

High-Order Discontinuous Galerkin Methods for Multi-Species Flows

Liki RAMOS MATSUYAMA



Computational Aerodynamics Group
Department of Mechanical Engineering
McGill University
Montréal, Québec, Canada

December, 2024

A thesis submitted to McGill University in partial fulfillment of the
requirements of the degree of

Master of Science in Mechanical Engineering

©Liki RAMOS MATSUYAMA, 2024

Abstract

Real gases consist of multiple chemical species, including oxygen, nitrogen, and hydrogen. For practical applications, a mono-species model is often used to simplify flow simulations by representing the gas with a single chemical species. However, for combustion or mixing flows where the different species have different masses, or for high-speed and high-temperature flows where thermal properties such as heat capacity and enthalpy exhibit nonlinear relations with temperature, a multi-species model under a calorically imperfect assumption becomes essential for accurate analysis.

In computational fluid dynamics (CFD), computational methods and modelling are carefully chosen to balance computational costs with achieving high accuracy. The multi-species model involves a larger number of conservation of mass equations, making it more computationally expensive compared to the mono-species model. To address this, parallel computing techniques are employed to efficiently utilize computational resources. The discontinuous Galerkin (DG) method stands out for its capability to achieve both high efficiency in parallel computing and high-order accuracy.

Motivated by the aforementioned research background, this thesis aims to develop numerical codes for multi-species flow simulations by implementing a non-reacting (mixing), calorically imperfect physics model and conservation of mass for each species into a DG solver. Extensive tests have been conducted to verify and validate the implemented multi-species solver, demonstrating its accuracy.

Résumé

Les gaz réels sont constitués de plusieurs espèces chimiques, dont l'oxygène, l'azote et l'hydrogène. Pour les applications pratiques, un modèle monospécifique est souvent utilisé pour simplifier les simulations d'écoulement en représentant le gaz avec une seule espèce chimique. Cependant, pour les écoulements de combustion ou de mélange, où les différentes espèces ont des masses différentes, ou pour les écoulements à grande vitesse et à haute température, où les propriétés thermiques telles que la capacité calorifique et l'enthalpie présentent des relations non linéaires avec la température, un modèle multi-espèces devient nécessaire.

Dans la dynamique des fluides numérique (CFD), les méthodes de calcul et la modélisation sont soigneusement choisies pour équilibrer les coûts de calcul et obtenir une grande précision. Le modèle multi-espèces implique un plus grand nombre d'équations de conservation de la masse, ce qui le rend plus coûteux par rapport au modèle monospécifique. Pour y remédier, des techniques de calcul parallèles sont employées pour utiliser efficacement les ressources informatiques. La méthode de Galerkin Discontinu (DG) se distingue par sa capacité à atteindre à la fois une grande efficacité dans le calcul parallèle et une précision d'ordre élevé.

Motivée par le contexte de recherche susmentionné, cette thèse vise à développer des codes numériques pour les simulations de flux multi-espèces en mettant en œuvre un modèle physique caloriquement imparfait sans réaction (mélange) et en conservant la masse pour chaque espèce dans un solveur DG. Des tests approfondis ont été menés pour vérifier et valider le solveur multi-espèces mis en œuvre, démontrant ainsi sa précision.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Siva Nadarajah, for his tremendous support throughout my Master's studies. His constant advice and encouragement, expert knowledge, and invaluable guidance have made this work possible. His feedback and teachings have expanded my skills and made me think more critically about my work. I appreciate the time and effort he has dedicated to help me grow.

Next, I am grateful for my colleagues from the McGill Computational Aerodynamics Group for their support throughout my thesis. Notably, I am thankful to Julien Brillion, who sat next to me, for helping expand my knowledge on multi-species physics and PHiLiP, engaging in late-night discussions, and working through bugs in my code. I would also like to thank Pranshul Thakur for always sparing time and willing to answer my questions on coding and theories. The time we shared in the lab and outside of school was very special to me, and I hope we remain in touch and see you all even after we start our careers.

