easily seen by noting that, in this case:

$$f(p+\epsilon_{\text{Jacob}}) = \sum_{i}^{N_p}(p_k+\epsilon_{\text{Jacob}})\frac{\mathrm{d}l_{id}}{\mathrm{d}\xi_d}(\boldsymbol{\xi}_i) = \sum_{i}^{N_p}(p_k)\frac{\mathrm{d}l_{id}}{\mathrm{d}\xi_d}(\boldsymbol{\xi}_i)+\sum_{i}^{N_p}(\epsilon_{\text{Jacob}})\frac{\mathrm{d}l_{id}}{\mathrm{d}\xi_d}(\boldsymbol{\xi}_i) = q_{i,d,k,\text{old}}+\sum_{i}^{N_p}\epsilon_{\text{Jacob}}\frac{\mathrm{d}l_{id}}{\mathrm{d}\xi_d}(\boldsymbol{\xi}_i) \tag{6.3.4}$$

However, since we mean to use conserved variables u_i instead of primitive variables p_i in the gradient, then this cannot be the case. Thus, the correction term in equation 6.3.3 must use the difference between the perturbed and unperturbed conserved variables, rather than the numerical perturbation directly. Thus, it should follow the following expression instead:

$$q_{i,d,\text{new}} = q_{i,d,\text{old}} + \sum_{i}^{N_p}(u_i(p_0, p_1, ..., p_k + \epsilon_{\text{Jacob}}, ..., p_{\text{last}}) - u_i(p))\frac{\mathrm{d}l_{id}}{\mathrm{d}\xi_d}(\boldsymbol{\xi}_i), \tag{6.3.5}$$

for a perturbation for the primitive variable p_k, which would be the k^{th} primitive variable.

The explanation in this chapter did not consider the correction part for the gradients present in equation 2.3.7. This part was already considered in section 6.3.1.1 in the NSJacobGradientComputer part where the correction is added. Since the variable is perturbed directly in the same place code variable (same place in memory) where the unperturbed one was stored, the correction applied in the NSJacobGradientComputer is valid. It should be noted that the variable is unperturbed at the end of the calculations for which the perturbed variable was required, before proceeding in the code.

6.3.1.3. Implementation of the Artificial Viscosity for Thermochemical Nonequilibrium

Based on the analysis of the previous sections, the artificial viscosity is found to be incomplete. Three implementations were, thus, executed: new interface fluxes, the gradient calculation and the perturbation of the gradients.

The interface flux function was changed to apply equation 6.3.1 with the $u_{\text{cell face}}$ making use of conserved variables instead of the primitive variables. This would guarantee the the flux formulation to be consistent.

The gradient computation for the TCNEQ case was implemented in the classes NSJacobBndGradientComputer and NSJacobGradientComputer, for the boundary and inner faces respectively. The implementation is twofold. First, the calculation of the gradients for the solution points without the face contribution is implemented into the function computeGradients. This function already computes the gradients for the diffusive (non AV) fluxes. The correction term for the influence of the faces is added to the function computeGradientFaceCorrections, also present in the mentioned classes. Both of these implementations execute a variable transformation from the primitive to the conserved variables before the rest of the calculations. Together, they implement the equation 2.3.7 with the conserved variables.

The perturbed gradients are implemented in the function computePerturbedGradientsAnalytical. The previous version of the function was rewritten in order to allow for the use of primitive variables. Thus, equation 6.3.5 was implemented in this function for all variables in each solution point.

These implementation allowed the code to be further extended to work for the TCNEQ case.

6.4. Jacobian Calculation and Backwards Euler

This section addresses how the Jacobian is calculated in the code and how it impacts the formulation of the system used by the Backwards Euler function in order to advance the simulation in time. An inconsistency in the variable set was found which does not allow the use of the Backwards Euler option. The inconsistency is detailed here.

The Jacobian calculation comes at the end of the execute function, as seen in figure 6.3. It calculates the partial derivative of the divergence of the fluxes to the variables in the solution points and adds them to a sparse matrix. This allows the construction of the linear system that represents the implicit time integration method.

The Jacobian calculation is done in the functions used for Euler cases, Navier Stokes cases and multiple species cases. As such, this analyses does not concern itself with possible errors in the code, but rather inconsistencies that might not be taking the case of different variable sets into account, or the use of functions present in further specialised classes for just multispecies cases.

The Jacobian calculation for the diffusive fluxes is presented in figure 6.5. The Jacobian calculation is analogous for the convective fluxes and for the AV fluxes. For each solution point, a variable (such as density, or temperature, for example) is perturbed, after which the gradients are recomputed, followed by the flux point values. The fluxes and their divergence are then computed for both neighbouring cells of the face. The difference in the divergence of the fluxes can then be used for the calculations of the numerical derivative. Thus, the Jacobian gives the variation in the divergence fluxes to the variation in the variable at the solution point, for all variables. This is correct from the standpoint of building a numerical Jacobian.

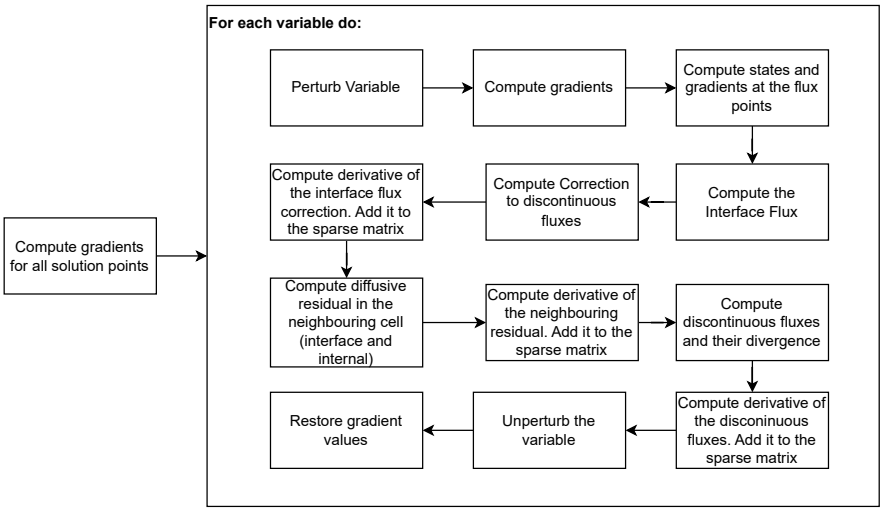


Figure 6.5: The Jacobian calculation for each solution point for each cell on the side of a given cell.

Importantly, the Jacobian will have the derivative in regards to the variable type used in the solution points. Thus, it must be in regards to the primitive variables when they are being used. The fluxes, however, are not necessarily in the primitive variables. This is a point where an inconsistency can occur.

The functions used for the gradients and for the flux computations already use variable transformations to before calling the flux computation function (such as a Riemann solver, for example), and thus should not be target of any problems, as the fluxes computation functions receive the correct set of variables. This was found to be true for all fluxes. All fluxes are, thus, returning fluxes of the conserved variables.

For the case with TCNEQ, where the primitive variables are used, the Jacobian has the numerical derivatives of the fluxes of the conserved variables in regard to the primitive variables. Special treatment is, thus, required. The backwards Euler system is given by:

$$p_k^{n+1} = p_k^n + \Delta t \left((\nabla \cdot F)_k + \left(\frac{\partial (\nabla \cdot F)_k}{\partial p_k} \right)_{|_t} (p_k^{n+1} - p_k^n) \right), \tag{6.4.1}$$

for each p_k for some iteration $n+1$, with the superscripts indicating the time iteration. For the case of the option for Backwards Euler, not further treatment is found for the Jacobian. However, the divergence of the fluxes is changed by the StdFinalize option from the conserved to the primitive variables. Thus, when Backwards Euler is applied, the divergence of the fluxes is in primitive variables while the Jacobian is in conserved variables. This is an inconsistency – as trivially evident by equation 6.4.1 – that does not allow the use of the option Backwards Euler in the code.

Another integration method is required. This method was found to be the Newton Iterator method. The work of Lani [5] details how the Newton Iterator takes into account the use of the Jacobian with primitive variables. However, such method was already tried. The reason for previous failures in using this method is addressed in section 6.5.

6.5. Relation between CFL contribution to the Jacobian and the Standard Finalise function

For the construction of the linear system, the CFL must be be taken into account in the Jacobian. However, under certain conditions, this can lead to inconsistencies with the standard finalise function implemented in the FinalizeRHS. These conditions are here addressed and explained.

6.5.1. Standard Finalise

The finalisation of the right hand side (by right hand side, it is understood to be the right hand side of equation 2.3.1, thus regarding the spatial discretisation in the method of lines) is the last step of the spatial discretisation, as seen in figure 6.3.

Usually this part is either omitted or ran with the standard finalisation implementation, stored in FinalizeRHS class. It concerns itself with a transformation of variables of the divergence of the fluxes. It does not act upon the Jacobian matrix. The transformation of variables is executed by making use of the numerical derivative of the primitive variables to the conserved variables at the point of calculation – that is, for a given variable , for a given solution point, for a given cell.

The vector holding the divergence of the fluxes is, then, left multiplied by the numerical derivative (matrix) around the given states for a flux point. This changes the divergence of the fluxes from using conserved to using primitive variables. The operation is given by:

$$(\nabla \cdot F)_k = \sum_j \frac{\partial p_{\text{primitive},k}}{\partial u_{\text{conserved},j}} (\nabla \cdot F)_j, \tag{6.5.1}$$

where $(\nabla \cdot F)_j$ is the divergence of the fluxes given in the primitive variable set, $(\nabla \cdot F)_k$ is the same but in the conserved variable set, $p_{\text{primitive},k}$ represents a primitive variable k and $u_{\text{conserved}}$, j represents a primitive variable j. For example, this would allow the change of the divergence fluxes from density, momentum and energy into density, velocity and temperature.

Given that this change of variables does not affect the Jacobian, there is an inconsistency here unless the Jacobian has its variables changed elsewhere, as already addressed in section 6.4. During its computation, the variables are not changed.

Importantly, in the code, equation 6.5.1 is applied by multiplying the change of variables matrix to a variable named rhs which should hold the divergence of the fluxes. As it will be seen in section 6.5.2, this is not always the case, namely for the option of the Newton Iterator.

6.5.2. Addition of the time contribution to the Jacobian

Once the Jacobian is calculated, given a CFL, a time variation must be included to it in order to build the numerical system. This is done by simply multiplying each entry in the sparse matrix that holds the Jacobian by the appropriate time interval. This can be done in two classes: StdTimeRHSJacob and PseudoSteadyStdTimeRHSJacob. The first one must be used with Backwards Euler, while the latter must be used with Newton Iterator. Failure to do so will result in an error being thrown at run time.

The StdTimeRHSJacob only adds the time contribution. However, the PseudoSteadyStdTimeRHSJacob further prepares the system. In it, beyond adding the time contribution, it also transforms both the

Jacobian and for variable rhs, which holds the divergence of the fluxes. This is done according to the transformation given in the work of Lani [5] for the use of Newton Iterator with primitive variables.

This change in creates an inconsistency with the use of the StdFinalize, as it will be changing the variable rhs after it has already been fully prepared for the Newton iterator system. Per consequence, the Newton iterator cannot work with the use of the StdFinalize as tested in section 5.5.4. Furthermore, it seems to suggest that the Backwards Euler option cannot run, with the current implementation, cases where the update variables and the solution variables are not the same. This undocumented – to the author's knowledge – behaviour further explains the divergent behaviour in chapter 5.

6.6. Results and Corrections Summary

The inconsistencies presented in this chapter explain why the setup used for the P1 cases led to divergence. Firstly, the backwards Euler, as it is present in COOLFluiD cannot be used with primitive variables, unlike the Newton iterator. Furthermore, the Newton Iterator cannot be used with the StdFinalize option.

Regarding the AV, the lack of the definition of the gradients for primitive variables for the multispecies case might explain the lack of smoothness in the shock waves for the P1 cases tested. As a solution, the calculation of these gradients were added to the NSJacobGradientComputer class and its boundary related class. Furthermore, the code for the function computePerturbedGradientsAnalytical was found not to be appropriate. Thus, a new implementation of the perturbation of the gradients was executed, allowing the expression 6.3.5 to be used instead.

These corrections and changes are expected to allow the AV to diffuse the shock wave better and for the simulation to be more robust. The new test cases, following the ongoing line of investigation of chapter 5 are tackled in chapter 7.

6.7. Answer to the Motivating Questions

Regarding the questions presented in section 6.1, after the presented analysis, they can be answered.

Regarding: "*Is there an implementation error in the AV for the multispecies case?*" it can be argued that the calculations of the gradients, the perturbed gradients and of the interface fluxes had an error in the implementation. However, both of these work for the use of the conserved variables as update variables. It is in fact a lack of extension or generalisation for the use of other variable sets.

Regarding: "*Is the AV for the multispecies model incomplete?*", it can be argued that it is indeed incomplete. It does not allow the use of the required update variable sets for thermochemical nonequilibrium, making it not usable for those cases. Such was found both in the calculation of the gradients, the interface flux as well as of the perturbed gradients for the Jacobian calculation.

Regarding: " *Is there an inconsistency in the setup of the case file regarding the integrator?*" it was found that there was one for when the Newton iterator option was used in section 5.5.4. The StdFinalize option cannot be used.

Regarding: "*Is there a variable set (primitive or conserved variables) inconsistency in the code?*" it was found that there were. Regarding the already aforementioned perturbed gradients, the lack of generalisation effectively consists of an inconsistency, since it always expects that the variables in the gradients are the

same as the variables being perturbed. Furthermore, it was found that the Jacobian is never transformed, unlike the divergence of the fluxes which are transformed in the class StdFinalize. As such, the Jacobian was not correct for the use with backwards Euler.

Regarding the "*Is the backwards Euler method correctly defined for the multispecies case?*" question, it can be said that the backwards Euler is only correctly defined if conserved variables are used as update and solution variables in this case. This method is, otherwise, unusable with the current code.

These questions led to the finding of several problems which were fixed as described in section 6.6. This effectively allows the extension of the code to a work with the thermochemical nonequilibrium cases that are sought after.

6.8. Conclusion

Based on the results of chapter 5, it was chosen to inspect the code for further clarification and possible corrections. Questions were posed that guided the search and analysis. They focused on the differences between the higher order Wedge cases and the P1 cases for thermochemical nonequilibrium, as to narrow down the analysis to the places where it was more likely to exist problems.

The artificial viscosity was found to have inconsistencies in the calculations of the interface fluxes and of the perturbed gradients, where the same variable set for the update variables and the solution variables was being assumed. The gradient calculations were not present for the cases with update variables: species denisity, velocities and temperature.

The Jacobian calculation was found to have all the required changes in variables before the calculation but no change afterwards. This leads to an inconsistency with the divergence of the fluxes, as it is modified by the StdFinalize. It was also found that this change of variables for the Jacobian is not present in the program, hence making backwards Euler unusable with the cases tested.

It was found that Newton iterator can be used with the cases tested as long as the StdFinalize is not used. The latter changed a variable in the system for Newton iterator, breaking the system and leading to divergence of the solution.

The inconsistencies for the AV were changed to what is understood here to be the correct formulation. The Newton iterator option was chosen for future runs.

7

Runs After Code Corrections

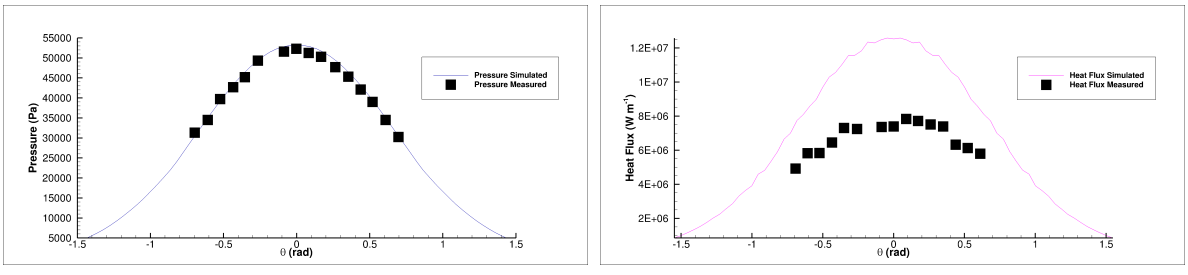
After the correction given in the previous chapter, a selection of runs were selected in order to further test these alterations as well as the in development methods. The HEG P0 case was rerun with Newton Iterator, and is presented in section 7.1, with the objective of showing that the Newton iterator can be ran in this new configuration. The P1 runs are tackled in section 7.2.

7.1. HEG P0 with Newton Iterations

The HEG P0 run was rerun now with the time integration option of Newton iterations, corresponding to solving the backwards Euler formulation with Newton iterations [5]. The same conditions as the ones described in chapter 5 were used, but now without the use of StdFinalize. The simulation converged residual wise and the results are here presented.

The temperature quantities, as well as the electron and NO^+ densities are given in figure 7.2. The wall pressure and density may be found in figure 7.1. These match the ones presented for the forward Euler case, showing how the Newton iterator can be used instead in this new configuration.

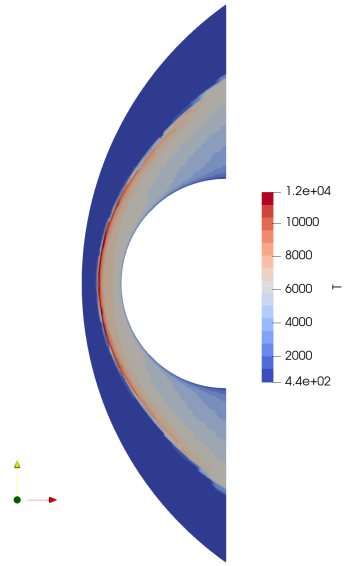
The residuals are given in figure 7.3, showing a good convergence. The simulation was stopped due to residual convergence automatically based on the stop conditions of relative norm of -6.



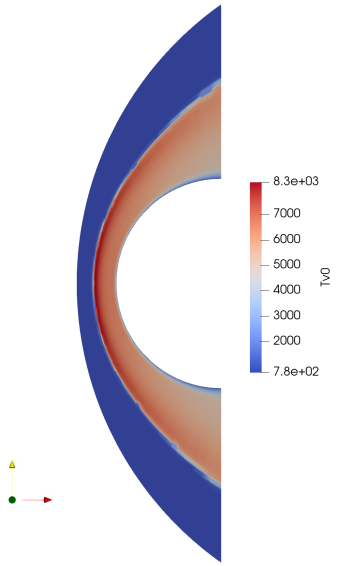
(a) Pressure at the wall for the HEG P0 simulation.

(b) Heat flux at the wall for the HEG P0 simulation.

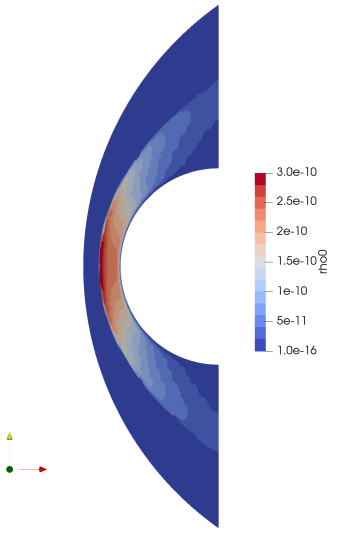
Figure 7.1: Wall quantities for the HEG P0 simulation compared with the measured values presented in the work of Knight, Longo, Drikakis, *et al.* [27].



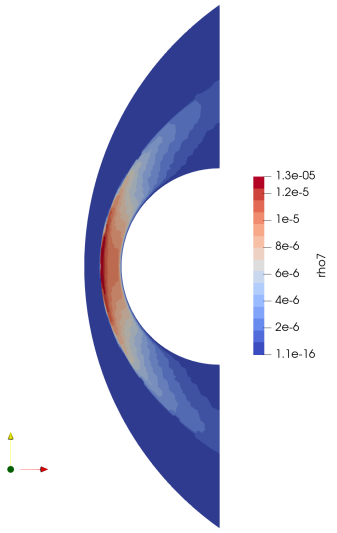
(a) Temperature T in K for the HEG P0 simulation.



(b) Temperature T_v in K for the HEG P0 simulation.



(c) Electron density in kg/m^{-3} for the HEG P0 simulation.



(d) NO^+ density in kg/m^{-3} for the HEG P0 simulation.

Figure 7.2: Temperature, pressure, and electron and NO^+ densities for the HEG P0 simulation.

This success shows that the work executed in chapter 6 has enabled implicit P0 runs with TCNEQ. While the heat flux is still not fully agreeing with the measured values, the focus is to find a spatially resolved run by increasing the order. This is in accordance with the fact that FR is meant to be run at higher order, and a P0 run could simply be run in FV instead. The current result successfully shows that P0 simulations can be run with positivity using the newfound method, and should be built upon at higher orders.

Further validity – despite the inefficiency and the consequent time consuming effort of running P0 simulations with a fine mesh with FR – could be found by further refining the boundary layer mesh as well as the shock. Given the chemically reacting nature of the case, the shock position affects the heat transfer [7], justifying the need to better discretise it.

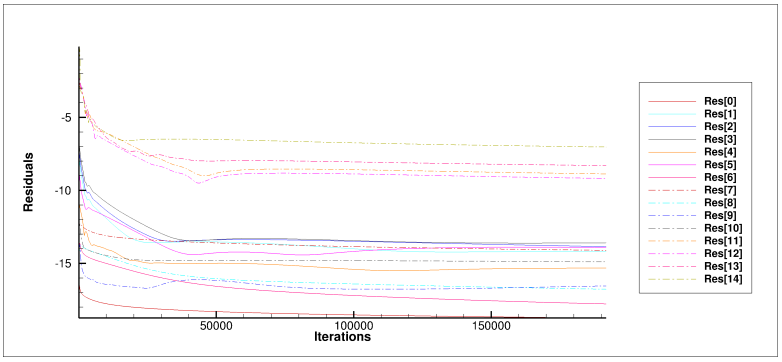


Figure 7.3: Residuals of the HEG P0 with Newton iterator.

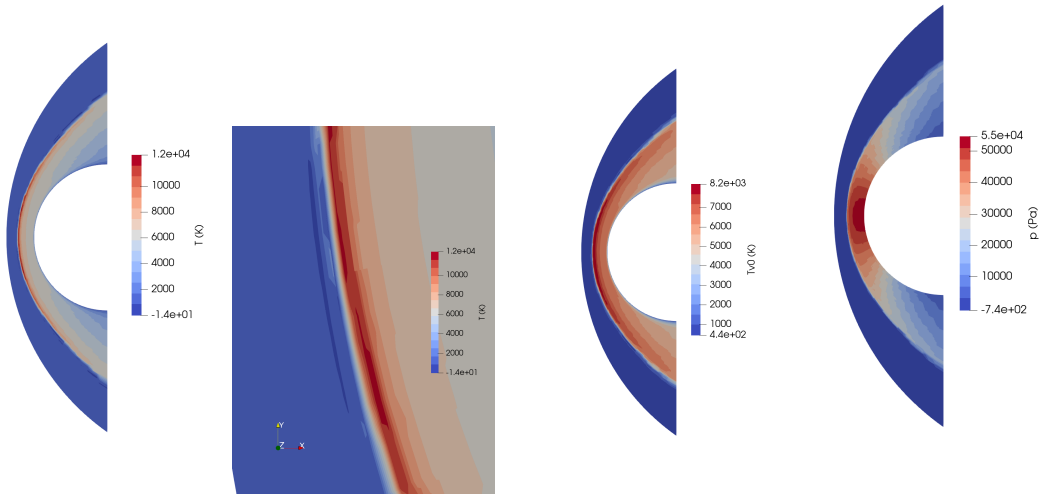
7.2. HEG P1 runs

A selection from the runs from section 5.5.2 were rerun. The runs were now run with Newton iterator and without the StdFinalize option enabled as previously described. The CFL was also changed to be smaller in order to accommodate possible CFL restrictions during restart. The runs chosen are presented and numbered in table 7.1. All other possible configuration options are unaltered from the previous P1 runs. The same boundary and initial conditions were also used.