I would like to thank my family, friends, and my girlfriend for their unconditional support over the past few years. Their messages and phone calls always cheered me up and pushed me to take the next steps.

Lastly, I would also like to thank the National Sciences and Engineering Research Council (NSERC) and the Department of Mechanical Engineering at McGill University for their financial support throughout the completion of this degree.

Table of Contents

Abstract	i
Résumé	ii
Acknowledgements	iii
List of Figures	x
List of Symbols	xi
1 Introduction	1
1.1 Computational Fluid Dynamics	1
1.2 Multi-Species Model	3
1.3 Calorically Imperfect Model	5
1.4 Thesis Objective	7
2 Methodology	9
2.1 Discontinuous Galerkin Method	9
2.2 Calorically Imperfect Multi-Species Euler Equations	11
2.2.1 Dimensional Form	14
2.2.2 NonDimensional Form	18
2.2.3 Algorithm	22
2.3 Error Estimation	32
3 Verification	34
3.1 1D Low-Temperature Vortex Advection	34
3.1.1 Problem Statement	35

3.1.2	Results	35
3.2	1D High-Temperature Vortex Advection	41
3.2.1	Problem Statement	42
3.2.2	Results	42
3.3	2D Vortex Advection	44
3.3.1	Problem Statement	45
3.3.2	Results	46
4	Validation	55
4.1	2D Isentropic Euler Vortex Advection	55
4.1.1	Problem Statement	56
4.1.2	Results	57
4.2	2D Fuel Droplet Advection	58
4.2.1	Problem Statement	59
4.2.2	Results	59
5	Conclusion	66
5.1	Future Work	67
5.1.1	3D Test Case (Multi-Species Inviscid Taylor-Green Vortex)	68

List of Figures

1.1	Multi-Species gas in aerospace engineering.	4
3.1	Profiles of mixture density, H_2 mass fraction, and temperature at the initial time for the 1D low-temperature vortex advection problem.	36
3.2	Profiles of mixture density, H_2 mass fraction, and temperature after 0.1 cycle using a $p2$ approximation for the 1D low-temperature vortex advection problem.	37
3.3	Profiles of mixture density, H_2 mass fraction, and temperature after 0.1 cycle using a $p5$ approximation for the 1D low-temperature vortex advection problem.	38
3.4	Profiles of mixture density, H_2 mass fraction, and temperature after 0.1 cycle using $p2$ and $p5$ approximations using different grid sizes such that DOFs are equivalent for the 1D low-temperature vortex advection problem.	39
3.5	Profiles of mixture density after 0.1 cycle comparing results by $\Delta t = 1 \times 10^{-6}$ and $\Delta t = 1 \times 10^{-5}$ using a $p2$ approximation with $N = 64$ for the 1D low-temperature vortex advection problem.	40
3.6	L_1 error norms of mixture density and H_2 mass fraction with a different number of elements for the 1D low-temperature vortex advection problem including results by Wang et al. [80]. Blue dot line: 2nd-order convergence rate; red dot line: 5th-order convergence rate; green solid line with circle symbol: $p2$ results; orange solid line with square symbol: $p5$ results; purple dot line: 2nd-order convergence rate by Wang et al.; blown dot line: 2nd-order convergence rate by Wang et al.; pink solid line with triangle symbol: $p2$ results by Wang et al.; gray solid line with diamond symbol: $p5$ results by Wang et al.	41