Table 7.1: P1 runs with Newton Iterator.

Run	Case	Order	S0	κ	Péclet	CFL	Physicality	Convergence
1C	HEG	P1	-1.5	1	0.5	0.005	LS with Energy based Correction	Diverged
2C	HEG	P1	-1.5	1	1	0.005	LS with Energy based Correction	Diverged
3C	HEG	P1	-1.5	1	2	0.005	LS with Energy based Correction	Diverged
4C	HEG	P1	-1.5	1	3	0.005	LS with Energy based Correction	Diverged
5C	HEG	P1	-1.5	1	10	0.005	LS with Energy based Correction	Diverged
6C	HEG	P1	-1.5	1	20	0.005	LS with Energy based Correction	Diverged
1D	HEG	P1	-1.5	1	0.5	0.005	Exponential Filter + LS with Energy based Correction	Diverged
2D	HEG	P1	-1.5	1	1	0.005	Exponential Filter + LS with Energy based Correction	Diverged
3D	HEG	P1	-1.5	1	2	0.005	Exponential Filter + LS with Energy based Correction	Diverged
4D	HEG	P1	-1.5	1	3	0.005	Exponential Filter + LS with Energy based Correction	Diverged
5D	HEG	P1	-1.5	1	10	0.005	Exponential Filter + LS with Energy based Correction	Diverged
6D	HEG	P1	-1.5	1	20	0.005	Exponential Filter + LS with Energy based Correction	Diverged
7D	Kranc	P1	-1.5	1	0.5	0.075	Exponential Filter + LS with Energy based Correction	Diverged
8D	Kranc	P1	-1.5	1	10	0.075	Exponential Filter + LS with Energy based Correction	Stalled
9D	Kranc	P1	-1.5	1	20	0.075	Exponential Filter + LS with Energy based Correction	Stalled

The HEG case was run in two batches: one with the exponential filter and the other without it. The cases without the exponential filter showed two different behaviours depending on the amount of AV.



(a) Temperature T in K. **(b)** Temperature T detail in K. **(c)** Temperature T_v in K. **(d)** Pressure in Pa.

Figure 7.4: HEG P1 run with AV Péclet 2 – temperature and pressure.

7.2.1. Runs Results and Analysis

The runs were run to either convergence, divergence or timeout. Upon timeout, the runs were restarted. These results are presented in table 7.1. Details regarding relevant details of each run are presented here.

The runs with Péclet 1 and 0.5 showed that they could run for thousands of iterations, diffuse the shock wave and stabilise the maximum smoothness factor. The ones with Péclet 2 or above had negative temperatures and diverged. Without the filter, temperature positivity is not addressed, as the LS method focuses on the densities' positivity. The smoothness factor was seen to increase to values above -1 for all of these simulations, before recovering to values bellow -1 for the simulations with Péclet 1 and 0.5.

Regarding the simulations 3C through 6C, the simulation 3C – the one with the highest AV – is here further analysed as it had the same behaviour as all the others with lower AV. The temperatures and pressure for this case are presented in figure 7.4. Negative temperatures and pressure are found just upstream of the shock wave. The simulations were found to diverge right after the negative temperatures were formed, suggesting that it drove the divergence. The shock is smoother looking than for the P0 case, possibly due to the effect of AV and due to the P1 representation being able to represent more smooth looking solutions.

Run 2C – with Péclet 1 – ran for 16811 iterations before diverging. No negative values were detected at runtime. The maximum value for the smoothness factor was found to be decreasing during the latter part of the run, with the last value being -1.45191. This smoothness factor is within reasonable values, being bellow 1. The residuals were found to mostly have a downwards trend, with the exception of the vibrational energy related residual (residual 14) and the one related with the mass conservation equation for O_2 (residual 6). Despite the low CFL, the last 4 residuals (momentum and energy equation residuals) were above 1 throughout the entire simulation. This shows how the high values of flux divergence require a low CFL in order for the simulation to advance at least in the beginning of the simulation.

Regarding the thermodynamic quantities, the temperatures and pressures are given in figure 7.6, together with electron density, as well as the smoothness factor. The temperatures were found to be

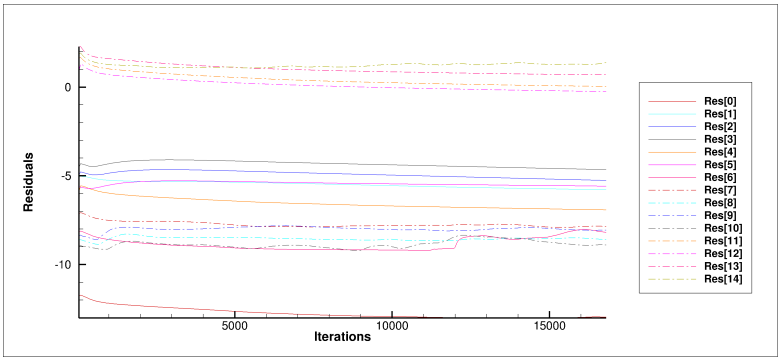


Figure 7.5: Run 2C residuals.

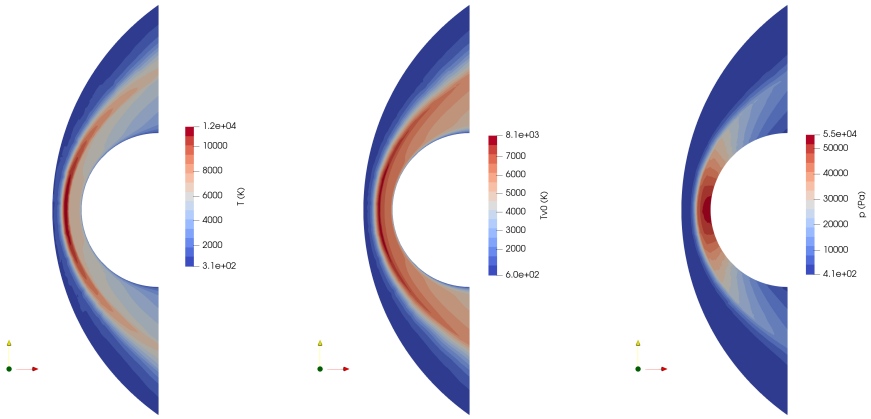
all positive. They show a clearly diffuse shock wave, with the ends of the shock wave being more diffused. The shock wave temperatures are bellow those simulated by Sharma, Giangaspero, Munafò, *et al.* [6], with temperature T being circa 12000 K instead of 14000 K, and temperature T_v being circa 8000 K instead of 9000 K. This simulation, then, has lower values than predicted by the FV simulation by Sharma, Giangaspero, Munafò, *et al.* [6], which is expected due to effect of AV lowering the maximum at the shock wave. Higher order or managing to use lower AV might improve this result.

The smoothness factor is clearly higher near the shock wave, as seen in figure 7.6e. It is near -1.5, which is reasonable for a P1 simulation. The smoothness is lower than -10 all across the domain. The smoothness is measured based on the pressure values, which are given in figure 7.6c.

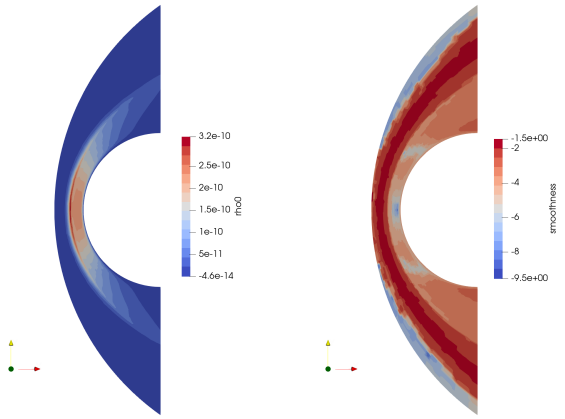
Ionisation is easily seen behind the shock wave, as in the P0 case, with electrons forming. This is seen in figure 7.6d. This is an expected behaviour. The electron density lower bound is negative, as seen from the colour bar. This negative density was not notified during run time, implying it is a result from the last time step whereupon divergence took place. Had the new spatial discretisation loop started, the LS method could possibly have handled this positivity issue.

The divergence of this simulation does not seem to have been driven by positivity issues, based on the given analysis, as no run time negative values were found. The AV is found to be smoothing the shock wave and stabilizing the maximum smoothness factor, suggesting that it is working as intended. The residuals were found to be mostly becoming smaller, with the exception of residual 6 and 14. However, despite this clear improvement in relation to the P1 simulations presented in section 5.5.2, divergence was still found. The given analysis cannot confirm or strongly suggest one single explanation. The residuals above 1 with such a low CFL might suggest that the CFL is still too high for certain parts of the simulation. The very diffuse shock wave could suggest that the AV is too high, leading to too much smoothing and affecting the solution or simply having too large diffusive fluxes. However, as seen from run 3C, the high AV is required to avoid a negative temperature for these conditions. As such, if the divergence is AV driven in some way, then a temperature positivity method, such as the use of the exponential filter, would be required.

Regarding the HEG case simulations with the exponential filter, divergence was found in all of them. A similar behaviour was found in all of them upon visualisation of the scalar fields. The case for Péclet 0.5 – run 1D – is here further explored, as all cases had a similar behaviour. The residuals are presented in figure 7.7. The residuals clearly grow over the 200 iterations before the simulation diverging. The momentum and energy related residuals are all above 0, showing a very unsteady behaviour where the



(a) Temperature T in K for run 2C. **(b)** Temperature T_v in K for run 2C. **(c)** Pressure in Pa for run 2C.



(d) Electron density in kg/m^{-3} for run 2C. **(e)** Smoothness factor for run 2C.

Figure 7.6: Temperature, pressure, smoothness and electron and NO^+ densities for run 2C (the HEG P1 Péclet 1 simulation). All colour bars span the interval from the maximmum to the minimum value.

divergence of the fluxes and the source terms have high values. This is in accordance with divergent behaviour. It is also in agreement with the increase in temperature in localised areas in the simulation, as seen in figures 7.8 and 7.9, as the continuous growth of these areas implies the existence of fluxes or source terms to maintain it, corresponding to not small residuals.

Figure 7.8 concerns the temperatures, the pressure, the electron density and the smoothness fact present in the simulation The temperatures spike right on the shock wave, with T growing to 35000 K and T_v growing to 20000 K. These values are larger than the ones given by Sharma, Giangaspero, Munafò, *et al.* [24]. The values peak in specific spot rather than along the shock wave. Both of these observations seem to indicate that these temperatures are numerical products rather than physical values. This behaviour is distinct from the case with just the LS positivity.

Pressure and electron density both appear with close values and distribution for the other LS positivity cases, showing a small evolution from the initial solution. The filter strength, as seen from figure 7.8e, is maximum near the inlet and exhibits non zero values immediately upstream of the shock wave. Both of these seem to indicate that numerical oscillations are leading to negative densities upstream of the shock wave, requiring the activation of the filter. Both areas where the filter is active concern zones where a jump is present. Near the shock, the spatially fast increase in quantities can lead – as already mentioned in this document – to oscillations due to the Gibbs phenomena, leading to negative densities. Near the

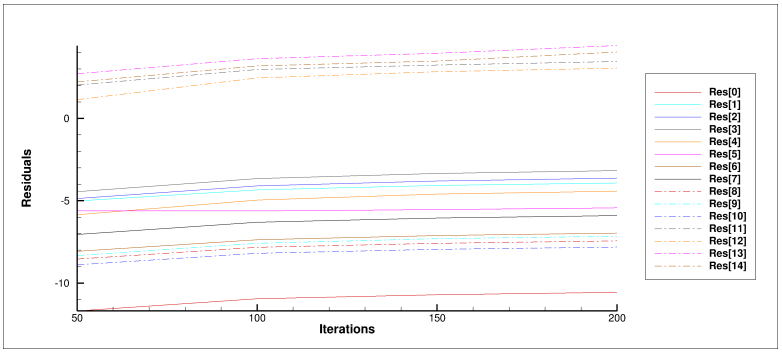
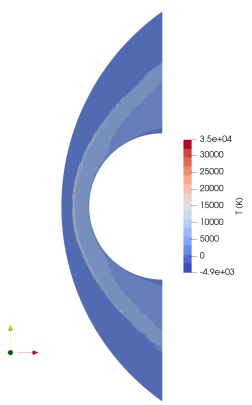
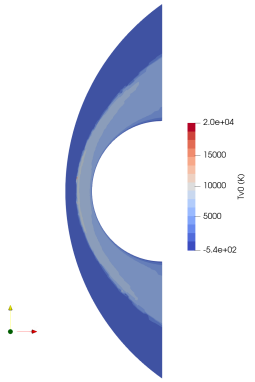


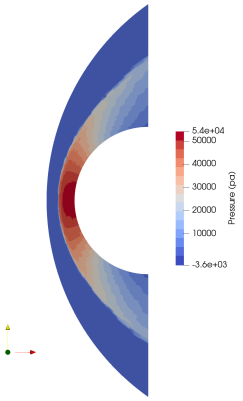
Figure 7.7: Run 1D residuals.



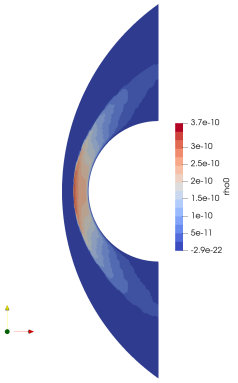
(a) Temperature T in K for run 1D.



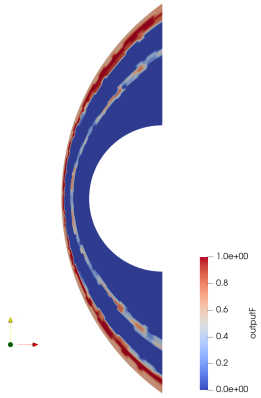
(b) Temperature T_v in K for run 1D.



(c) Pressure in Pa for run 1D.



(d) Electron density in kg/m^{-3} for run 1D.

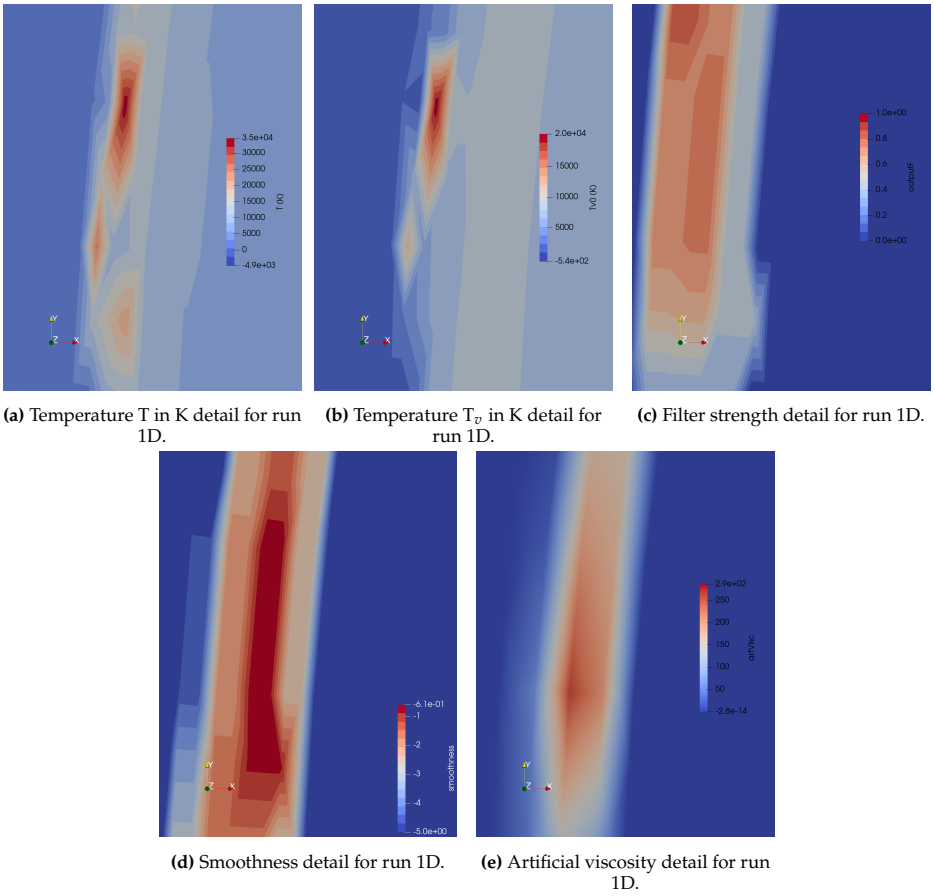


(e) Filter strength factor (named outputF) for run 1D. Maximum strength is obtained for when outputF is 1.

Figure 7.8: Temperature, pressure, smoothness and electron and NO^+ densities for run 1D. All colour bars span the interval from the maximum to the minimum value.

inlet, given that it is a supersonic inlet, quantities are fixed through Dirichlet boundary conditions, not allowing any oscillations to further lower the value there, creating a dip right upstream of the inlet where very low densities can become negative, requiring activation of the filter. The filter is, then, active in the expected places.

Figure 7.9 shows the place where the temperatures are seen to peak. Both T and T_v peak in the same place. This place is under artificial viscosity, as seen in figure 7.9e. In accordance to the AV placement and the sharp rise in temperature, this place has high smoothness fact, nearing -0.61. This high value



(a) Temperature T in K detail for run 1D.

(b) Temperature T_v in K detail for run 1D.

(c) Filter strength detail for run 1D.

(d) Smoothness detail for run 1D.

(e) Artificial viscosity detail for run 1D.

Figure 7.9: Temperature, smoothness, filter strength and artificial viscosity detail run 1D. All colour bars span the interval from the maximum to the minimum value. The detail focuses on upstream of the shock wave, where the temperatures have grown to large values.

seems to suggest that the simulation diverged due to this temperature rise. The filter is seen to be active right in front and on top of the zone with high temperature. The value for the filter on the node with the peak in temperature is 0.442379. This means the filter is at less than half the strength, but still active in this region.

The behaviour in this simulation is distinct from that of the without the filter, having for the same Péclet a rise in temperature near the shock wave unseen before. The place whereupon this growth is seen coincides with the area where the filter is activated. The filter, then, is, for this case, influencing the simulation leading to divergence. The filter being used is exactly the same implementation as for the Wedge case, as the filter definition is given in the BaseFiltering class of the code, as already discussed. An implementation error should not, then, be at fault for the running of the filter. The location of the filter is also sensical, as already discussed. Thus, the behaviour seems to be more driven by some interaction in the simulation rather than a incorrect implementation in the required filtering checks.

The filter itself might at fault and not be adequate for thermochemical equilibrium cases. Another probable cause is the interaction with the AV. The work of Dzanic and Witherden [73] makes use the filter with the use of order blending rather than AV. The proximity between the AV and the place where the filter is active – which is to be expected, given that both are highly related to places where shocks occur – would foment whatever negative interaction might be present.

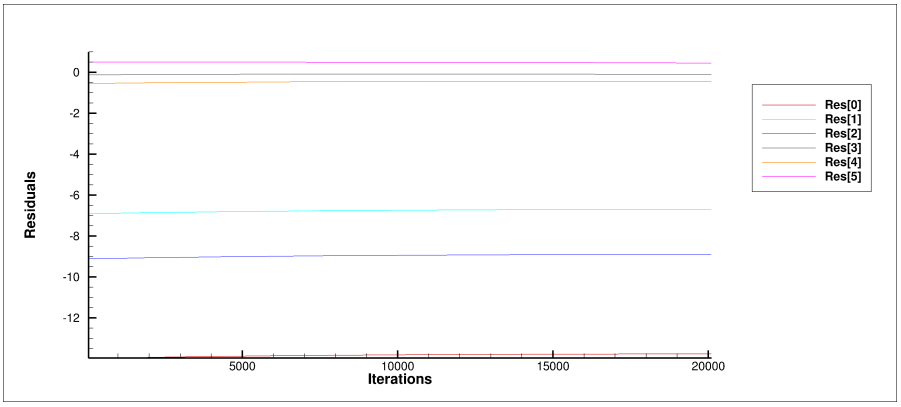


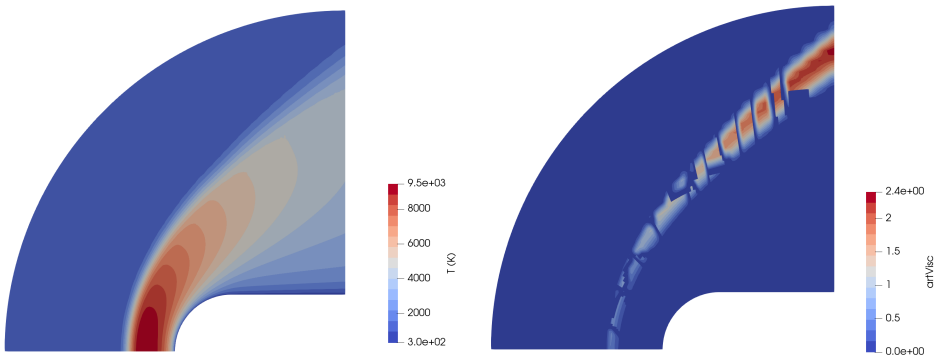
Figure 7.10: Residuals for the 8D run after restart

Given the failure in convergence of all HEG cases, the validations based on wall quantities is disregarded, as the cases are not yet valid. Upon convergence, the same validation procedure as done for P0 can be executed.

The Kranc case concerned a different behaviour. The simulations for runs 8D and 9D, effectively, stalled. Run 7D diverged. Figure 7.10 shows the residuals of the simulation 8D after a restart. The original run ran for 20000 iterations before timing out. The simulation, then, ran for over 40000 iterations. A negative slope is found for the higher residuals residuals, but due to its low value, the simulation could only be converged in an impractical number of iterations. Focus on run 8D is here taken, as run 9D has very similar results.

In figure 7.11 the temperature and the AV for the run 8D are plotted. Both quantities outline the shock wave. This shock wave is smoother than its P0 counterpart is found. AV is placed on top of the shock wave, in agreement with the expected behaviour. The positivity is not active at the end of the simulation, as expected. Still regarding the shock wave, the stand off distance between circa 19 and 11 mm – with the shock spanning that distance. The inner boundary of the shock now has a ≈11.37% which is larger than for the P0 case. However, given that the shock wave is thinner, it might be indicative that the previous value was due to over diffusion of the shock. Furthermore, given the stalled residuals, it might be that the shock wave is attempting to move to position, given that steady state has not been achieved yet. Thus, for not yet steady solution, the shock wave is reasonably close, possibly moving to position, or requiring a higher order simulation in order to do so. This stand off distance is taken from the stagnation point to the shock. Higher diffusivity of the shock is found away from the body, on the ends of the shock.