3.7	L_2 error norms of mixture density and H_2 mass fraction with a different number of elements for the 1D low-temperature vortex advection problem. Blue dot line: 2nd-order convergence rate; red dot line: 5th-order convergence rate; green solid line with circle symbol: $p2$ results; orange solid line with square symbol: $p5$ results.	42
3.8	L_∞ error norms of mixture density and H_2 mass fraction with different number of elements for the 1D low-temperature vortex advection problem including results by Wang et al. [80]. Blue dot line: 2nd-order convergence rate; red dot line: 5th-order convergence rate; green solid line with circle symbol: $p2$ results; orange solid line with square symbol: $p5$ results; purple dot line: 2nd-order convergence rate by Wang et al.; blown dot line: 2nd-order convergence rate by Wang et al.; pink solid line with triangle symbol: $p2$ results by Wang et al.; gray solid line with diamond symbol: $p5$ results by Wang et al.	43
3.9	Profiles of mixture density, H_2 mass fraction, and temperature at the initial time for the 1D high-temperature vortex advection problem.	44
3.10	Profiles of mixture density, H_2 mass fraction, and temperature after 0.1 cycle for the 1D high-temperature vortex advection problem.	45
3.11	L_1 error norms of mixture density and temperature with different number of elements for the 1D high-temperature vortex advection problem. Blue dot line: 2nd-order convergence rate; red dot line: 5th-order convergence rate; green solid line with circle symbol: $p2$ results; orange solid line with square symbol: $p5$ results.	46
3.12	L_2 error norms of mixture density and temperature with different number of elements for the 1D high-temperature vortex advection problem. Blue dot line: 2nd-order convergence rate; red dot line: 5th-order convergence rate; green solid line with circle symbol: $p2$ results; orange solid line with square symbol: $p5$ results.	47

3.13	L_∞ error norms of mixture density and temperature with different numbers of elements for the 1D high-temperature vortex advection problem. Blue dot line: 2nd-order convergence rate; red dot line: 5th-order convergence rate; green solid line with circle symbol: $p2$ results; orange solid line with square symbol: $p5$ results.	48
3.14	Contours of mixture density, H_2 mass fraction, and temperature at the initial time for the 2D vortex advection problem.	49
3.15	Contours of mixture density, H_2 mass fraction, and temperature after 0.1 cycle for the 2D vortex advection problem.	50
3.16	Profiles of mixture density, H_2 mass fraction, and temperature at $y^* = x^*$ axis after 0.1 of a cycle using $p1$ and $p3$ approximations using different grid sizes such that DOFs are equivalent for the 2D low-temperature vortex advection problem.	51
3.17	L_1 error norm of mixture density and H_2 mass fraction with a different number of elements for the 2D vortex advection problem. Blue dot line: 1st-order convergence rate; red dot line: 2nd-order convergence rate; green dot line: 3rd order convergence rate; orange solid line with circle symbol: $p1$ results; purple solid line with square symbol: $p2$ results; brown solid line with triangle symbol: $p3$ results.	52
3.18	L_2 error norm of mixture density and H_2 mass fraction with a different number of elements for the 2D vortex advection problem. Blue dot line: 1st-order convergence rate; red dot line: 2nd-order convergence rate; green dot line: 3rd order convergence rate; orange solid line with circle symbol: $p1$ results; purple solid line with square symbol: $p2$ results; brown solid line with triangle symbol: $p3$ results.	53

3.19	L_∞ error norm of mixture density and H_2 mass fraction with different number of elements for the 2D vortex advection problem. Blue dot line: 1st-order convergence rate; red dot line: 2nd-order convergence rate; green dot line: 3rd order convergence rate; orange solid line with circle symbol: $p1$ results; purple solid line with square symbol: $p2$ results; brown solid line with triangle symbol: $p3$ results.	54
4.1	Contours of N_2 density with the time progress using a $p4$ approximation for the 2D isentropic Euler vortex advection problem.	61
4.2	Contours of N_2 density after flow reversal with the grid refinement for the 2D isentropic Euler vortex advection problem.	62
4.3	Comparison of the cross-section of N_2 density at $x^* = 0$ after flow reversal using a $p4$ approximation. Blue dot line: $N = 20$ by Trojak et al.; red solid line: $N = 16$; green dot line: $N = 40$ by Trojak et al.; orange solid line: $N = 32$; purple dot line: $N = 80$ by Trojak et al.; brown solid line: $N = 64$. .	63
4.4	Comparison of the cross-section of N_2 density at $x^* = 0$ after flow reversal. Blue dot line: $p = 2, N = 32, DOFs = 96$; red solid line: $p = 4, N = 16, DOFs = 80$; green dot line: $p = 2, N = 64, DOFs = 192$; orange solid line: $p = 4, N = 32, DOFs = 160$; purple dot line: $p = 2, N = 128, DOFs = 384$; brown solid line: $p = 4, N = 64, DOFs = 320$	63
4.5	Contours of fuel mass fraction with time progress for the 2D fuel drop advection problem ($p = 3$ and $N = 32$).	64
4.6	Comparison of the cross-section of fuel mass fraction at $y^* = 0.5$ after flow reversal for the 2D fuel droplet advection problem. Black dot line: the initial conditions; blue solid line: results by Ma et al.; red solid line: $p2$ results with low-temperature; $p5$ results with low-temperature; orange solid line: $p2$ results with high-temperature.	65