A rise in temperature is found behind the shock wave, as expected. The species densities, much like the P0 simulation, are higher at the wall, as seen in figure 7.12. Attempts were made to restart the simulation with higher CFL. Any CFL higher than 0.1 lead to divergence, irrespective of positivity scheme. The positivity tested was the exponential filter with the LS positivity and the energy based correction, just the LS positivity and the energy correction, and both of these without the energy based correction; all of these schemes were tested both taking and not taking the entropy constraint into account. In all cases, high temperatures developed downstream of the shock wave, but away from the wall, leading to divergence. This behaviour might indicate stiffness of the system, as this zone is reactive, or further not yet found problems with the implicit method.



(a) Temperature T in K for the Kranc P1 simulation with Péclet 20 – run 8D.

(b) AV strength the Kranc P1 simulation with Péclet 20 – run 8D.

Figure 7.11: Temperature and AV strength for the Kranc P1 simulation with Péclet 20 – run 8D.

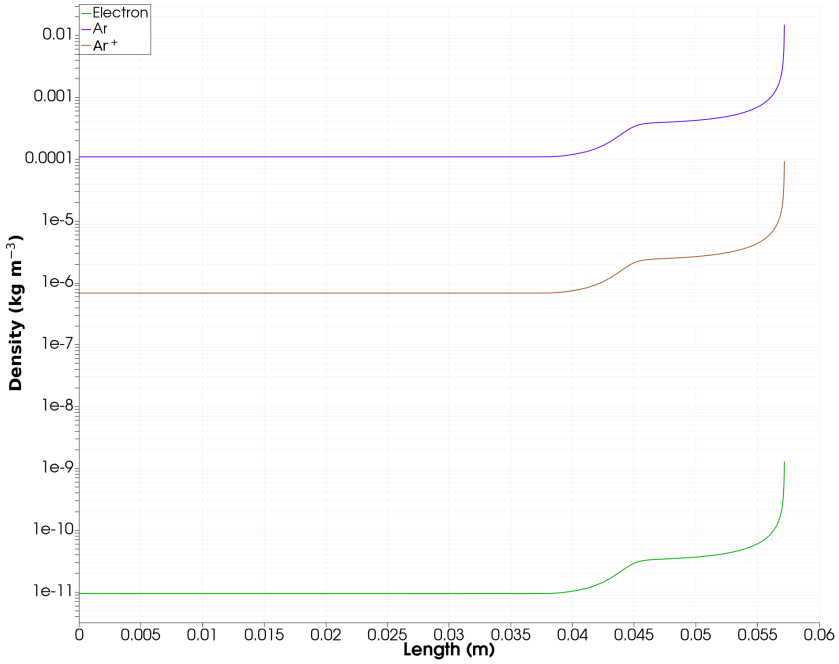


Figure 7.12: Species densities for run 8D along the stagnation line.

7.2.1.1. Conclusion

The P1 cases were found to not yet be able to fully converge. The positivity methods used both allowed the positivity of the solution for the variables for which they were designed. As such, the LS method allowed for the positivity of the densities, while the exponential filter together with the LS method allowed for the positivity of the densities and temperatures.

The LS method was found to not be able to be used on its own for the HEG case, due to negative temperatures existing for cases with not high enough AV. The cases with high enough AV were found to diverge later. For these latter cases, the AV managed to successfully diffuse the shock wave. The reason for divergence might be due to excessive diffusivity, but it could not be confirmed, as lower AV simulations do not currently advance enough in the simulation.

The HEG cases with filtering were found to be diverging. The temperatures, in these cases, grew in the

zones just upstream of the shock where filtering was active. They did not grow as large in other areas where filtering was active. The AV was found to be correctly placed on the shock wave. The filtering was done right in front of the shock wave and near the inlet, which are both zones where numerical oscillations due to the shock might cause species' density related positivity issues. A growth of the residues was also present in all of the cases. As such, high fluxes which drove the temperatures up were found to be present, and possibly of numerical nature. This behaviour was unforeseen, and possibly specific to the thermochemical nonequilibrium cases.

The Kranc case was found to be able to run with AV appropriate for P1 – Péclet 20 – and to improve in terms of shock wave position and shape in comparison to its P0 counterpart. The simulation could not be ran to convergence, as the residuals showed small variation over the course of 20000 iterations at the current CFL level. Higher CFLs were attempted, but a CFL over 0.1 was found to lead to divergence. This was linked to possibly a stiffness problem of the equations or to possibly further issues with the implicit method used.

These simulations showed an improvement on the past P1 simulations, allowing for the functioning of the AV and a successful verification of the use of the positivity methods in their task of maintaining positivity. However, the filtering method was found to interact with the solution in a negative way, which is found to be a new negative phenomena without clear cause. Both these success and new problems constitute one further step in the path to running higher order methods with thermochemical nonequilibrium successfully.

7.3. Answers to Research Questions

Given the present results, the research questions can be answered for the present state of the implementation of the positivity methods, together with the alterations done to the AV code. Regarding the *"Can the least squares method guarantee positivity of the densities during FR simulations for the thermochemical nonequilibrium case?"* question, it can be answered that for all cases tested – the HEG and Kranc case – it was answered positively. However, given that these cases cannot yet be ran to convergence in a high enough order to be appropriately validated, this answer cannot be answer positively in full; rather, it can be argue that the current path of research suggests that the method can indeed work as intended. Thus, this question must still be a target of further discussion.

Regarding the *"Can the exponential filter together with the least squares method guarantee positivity of the densities and temperatures during FR simulations for the thermochemical nonequilibrium case?"* question, it can be answered that for the cases tested – the HEG and Kranc case – the answer is answered positively in a very narrow sense. However, in the context of a successful CFD simulations, this question cannot be readily answered. The cases were not yet fully driven to convergence and validated to the desired degree and hence cannot yet be used to answer this answer with desired rigour. They seem to suggest so far that the methods can hold the values positive.

The latter question did not take into account negative effects of the methods in the solution. However, a method which prevents the convergence of the simulation, possibly driving instabilities, cannot be used. Per consequence, despite the filtering working in terms of maintaining positive temperatures, it must be further studied also in the context of its interaction with the simulation. The analysis given in the preceding section effectively promotes the idea that the filtering method has an interaction in the HEG case which is driving the divergence.

Thus, these questions are so far suggested to be answered positively, but demanding further research. This further research seems to not only encompass the simple inclusion of the methods into the current code base, but analysing the current interactions. Furthermore, given the already required analysis and extension of the base code given in chapter 6, it could be that further extensions are required in this code base rather than in the positivity method. Thus, the current answers to these research questions further motivate future work.

7.4. Conclusion

As a way to validate the use of the new setup for the Newton Iterator which was argued in chapter 6, the P0 HEG case simulation tackled in chapter 5.3 was run with this integrator. The run was successfully, giving the same result as the previous run with forward Euler. Thus, it was concluded that the correct way to run P0 TCNEQ simulations in COOLFluiD was found.

As a continuation of the initial study on enabling positivity with flux reconstruction, new P1 runs for the two selected cases – the HEG and the Kranc case – were selected. These runs proceed from code alterations in order to enable a AV for these cases as well as a change in the case setup in order to enable an adequate time integration algorithm.

The runs focused on testing the use of the filtering and the LS positivity techniques for the two cases chosen. A wide range of AV strength was chosen, as it is a user tuned parameter with effects on both the running of the simulation as well as on the final converged solution. The CFL was kept small in order to not drive diverge due to an inappropriate CFL.

All HEG case runs were found to diverge regardless of positivity method. The AV was found to be active on the shock wave and confirmed to diffuse the shock wave; the cases without the filter being active and higher AV were seen to have this behaviour. This showed an improvement in relation to previous simulations, where the AV was not diffusing the shock wave appropriately, and also a success of the code alteration given in chapter 6.

The high AV was found to be a possible cause for the divergence of the simulations without the filter active. Those simulations, with lower AV, were found to instead diverge due to a negative temperature, which is not accounted for in the LS based method, as such positivity constraints were the goal of the filter to maintain, and not of the LS method. No negative densities were found during run time, and the LS method was not found to be the cause of divergence of the simulation.

The HEG runs with the exponential filter active were found to have no negative temperatures. However, a growth of the temperatures was found in zones of filtering. These increases in temperature were accompanied with growing residuals. This behaviour was linked to the divergent behaviour of the simulation. The filtering was found to be correctly placed, as well as the AV. The filtering method was not considered to be at fault due to miss implementation since it was the same implementation already tested for the Wedge case. A case specific interaction was considered to be more likely.

The Kranc case was found to effectively stall. While it was also considered that the simulation might eventually converge, the small slopes of the residuals seem to suggest that the simulations is for practical running purposes stalled at that CFL. The CFL could not be increased above 0.1 lest the simulation diverge – given any positivity method – showing a hard CFL constraint.

All positivity methods worked to ensured the positivity constraints for which they were designed. The

LS method was found to be unsuitable to be used on its own for the P1 simulation; which is not what the method was designed for, as it does not ensure temperature positivity. The filtering method was found to have a case specific problem which, despite ensuring positivity, also drives divergence. This was found to require further study.

8

Future Work

Based on the current achievements and findings of the current work, as well as the goals given by the current research questions, future work may be defined. The current research focused on running the implemented positivity methods on selected cases for testing them. Success was found in keeping the variables positive, but lacked in converging the simulations fully. Immediate future work should focus on enabling this convergence in order to fully test the positivity methods.

Given that it was found that a positivity method is required for temperature for P1 simulations – that is, the LS method does not suffice – further focus should placed on studying cases with the use of the filter for positivity. The current two test cases can keep being taken basis for the research. The current behaviour of the fluxes in the HEG case with filtering requires further investigation, namely in terms of what is driving that behaviour.

Given that the exponential filter was presented by Dzanic and Witherden [73] with order blending rather than AV in order to enable P1 and higher simulations, if the the behaviour of the filter is found to be related to the AV in a way that cannot be fixed, implementing and making use of order blending instead might improve the usability of the filter. Furthermore, even if the filter can be used with AV, the avenue of flux order blending, given the success by Dzanic and Witherden [73] would also be of interest for future work as possible alternative to AV in COOLFluiD. It could improve the shock wave definition and, possibly, robustness. However, such cannot be argued in advanced based on the present knowledge.

Regarding further testing the implementation with AV, a model simplification might help identify the specific interaction with the filter that is running the divergence. As such, redoing the HEG case in a single temperature formulation, as well as redoing it without chemical source terms might help bridge the differences between the successful test with the Wedge case (Euler model) and the current HEG model.

The finding of these model reductions would lead to possibly either altering the filter formulation and acting, or finding that the filter is not at fault here. If the interaction disappears for the single temperature – which can be expected since it is not currently seen in the Kranc case – it might indicate

that the filter is usable for those cases, but requires further investigation into how to be formulated or how to act upon the more complicated model.

If the problem persists, alternate positivity method might have to be considered. Returning to a limiting approach for temperature (or pressure) might suffice, given that the main issue found with the limiting approach was how it treated densities. This would decouple fully the positivity of densities and temperatures and might have negative effects on the solution. Furthermore, it does not take into account the oscillatory behaviour in the same way as a filter, acting solution point wise. However, it would avoid all the filter related characteristics which might be interacting incorrectly with the solution. Given its presence already in COOLFluiD's code, an extension to multiple temperatures and the removal of density limiting – opting instead for the current LS method – could suffice. This work would be non trivial, as the method is not readily available in the correct format, but it rises as a possible alternative.

Once the current cases have been used successfully to answer the research questions, in case the positivity methods work (has it has been suggested so far), the research should be extended to more cases to further test and possibly strengthen the confidence in the use of these methods for positivity for thermochemical nonequilibrium FR simulations in COOLFluiD. These possible cases have already been tackled and motivated in section 2.6. All these cases would have to be converged spatially and residual wise and validated accordingly to their reference data, be it the shock wave stand off distance, wall quantities such as pressure, heat flux or drag, or possibly L1 and L2 norms of scalar field quantities.