5.1	Contours of vorticity magnitude with the time progress using a $p = 3$ approximation and $N = 8$ grid resolution in each direction for the multi-species inviscid Taylor-Green vortex problem.	71
5.2	Contours of N_2 mass fraction with the time progress using a $p = 3$ approximation and $N = 8$ grid resolution in each direction for the multi-species inviscid Taylor-Green vortex problem.	72

List of Symbols

Variables

x	x -direction
y	y -direction
z	z -direction
t	Time
ρ	Density
u	Velocity (x -direction)
v	Velocity (y -direction)
w	Velocity (z -direction)
P	Pressure
T	Temperate
T_{ref}	Reference temperature (= 298.15 K)
k	Specific kinetic energy (kinetic energy per unit mass)
E	Specific total energy (total energy per unit mass)
e	Specific internal energy (internal energy per unit mass)

e_t	Specific translational energy (translational energy per unit mass)
e_r	Specific rotational energy (rotational energy per unit mass)
e_v	Specific vibrational energy (vibrational energy per unit mass)
e_e	Specific electronic energy (electronic energy per unit mass)
e_f	Specific energy of formation (energy of formation per unit mass)
H	Specific total enthalpy (total enthalpy per unit mass)
h	Specific enthalpy (enthalpy per unit mass)
h_f	Specific enthalpy of formation (enthalpy of formation per unit mass)
C_p	Specific heat at constant pressure
C_v	Specific heat at constant volume
γ	Specific heat ratio
MW	Molecular weight
R_u	Universal gas constant (molecular)
$R = \frac{R_u}{MW}$	Gas constant (per unit mass)
a	Speed of sound
Y_s	Mass fraction of the species
σ	Physics property representing one of $\rho, P, E, e, H, h, C_p, C_v, \gamma, R, M$ and a
$n_{species}$	Number of species (≥ 2)
n_{state}	Number of elements in state vectors
a_i, b_i	Polynomial coefficients given by NASA CAP program [1, 2]

tol	Tolerance for Newton-Raphson method
err	Error in iterations of Newton-Raphson method
dim	Number of dimensions
$vel2$	Squared value of velocities
L	Length of the domain
N	Number of elements per dimension
$DOFs$	Degrees of freedom per dimension
L_p	L-p norm
$\mathbf{Q}$	Conservative solutions
$\mathbf{Q}[\mathbf{i}]$	i^{th} row element of $\mathbf{Q}$
$\mathbf{E}$	Convective flux (x -direction)
$\mathbf{F}$	Convective flux (y -direction)
$\mathbf{P}$	Primitive solutions
ϵ_p	L_p error norm
r	Ratio of coarse to fine mesh element spacing
p	Order of accuracy
u_h	Discrete numerical solutions
u_h^k	Local solutions
$\mathcal{R}$	Residual
$\mathcal{F}$	Flux vector

$\hat{\mathcal{F}}$	Numerical flux vector
$\mathcal{S}$	Source term
Ω	Physical domain
Γ	Physical boundaries
Ω_h	Computational domain
Ω_k	k^{th} element of computational domain
$\partial\Omega_k$	Boundaries of k^{th} element
ϕ	Basis functions
ψ	Test functions
p_k	Local polynomial order

Notations

$()_s$	Species property
$\tilde{()}$	Mixture property
$()^*$	Nondimensional notation
$()_{ref}$	Reference values (defined dimensionally)
$()^n$	Properties at n^{th} time step
$()_n$	Properties at n^{th} step in the Newton-Raphson iteration
$()'$	Prime notation for derivatives
$()^+$	An exterior quantity at the interface
$()^-$	An interior quantity at the interface