If the positivity method is found to succeed in the future, entropy stable or entropy conservative fluxes – as tackled in section 2.3.4.2 – might serve as a future development for COOLFluiD. However, these would require fundamental work in developing the expressions for the AV fluxes for the reacting case, as well as implementation work that surpasses the FR discretisation part of the code. It would require implementing not only the entropy fluxes, but also the variable set, the required transformations between variable sets, and adding any checks to the FR discretisation which might be required. Further work might be required in order to fully couple any new variable set. Thus, this work, given its complexity, remains as a possible future work based on literature for after the current issues with positivity have been tackled.

9

Conclusion

An extensive research regarding the use of positivity enabling techniques in FR simulations with thermochemical nonequilibrium was presented. Motivated by a literature review, the problem of positivity in higher order simulations – such as those with FR – was identified as a current area of research where targeted techniques are currently developed and used. The literature review also focused on the current on going efforts to extend FR in COOLFluiD to include thermochemical nonequilibrium. However, the positivity method in use in COOLFluiD does not suffice in maintaining positivity of the chemical species densities for the cases with thermochemical nonequilibrium, not allowing the simulations to run. Thus, it was identified a clear research gap: positivity enforcement in COOLFluiD for FR simulations with thermochemical nonequilibrium.

From the surveyed methods, it was identified that filtering – namely with the entropy based exponential filter– could improve upon the current in use method for positivity, having been used successfully for a wide variety of 1D and 2D cases in supersonic regime. However, it was also found, from the literature review, that this filtering method – as well as all positivity methods found in literature – still suffered from the same base constraints as the in use method for COOLFluiD, making it unfit on its own for the problems detected for thermochemical nonequilibrium cases. Thus, it was concluded that a new positivity method focused on densities was required.

Given on going efforts within the current research group to explore the filtering alternative, the entropy based exponential filter was implemented making use of in development code for the implementation of the filter which was, thus, completed. This in development code focused on creating a fundamental general class for filtering as well as a specific inherited class for the Euler equations physical model. The entropy based exponential filter was thus implemented, based on literature.

Given the interest in filtering, further filters were procured, allowing for not only alternatives, but testing the modularity of the filtering implementation, as well as allowing for the study of how different filters might impact differently the solution. The convolution based Fejér filter was chosen given its positive properties for Gibbs effect reduction. It was found also, during this search, that the exponential filter also has these properties, according to literature. A derivation for convolution filters for FR for 2D

structured grids was done, which is in general valid for convolution filters, rather than just for the Fejér filter. This derivation made use of the properties of FR in order to simplify the convolution expression. It, thus, concerns a new development of filtering methods for FR.

The Fejér filter was implemented into the same class as the exponential filter, also making used of the entropy considerations, with the care to give the user the option to choose which one to be used for the simulation. This focus on modularity allows for further developments or studies with other methods in the future.

The Fejér filter, the exponential filter and the previous method being used in COOLFluiD were tested against the Wedge case, consisting of an inviscid case where upon air impacted a wedge. The case was chosen based on the several shock reflections present, which gave rise to positivity issues. Research questions regarding the effect of the filter on the running of the simulation and on the final solution were posed. It was found that all method could, under certain AV conditions, allow for the simulation to fully converge; in which case the solution was the same regardless of filtering used. However, if the AV was not strong enough, the Fejér filter was found to diverge where the exponential filter worked. The wedge simulations were target of a mesh refinement study based on shock wave position in order to show that all methods allowed for full convergence.

Further testing was done by running the Wedge case without AV for all the filtering methods. The exponential filter was found to have a similar behaviour to the previous in place method, while the Fejér filter was found to diverge. The positivity methods were, then, found not to have all the same robustness. The exponential filter was found to be the appropriate substitute for the previous method rather than the Fejér filter. Despite being a filtering method, the Fejér showed worse results than the limiting approach, revealing that filtering is not necessarily better than limiting, despite its direct approach to oscillations.

Given the success in implementing the exponential filter approach for the Euler case, this integration and implementation knowledge was transferred to the implementation for the multispecies and multitemperature case, which required the creation of a new filtering class. Given the use of PLATO for the calculation of the energy quantities for the simulation, the entropy had to be re-derived for these cases making use of the literature regarding PLATO for the required physical models. Pressure was also derived for the multitemperature case. The filter was, thus, implemented into a new class with the new positivity checks based on the derived physical quantities for the new physical model.

Given the need for a positivity method for the densities identified during the literature review, a new method based on the analytical solution to a constraint least squares problem was derived. It was found that a weighted average approach together with the overriding of certain negative densities was the solution to this problem. A method for going over the set of possible solution was presented. This method was integrated into the filtering class together with the exponential filter for the multispecies and multitemperature model. A temperature correction based on the energy was included in order to take into account the alteration of the densities.

The LS method, unlike the filter, was created to also run for P0 cases. Thus, this method was tested with P0 simulations. The simulated cases – the HEG and the Kranc case – were chosen based on the literature review with the goal of showing cases with a variety of species sets, reacting flow and a detached shock wave that could induce positivity issues. The cases setup was presented, with the physical models, boundary conditions and initial conditions justified based on the literature surrounding the case. The setup was presented with the goal of allowing reproducibility with COOLFluiD. The simulations were

run to residual wise convergence. The mesh was created with a purposely coarse mesh around the shock with the goal of restarting the simulations in a higher order than P0. Furthermore, the FR code is not optimised to run meshes as thin as FV ones, which would be required to fully converge a P0 simulation. The current approach was deemed appropriate for FR studies.

The HEG case saw a good agreement in terms of pressure distribution on the wall and shock wave position in comparison with the validation data. The heat flux at the wall was found to be over estimated. The Kranc case was found to have an overly diffused shock wave, possibly due to numerical diffusion. Both simulations were found to have a difference in species concentrations after the shock wave, showing the influence of the reacting terms. Thermal nonequilibrium was found for the HEG case at the shock wave, as expected. The cases were, thus, successfully ran in P0 with room for improvement for higher orders.

The setup for the P1 cases was presented. The cases were restarted in P1 from their P0 solutions. A batch of runs was executed based on varying AV parameters. All simulations diverged. It was identified that the AV seemed to be having a small effect on the solution, with little visible diffusivity at the shock wave. The AV was found to be correctly placed. A stringent CFL constraint was also found. This findings led to the conjecture that the AV and the implicit method might be influenced by problems in the code, namely due to lack of extension to this new cases.

An inspection of the code revealed that the formulation used for the time integration in the P1 simulations was not valid and could not be used due to the setup making use of primitive variables as the variables saved at solution points. The integration method was changed accordingly to the Newton Iterator method, as it is dubbed in COOLFluiD. It was also found that this method could not run with part of the configuration used in the previous setup, required said configuration to be turned off. Regarding the AV, it was found that the gradient computation, the gradient interface correction, the AV interface flux and the perturbed gradients for the calculation of the Jacobian had not been adequately extended to these new cases, namely due to the use of primitive variables in the solution points, which was not foreseen and implicitly dismissed as a possibility. This code was corrected and extended, requiring a rewriting of the code for these new cases.

The P1 simulations were repeated. Furthermore, the HEG case was also tested with just the LS method. It was found that the HEG case required a temperature positivity method and, thus, cannot be ran with just the LS method. The AV was found to now diffuse the shock wave and to have a meaningful impact on the simulations, actively impeding numerical oscillations, as seen in simulations with high values of AV. The filter was found to have a negative interaction in the HEG case, leading to increased residuals and a rise in temperature, namely T_v, just upstream of the shock. The exact cause could not be identified, but it was conjectured that an interaction of the filter with the AV or with the physical model could be at fault. Given that the filter works well with the Wedge case, the implementation was not considered to be at fault.

The Kranc case was found to stall the residuals and still have a CFL constraint at 0.1 irrespective of the positivity method used. The the shock wave was found to become thinner than its P0 counterpart, having the shape much better defined. All simulations had the shock wave AV on the shock wave and nowhere else.

These cases were found to be a large improvement from the initial status before this work. Positivity has been found for P0, with room for further validation. For P1, on going research is focusing on the use of

the exponential filter. This research also enabled further extension of COOLFluiD's AV to these cases, which was unknown to be incomplete. The use of the time integrator for COOLFluiD for these cases was also well defined.

Future research was found to mainly focus on enabling the P1 simulations before moving to higher order ones. This focus should be on fully checking if the filter can be used with the current models, and gathering alternatives in case it cannot. Long term future research focuses on fully validating these cases with appropriate positivity methods at higher orders, extending the study to more cases of interest identified in the literature review, as well as implementing further robustness methods identified in the literature review, namely entropy fluxes configurations.

The current work allowed for developing and implementing new methods for positivity, including an original contribution, as well as identifying and correcting implemented limitations of the code. The simulations are on the path for further studies to be fully converged, with a solid ground for future studies, and possibly enabling FR with thermochemical nonequilibrium.

References

[1] H. T. Huynh, "A Reconstruction Approach to High-Order Schemnes Including Discontinuous Galerkin for Diffusion," en, in *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, Orlando, Florida: American Institute of Aeronautics and Astronautics, Jan. 2009, ISBN: 9781600869730. DOI: 10.2514/6.2009-403. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2009-403 (visited on 10/29/2024).

[2] R. Vandenhoeck, "Towards massively parallel and robust high-order methods for transitional hypersonic flow modelling on unstructured grids: Application to reusable launcher stages," eng, Ph.D. dissertation, KU Leuven, 2022.

[3] F. B. Ameur, R. Vandenhoeck, and A. Lani, "High-order flux reconstruction scheme for thermochemical nonequilibrium high-speed flows," in *AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, 2019. DOI: 10.2514/6.2019-1391. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2019-1391. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2019-1391.

[4] A. Lani, *Coolfluid wiki*, GitHub, Feb. 2024. [Online]. Available: https://github.com/andrealani/COOLFluiD/wiki (visited on 10/28/2024).

[5] A. Lani, "An object oriented and high performance platform for aerothermodynamics simulation," Ph.D. dissertation, Université Libre de Bruxelles, 2008.

[6] V. Sharma, V. F. Giangaspero, A. Munafò, S. Poedts, and A. Lani, "Validation and Verification of the Implicit Thermo-Chemical Non-Equilibrium CFD Solver in COOLFluiD with PLATO," in *AIAA Scitech 2024 Forum*. American Institute of Aeronautics and Astronautics, 2024. DOI: 10.2514/6.2024-2728. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2024-2728 (visited on 10/27/2024).

[7] J. Anderson, *Hypersonic and high-temperature gas dynamics*. American Institute Of Aeronautics and Astronautics, Inc, 2019.

[8] R. N. Gupta, J. M. Yos, R. A. Thompson, and 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," *ntrs.nasa.gov*, Aug. 1990. [Online]. Available: https://ntrs.nasa.gov/citations/19900017748.

[9] D. Bose, J. L. Brown, D. K. Prabhu, P. Gnoffo, C. O. Johnston, and B. Hollis, "Uncertainty assessment of hypersonic aerothermodynamics prediction capability," *Journal of Spacecraft and Rockets*, vol. 50, pp. 12–18, Jan. 2013. DOI: 10.2514/1.a32268. (visited on 04/03/2022).

[10] C. Park, *Nonequilibrium hypersonic aerothermodynamics*. John Wiley & Sons, 1989. [Online]. Available: https://ntrs.nasa.gov/citations/19910029860 (visited on 10/27/2024).

[11] *Commercial crew program press kit - nasa*, NASA, Jun. 2024. [Online]. Available: https://www.nasa.gov/commercial-crew-program-press-kit/ (visited on 10/28/2024).

[12] B. Ridgeway, *Splashdown! nasa's orion returns to earth after historic moon mission - nasa*, NASA, Dec. 2022. [Online]. Available: https://www.nasa.gov/centers-and-facilities/hq/splashdown-nasas-orion-returns-to-earth-after-historic-moon-mission/.

[13] *Spacex*, SpaceX. [Online]. Available: https://www.spacex.com/launches/mission/?missionId=starship-flight-4.

[14] A. Viviani, A. Aprovitola, G. Pezzella, and C. Rainone, "Cfd design capabilities for next generation high-speed aircraft," *Acta Astronautica*, vol. 178, pp. 143–158, 2021, ISSN: 0094-5765. DOI: https://doi.org/10.1016/j.actaastro.2020.09.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576520305439.

[15] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems* (Cambridge Texts in Applied Mathematics). Cambridge University Press, 2002.

[16] R. Abgrall and C.-W. Shu, *Handbook of Numerical Methods for Hyperbolic Problems*. Elsevier, Nov. 2016.

[17] J. E. Marsden, L. Sirovich, and S. S. Antman, *Nodal Discontinuous Galerkin Methods : Algorithms, Analysis, and Applications.* Springer New York, 2008.

[18] R. Dhib, F. B. Ameur, R. Vandenhoeck, A. Lani, and S. Poedts, "Development of an implicit high-order flux reconstruction solver for high-speed flows on simplex elements," *Computer Physics Communications*, vol. 295, p. 109 006, 2024, ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2023.109006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S001046552300351X.

[19] M. Brchnelova, S. Poedts, and A. Lani, "Towards faster and more physical mhd and multi-fluid data-driven global coronal models," eng, Ph.D. dissertation, KU Leuven, 2024.

[20] *Orion Components - NASA*. [Online]. Available: https://www.nasa.gov/reference/orion-components/ (visited on 01/03/2025).

[21] A. Munafò and M. Panesi, "Plato: A high-fidelity tool for multi-component plasmas," in *AIAA AVIATION 2023 Forum*. American Institute of Aeronautics and Astronautics, 2023. DOI: 10.2514/6.2023-3490. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2023-3490. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2023-3490.

[22] A. Peyvan, K. Shukla, J. Chan, and G. Karniadakis, "High-order methods for hypersonic flows with strong shocks and real chemistry," *Journal of Computational Physics*, vol. 490, p. 112 310, 2023, ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2023.112310. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999123004059.

[23] D. S. Hoskin, R. L. Van Heyningen, N. C. Nguyen, J. Vila-Pérez, W. L. Harris, and J. Peraire, "Discontinuous galerkin methods for hypersonic flows," *Progress in Aerospace Sciences*, vol. 146, p. 100 999, 2024, ISSN: 0376-0421. DOI: https://doi.org/10.1016/j.paerosci.2024.100999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0376042124000253.

[24] S. Sharma, W. Huo, and C. Park, "The rate parameters for coupled vibration-dissociation in a generalized SSH approximation," en, in *23rd Thermophysics, Plasmadynamics and Lasers Conference*, San Antonio,CA,U.S.A.: American Institute of Aeronautics and Astronautics, Jun. 1988. DOI: 10.2514/6.1988-2714. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.1988-2714 (visited on 11/08/2024).

[25] E. Nagnibeda and E. Kustova, *Non-Equilibrium Reacting Gas Flows*. Springer Nature, Jan. 2009. DOI: 10.1007/978-3-642-01390-4. (visited on 10/27/2024).

[26] K. E. Niemeyer, N. J. Curtis, and C.-J. Sung, "Pyjac: Analytical jacobian generator for chemical kinetics," *Computer Physics Communications*, vol. 215, pp. 188–203, 2017, ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2017.02.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010465517300462.

[27] D. Knight, J. Longo, D. Drikakis, *et al.*, "Assessment of cfd capability for prediction of hypersonic shock interactions," *Progress in Aerospace Sciences*, vol. 48-49, pp. 8–26, 2012, Assessment of Aerothermodynamic Flight Prediction Tools, ISSN: 0376-0421. DOI: https://doi.org/10.1016/j.paerosci.2011.10.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0376042111001023.

[28] Z. Wang, K. Fidkowski, R. Abgrall, *et al.*, "High-order cfd methods: Current status and perspective," *International Journal for Numerical Methods in Fluids*, vol. 72, no. 8, pp. 811–845, 2013. DOI: https://doi.org/10.1002/fld.3767. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.3767. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.3767.

[29] H. Huynh, Z. Wang, and P. Vincent, "High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids," *Computers & Fluids*, vol. 98, pp. 209–220, 2014, 12th USNCCM mini-symposium of High-Order Methods for Computational Fluid Dynamics - A special issue dedicated to the 80th birthday of Professor Antony Jameson, ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2013.12.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045793013004829.

[30] W. H. Reed and T. R. Hill, "Triangular mesh methods for the neutron transport equation," Los Alamos Scientific Lab., N. Mex.(USA), Tech. Rep., 1973.

[31] D. A. Kopriva and J. H. Kolias, "A Conservative Staggered-Grid Chebyshev Multidomain Method for Compressible Flows," en, *Journal of Computational Physics*, vol. 125, no. 1, pp. 244–261, Apr. 1996, ISSN: 00219991. DOI: 10.1006/jcph.1996.0091. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999196900911 (visited on 10/30/2024).

[32] H. T. Huynh, "A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods," en, in *18th AIAA Computational Fluid Dynamics Conference*, Miami, Florida: American Institute of Aeronautics and Astronautics, Jun. 2007, ISBN: 9781624101298. DOI: 10.2514/6.2007-4079. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2007-4079 (visited on 10/29/2024).

[33] H. T. Huynh, "Collocation and Galerkin Time-Stepping Methods," en, in *19th AIAA Computational Fluid Dynamics*, San Antonio, Texas: American Institute of Aeronautics and Astronautics, Jun. 2009, ISBN: 9781624101373. DOI: 10.2514/6.2009-4323. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2009-4323 (visited on 10/30/2024).

[34] H. Huynh, "High-Order Methods Including Discontinuous Galerkin by Reconstructions on Triangular Meshes," en, in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Orlando, Florida: American Institute of Aeronautics and Astronautics, Jan. 2011, ISBN: 9781600869501. DOI: 10.2514/6.2011-44. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2011-44 (visited on 10/30/2024).

[35] Z. Wang, L. Zhang, and Y. Liu, "Spectral (finite) volume method for conservation laws on unstructured grids IV: Extension to two-dimensional systems," en, *Journal of Computational Physics*, vol. 194, no. 2, pp. 716–741, Mar. 2004, ISSN: 00219991. DOI: 10.1016/j.jcp.2003.09.012. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999103005035 (visited on 10/30/2024).

[36] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, "Uniformly high order accurate essentially non-oscillatory schemes, iii," *Journal of Computational Physics*, vol. 71, no. 2, pp. 231–303, 1987, ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(87)90031-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0021999187900313.

[37] G.-S. Jiang and C.-W. Shu, "Efficient implementation of weighted eno schemes," *Journal of Computational Physics*, vol. 126, no. 1, pp. 202–228, 1996, ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1996.0130. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999196901308.

[38] B. Cockburn and C.-W. Shu, "The runge–kutta discontinuous galerkin method for conservation laws v: Multidimensional systems," *Journal of Computational Physics*, vol. 141, no. 2, pp. 199–224, 1998, ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1998.5892. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999198958922.

[39] F. Bassi and S. Rebay, "High-order accurate discontinuous finite element solution of the 2d euler equations," *Journal of Computational Physics*, vol. 138, no. 2, pp. 251–285, 1997, ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1997.5454. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999197954541.

[40] F. Bassi and S. Rebay, "A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier–stokes equations," *Journal of Computational Physics*, vol. 131, no. 2, pp. 267–279, 1997, ISSN: 0021-9991. DOI: https://doi.org/10.1006/jcph.1996.5572. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999196955722.

[41] G. J. Gassner, A. R. Winters, and D. A. Kopriva, "A well balanced and entropy conservative discontinuous galerkin spectral element method for the shallow water equations," *Applied Mathematics and Computation*, vol. 272, pp. 291–308, 2016, Recent Advances in Numerical Methods for Hyperbolic Partial Differential Equations, ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2015.07.014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0096300315009261.

[42] Y. Lv and M. Ihme, "High-order discontinuous Galerkin method for applications to multicomponent and chemically reacting flows," en, *Acta Mechanica Sinica*, vol. 33, no. 3, pp. 486–499, Jun. 2017, ISSN: 0567-7718, 1614-3116. DOI: 10.1007/s10409-017-0664-9. [Online]. Available: http://link.springer.com/10.1007/s10409-017-0664-9 (visited on 10/29/2024).

[43] P. Fernandez, N. Nguyen, and J. Peraire, "The hybridized discontinuous galerkin method for implicit large-eddy simulation of transitional turbulent flows," *Journal of Computational Physics*, vol. 336, pp. 308–329, 2017, ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2017.02.015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999117301080.

[44] M. Yano and D. L. Darmofal, "An optimization-based framework for anisotropic simplex mesh adaptation," en, *Journal of Computational Physics*, vol. 231, no. 22, pp. 7626–7649, Sep. 2012, ISSN: 00219991. DOI: 10.1016/j.jcp.2012.06.040. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999112003749 (visited on 10/30/2024).

[45] M. Ceze and K. J. Fidkowski, "Anisotropic hp-Adaptation Framework for Functional Prediction," en, *AIAA Journal*, vol. 51, no. 2, pp. 492–509, Feb. 2013, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J051845. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.J051845 (visited on 10/30/2024).

[46] C. Klaij, M. Van Raalte, H. Van Der Ven, and J. Van Der Vegt, "H-Multigrid for space-time discontinuous Galerkin discretizations of the compressible Navier–Stokes equations," en, *Journal of Computational Physics*, vol. 227, no. 2, pp. 1024–1045, Dec. 2007, ISSN: 00219991. DOI: 10.1016/j.jcp.2007.08.034. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999107003762 (visited on 10/30/2024).

[47] S. M. Kast and K. J. Fidkowski, "Output-based mesh adaptation for high order Navier–Stokes simulations on deformable domains," en, *Journal of Computational Physics*, vol. 252, pp. 468–494, Nov. 2013, ISSN: 00219991. DOI: 10.1016/j.jcp.2013.06.007. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999113004300 (visited on 10/30/2024).

[48] B. Cockburn and C.-W. Shu, "The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems," en, *SIAM Journal on Numerical Analysis*, vol. 35, no. 6, pp. 2440–2463, Dec. 1998, ISSN: 0036-1429, 1095-7170. DOI: 10.1137/S0036142997316712. [Online]. Available: http://epubs.siam.org/doi/10.1137/S0036142997316712 (visited on 10/30/2024).

[49] N. Nguyen and J. Peraire, "Hybridizable discontinuous Galerkin methods for partial differential equations in continuum mechanics," en, *Journal of Computational Physics*, vol. 231, no. 18, pp. 5955–5988, Jul. 2012, ISSN: 00219991. DOI: 10.1016/j.jcp.2012.02.033. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999112001544 (visited on 10/30/2024).

[50] B. Cockburn, J. Gopalakrishnan, and R. Lazarov, "Unified Hybridization of Discontinuous Galerkin, Mixed, and Continuous Galerkin Methods for Second Order Elliptic Problems," en, *SIAM Journal on Numerical Analysis*, vol. 47, no. 2, pp. 1319–1365, Jan. 2009, ISSN: 0036-1429, 1095-7170. DOI: 10.1137/070706616. [Online]. Available: http://epubs.siam.org/doi/10.1137/070706616 (visited on 10/30/2024).

[51] N. Nguyen, J. Peraire, and B. Cockburn, "A class of embedded discontinuous Galerkin methods for computational fluid dynamics," en, *Journal of Computational Physics*, vol. 302, pp. 674–692, Dec. 2015, ISSN: 00219991. DOI: 10.1016/j.jcp.2015.09.024. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999115006178 (visited on 10/30/2024).