Acronym

CFD	Computational Fluid Dynamic
DG	Discontinuous Galerkin
WENO	Weighted Essentially Non-Oscillatory
GFM	Ghost Fluid Method
CAP	Coefficients and Properties
RANS	Reynolds-Averaged Navier–Stokes
FEM	Finite Element Method
FVM	Finite Volume Method
GL	Gauss-Lobatto
DOF	Degree of Freedom
DNS	Direct Numerical Simulation

Chapter 1

Introduction

This thesis provides a comprehensive description of the calorically imperfect model and the multi-species model implemented in the inviscid flow solver. These models can handle both low-temperature, low-speed conditions and high-temperature, high-speed flow conditions. The solver used in this work is based on the Discontinuous Galerkin (DG) method, which has gained popularity for its ability to achieve high-order accuracy in computational simulations.

1.1 Computational Fluid Dynamics

Fluid dynamics plays a critical role in the design of various aerospace products, such as engines. There are two primary approaches to addressing fluid dynamics problems: experimental and computational methods. The experimental approach provides valuable insights into the complex physical phenomena involved in fluid dynamics. However, the cost of the necessary equipment is high, particularly for large-scale and complex applications such as combustor flows, and measuring flow quantities can be challenging. On the other hand, the computational method relies on physics-based models and offers detailed flow data. Recent advancements in computer hardware, including processors and memory capacity, have significantly enhanced computational capabilities. As a result, Computational Fluid Dy-

namics (CFD) has emerged as a competitive and viable approach to not only studying fluid phenomena but also designing new products through simulations.

In CFD, the methods for solving governing equations have evolved from traditional approaches such as finite difference and finite volume techniques to more advanced strategies, such as finite element methods and high-order schemes. High-order methods, in particular, are employed to achieve greater accuracy in fluid flow simulations. Although low-order methods are robust, reliable, and widely used in practical computations, high-order methods have garnered significant academic and industrial interest over the past decade [3]. The primary reason for this interest is that high-order methods offer superior accuracy at a reduced computational cost when compared to low-order methods [4]. By using higher-degree polynomials to approximate solutions, high-order methods can capture finer details of fluid flow, leading to a deeper understanding of complex fluid phenomena. In applications, high-order methods have proven particularly effective in simulating complex flow behaviors, such as resolving near-wall turbulence flow structures at Kolmogorov microscales. One of their key advantages is the ability to produce accurate results with fewer computational grid points, due to their low dissipation properties [5]. This efficiency makes them highly suitable for large-scale, intricate scale resolving simulations, where computational resources are a concern. For instance, Shu [6] discussed the applications of several non-oscillatory high-order methods, including the Weighted Essentially Non-Oscillatory (WENO) finite difference and finite volume schemes, as well as the Discontinuous Galerkin (DG) finite element schemes. Among these, the DG method stands out as one of the most widely used high-order numerical approaches for solving compressible flow equations on unstructured meshes [7]. The details of the DG method, including its fundamental mathematics and advantages, will be explored in the following chapter.

1.2 Multi-Species Model

Multi-species flows consist of multiple chemical species interacting and mixing within a fluid. These phenomena are extensively studied across various fields, including combustion, climate modeling, and chemical process engineering [8–10]. A multi-species model is employed in CFD to simulate mixing or chemically reacting flows. In aerodynamics, multi-species gases are particularly relevant in combustions flows, ranging from subsonic to hypersonic speeds, as illustrated in Figure 1.1. At subsonic speeds, a jet engine (drawn in Figure 1.1a) combusts jet fuel and air to generate thrust. For supersonic speeds, a scramjet engine (illustrated in Figure 1.1b), considered a future propulsion system for supersonic airplanes or reusable launch vehicles, utilizes hydrogen as fuel and air as an oxidizer. At hypersonic speeds, a rocket engine (shown in Figure 1.1c) burns liquid oxygen and liquid hydrogen to produce energy and accelerate the flow to hypersonic velocities. Given the widespread use of jet engines in commercial aviation and the growing potential for supersonic transportation and space exploration, there is significant interest among researchers and engineers in studying mixing and chemically reacting flows. Corresponding to this interest, multi-species models have been implemented in CFD solvers [11] such as ANSYS (CFX and Fluent) [12–14], Comsol Multiphysics [15], OpenFOAM [16, 17], as well as simulation codes developed in various programming languages [18–21].