[52] J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods* (Texts in Applied Mathematics), J. E. Marsden, L. Sirovich, and S. S. Antman, Eds. New York, NY: Springer New York, 2008, vol. 54, ISBN: 978038772065. DOI: 10.1007/978-0-387-72067-8. [Online]. Available: http://link.springer.com/10.1007/978-0-387-72067-8 (visited on 10/30/2024).

[53] F. B. Ameur, "Development of r-adaptive flux reconstruction algorithms for high-speed flows with uncertainties," eng, Ph.D. dissertation, KU Leuven, 2023.

[54] P. E. Vincent, P. Castonguay, and A. Jameson, "A New Class of High-Order Energy Stable Flux Reconstruction Schemes," en, *Journal of Scientific Computing*, vol. 47, no. 1, pp. 50–72, Apr. 2011, ISSN: 0885-7474, 1573-7691. DOI: 10.1007/s10915-010-9420-z. [Online]. Available: http://link.springer.com/10.1007/s10915-010-9420-z (visited on 11/12/2024).

[55] P. Castonguay, P. Vincent, and A. Jameson, "Application of High-Order Energy Stable Flux Reconstruction Schemes to the Euler Equations," en, in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Orlando, Florida: American Institute of Aeronautics and Astronautics, Jan. 2011, ISBN: 9781600869501. DOI: 10.2514/6.2011-686. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2011-686 (visited on 11/12/2024).

[56] P. Castonguay, D. Williams, P. Vincent, M. López, and A. Jameson, "On the Development of a High-Order, Multi-GPU Enabled, Compressible Viscous Flow Solver for Mixed Unstructured Grids," en, in *20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii: American Institute of Aeronautics and Astronautics, Jun. 2011, ISBN: 9781624101489. DOI: 10.2514/6.2011-3229. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2011-3229 (visited on 11/12/2024).

[57] R. Vandenhoeck and A. Lani, "Implicit High-Order Flux Reconstruction Positivity Preserving LLAV Scheme for Viscous High-Speed Flows," en, in *AIAA Scitech 2019 Forum*, San Diego, California: American Institute of Aeronautics and Astronautics, Jan. 2019, ISBN: 9781624105784. DOI:

10.2514/6.2019-1153. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2019-1153 (visited on 10/30/2024).

[58] E. J. Ching, Y. Lv, P. Gnoffo, M. Barnhardt, and M. Ihme, "Shock capturing for discontinuous galerkin methods with application to predicting heat transfer in hypersonic flows," *Journal of Computational Physics*, vol. 376, pp. 54–75, 2019, ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2018.09.016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999118306107.

[59] M. J. Zahr and J. M. Powers, "High-Order Resolution of Multidimensional Compressible Reactive Flow Using Implicit Shock Tracking," en, *AIAA Journal*, vol. 59, no. 1, pp. 150–164, Jan. 2021, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J059655. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.J059655 (visited on 10/30/2024).

[60] T. Huang and M. J. Zahr, "A robust, high-order implicit shock tracking method for simulation of complex, high-speed flows," *Journal of Computational Physics*, vol. 454, p. 110 981, 2022, ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2022.110981. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999122000432.

[61] J. S. Park and C. Kim, "Higher-order multi-dimensional limiting strategy for discontinuous galerkin methods in compressible inviscid and viscous flows," *Computers & Fluids*, vol. 96, pp. 377–396, 2014, ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2013.11.030. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045793013004842.

[62] H. You and C. Kim, "High-order multi-dimensional limiting strategy with subcell resolution I. Two-dimensional mixed meshes," en, *Journal of Computational Physics*, vol. 375, pp. 1005–1032, Dec. 2018, ISSN: 00219991. DOI: 10.1016/j.jcp.2018.09.011. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999118306053 (visited on 10/30/2024).

[63] J. Kim, H. You, and C. Kim, "Shock-capturing model using pid controller for high-order discontinuous galerkin method," in *AIAA SCITECH 2023 Forum*. American Institute of Aeronautics and Astronautics, 2023. DOI: 10.2514/6.2023-2630. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2023-2630. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2023-2630.

[64] A. E. Papoutsakis, I. Nompelis, and J. A. Ekaterinaris, "Discontinuous Galerkin Discretization of Chemically Reacting Flows," en, in *52nd Aerospace Sciences Meeting*, National Harbor, Maryland: American Institute of Aeronautics and Astronautics, Jan. 2014, ISBN: 9781624102561. DOI: 10.2514/6.2014-0068. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2014-0068 (visited on 10/30/2024).

[65] Y. Li and Z. J. Wang, "A convergent and accuracy preserving limiter for the FR/CPR method," en, in *55th AIAA Aerospace Sciences Meeting*, Grapevine, Texas: American Institute of Aeronautics and Astronautics, Jan. 2017, ISBN: 9781624104473. DOI: 10.2514/6.2017-0756. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2017-0756 (visited on 10/30/2024).

[66] J. S. Park, M. Yu, C. Kim, and Z. Wang, "Comparative study of shock-capturing methods for high-order cpr: Mlp and artificial viscosity," in *The Eighth International Conference on Computational Fluid Dynamics (ICCFD8)*, Jul. 2014.

[67] X. Zhang and C.-W. Shu, "On positivity-preserving high order discontinuous galerkin schemes for compressible euler equations on rectangular meshes," *Journal of Computational Physics*, vol. 229, no. 23, pp. 8918–8934, 2010, ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2010.08.016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999110004535.

[68] X. Zhang and C.-W. Shu, "Positivity-preserving high order discontinuous galerkin schemes for compressible euler equations with source terms," *Journal of Computational Physics*, vol. 230, no. 4, pp. 1238–1248, 2011, ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2010.10.036. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999110006017.

[69] C. Rising, G. B. Goodwin, E. J. Ching, K. Viswanath, and R. Johnson, "On the use of h and p-refinement for simulations of high enthalpy, compressible, chemically reacting flow using a discontinuous galerkin approach," en, in *AIAA SCITECH 2024 Forum*, American Institute of Aeronautics and Astronautics, 2024. DOI: 10.2514/6.2024-0911. [Online]. Available: https://doi.org/10.2514/6.2024-0911 (visited on 2024).

[70] A. Gouasmi, K. Duraisamy, and S. M. Murman, "Formulation of entropy-stable schemes for the multicomponent compressible euler equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 363, p. 112 912, 2020, ISSN: 0045-7825. DOI: https://doi.org/10.1016/j.cma.2020.112912. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045782520300955.

[71] J. P. Slotnick, A. Khodadoust, J. Alonso, *et al.*, "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," NASA, Tech. Rep. NF1676L-18332, Mar. 2014, NTRS Author Affiliations: Boeing (United States), Stanford University, Massachusetts Institute of Technology, National Center for Supercomputing Applications, Pratt & Whitney United Technologies Corp., University of Wyoming NTRS Document ID: 20140003093 NTRS Research Center: Langley Research Center (LaRC). [Online]. Available: https://ntrs.nasa.gov/citations/20140003093 (visited on 10/31/2024).

[72] E. J. Ching, R. F. Johnson, and A. D. Kercher, "Positivity-preserving and entropy-bounded discontinuous galerkin method for the chemically reacting, compressible euler equations. part ii: The multidimensional case," *Journal of Computational Physics*, vol. 505, p. 112 878, 2024, ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2024.112878. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S002199912400127X.

[73] T. Dzanic and F. Witherden, "Positivity-preserving entropy-based adaptive filtering for discontinuous spectral element methods," *Journal of Computational Physics*, vol. 468, p. 111 501, 2022, ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2022.111501. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999122005630.

[74] T. Dzanic and F. Witherden, "Positivity-preserving entropy filtering for the ideal magnetohydrodynamics equations," *Computers & Fluids*, vol. 266, p. 106 056, 2023, ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2023.106056. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045793023002815.

[75] S. Hennemann, A. M. Rueda-Ramírez, F. J. Hindenlang, and G. J. Gassner, "A provably entropy stable subcell shock capturing approach for high order split form dg for the compressible euler equations," *Journal of Computational Physics*, vol. 426, p. 109 935, 2021, ISSN: 0021-9991. DOI: https://doi.org/10.1016/j.jcp.2020.109935. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999120307099.

[76] F. Basile, J.-B. Chapelier, M. de la Llave Plata, R. Laraufie, and P. Frey, "Unstructured h- and hp-adaptive strategies for discontinuous galerkin methods based on a posteriori error estimation for compressible flows," *Computers & Fluids*, vol. 233, p. 105 245, 2022, ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2021.105245. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045793021003509.

[77] Blanes, Sergio, Iserles, Arieh, and Macnamara, Shev, "Positivity-preserving methods for ordinary differential equations," *ESAIM: M2AN*, vol. 56, no. 6, pp. 1843–1870, 2022. DOI: 10.1051/m2an/2022042. [Online]. Available: https://doi.org/10.1051/m2an/2022042.

[78] M. Ciallella, L. Micalizzi, P. Öffner, and D. Torlo, "An arbitrary high order and positivity preserving method for the shallow water equations," *Computers & Fluids*, vol. 247, p. 105 630, 2022, ISSN: 0045-7930. DOI: https://doi.org/10.1016/j.compfluid.2022.105630. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045793022002237.

[79] S. Karl, J. M. Schramm, and K. Hannemann, "High enthalpy cylinder flow in heg: A basis for cfd validation," in *33rd AIAA Fluid Dynamics Conference and Exhibit*. DOI: 10.2514/6.2003-4252. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2003-4252. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2003-4252.

[80] P. Tran, J. Paulat, and P. Boukhobza, *Rto-en-avt-130 10 - 1 re-entry flight experiments lessons learned – the atmospheric reentry demonstrator ard*, 2007. [Online]. Available: https://www.sto.nato.int/publications/STO%20Educational%20Notes/RTO-EN-AVT-130/EN-AVT-130-10.pdf.

[81] M. Yu, Z. Qiu, and Y. Takahashi, "Numerical investigation of surface catalytic effect on the plasma sheath of a hypersonic re-entry capsule," *Physics of Fluids*, vol. 35, no. 5, p. 056 106, May 2023, ISSN: 1070-6631. DOI: 10.1063/5.0149660. eprint: https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/5.0149660/17475950/056106_1_5.0149660.pdf. [Online]. Available: https://doi.org/10.1063/5.0149660.

[82] S. Kranc, M. C. Yuen, and A. B. Cumbel, *Nasa cr-1393 experimental investigation of magnetoaerodynamic flow around blunt bodies*, Aug. 1969. [Online]. Available: https://ntrs.nasa.gov/api/citations/19690025519/downloads/19690025519.pdf (visited on 06/03/2025).

[83] D. Gottlieb and C.-W. Shu, "On the gibbs phenomenon and its resolution," *SIAM Review*, vol. 39, pp. 644–668, Jan. 1997. DOI: 10.1137/s0036144596301390. (visited on 01/27/2021).

[84] E. M. Stein and R. Shakarchi, *Fourier Analysis*. Princeton University Press, Feb. 2011.

[85] C. Geuzaine and J.-F. Remacle, *Gmsh: A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*, gmsh.info. [Online]. Available: https://gmsh.info/.

[86] M.-S. Liou, "A Sequel to AUSM: AUSM+," en, *Journal of Computational Physics*, vol. 129, no. 2, pp. 364–382, Dec. 1996, ISSN: 00219991. DOI: 10.1006/jcph.1996.0256. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999196902569 (visited on 06/12/2025).

[87] M.-S. Liou, "A sequel to AUSM, Part II: AUSM+-up for all speeds," en, *Journal of Computational Physics*, vol. 214, no. 1, pp. 137–170, May 2006, ISSN: 00219991. DOI: 10.1016/j.jcp.2005.09.020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0021999105004274 (visited on 06/12/2025).