There are several approaches to modeling flows with multiple chemical species. One approach is the multi-phase model using the Ghost Fluid Method (GFM), developed by Fedkiw et al. [22, 23]. This method is particularly useful for modeling multi-phase flows, such as when fuel is in a liquid state and air is in a gaseous state. The GFM is a flexible technique to treat compressible two-phase flows. It captures the material interface by solving the level set equation [24–27] and treats the interface as a boundary that separates a real fluid on one side and its corresponding ghost fluid on the other side [28]. However, a significant drawback of the GFM is that it is a non-conservative method, potentially leading to inaccuracies in preserving key physical quantities such as mass, momentum, and energy. Another approach



(a) Jet engine [36]



(b) Scramjet engine [37]



(c) Rocket Engine [38]

Figure 1.1: Multi-Species gas in aerospace engineering.

involves modeling the flow as a multi-species gas, where different species are distinguished within the same state. Similar to mono-species flow simulations, the multi-species model solves systems of conservation laws. However, unlike the mono-species model, multi-species flows require additional conservation laws for the mass of each species. The mass fraction of each species is considered to track the various flow properties of each chemical species [29]. Details of the multi-species conservation laws and the methods for computing mixture and species properties are well presented in [30], and the multi-species Euler equations are summarized in [31–35]. The Euler equations are derived without considering viscous effects, making them simpler and widely used in gas dynamics research [31].

When simulating non-reacting and low-temperature flows, such as ambient air under standard conditions—comprising nitrogen, oxygen, argon, etc.—average values of molar weight and thermal properties across species can approximate the air as a single-species gas. However, in combustor or mixing flows, where different chemical species exhibit significantly varied behaviors, a multi-species model becomes essential for accurate analysis.

Given these considerations, the multi-species Euler equations are employed in this work as the governing equations for solving multi-species flows. These equations account for the conservation laws of mixture mass, mixture momentum, mixture energy, and the mass of each species.

1.3 Calorically Imperfect Model

In CFD, the selection of physical models is crucial for balancing accuracy and computational efficiency. From a thermal physics perspective, gas models can be classified into two categories: the calorically perfect model and the calorically imperfect model. By definition, a calorically perfect model assumes specific heat at constant pressure (C_p) and specific heat at constant volume (C_v) as constant values [39]. In this model, the perfect gas equations typically use a value of 1.40 for the ratio of specific heats ($\gamma = C_p/C_v$). This assumption is true for air at low-speed or at low-temperature conditions [40]. The calorically perfect gas assumption significantly reduces mathematical complexity and computational effort required, and it is still applicable to a wide variety of scenarios in CFD [41]. As a result, the calorically perfect model is often employed in general CFD applications to streamline simulations and reduce computational costs.

However, the applicability of the calorically perfect model is limited to low-temperature conditions. Many aeronautical engineering applications, particularly those involving combustion or high-speed flows, operate beyond the temperature range where this assumption is valid [39]. For instance, in hypersonic flows where static temperatures are high, the calorically perfect gas assumption becomes inaccurate [42]. To address these limitations, the calorically imperfect model is utilized. In this model, specific heat capacities vary with temperature due to the excitation of the vibrational modes within molecules [40]. This behavior is often modeled using empirical polynomial expressions for specific heat capacities, initially represented by five-term polynomials [43] and later extended to eight terms [1]. In this study, the calorically imperfect model is chosen to accurately compute high-temperature and high-speed flows. The NASA Coefficients and Properties (CAP) program [1,2] is employed, providing polynomial coefficients for specific heat capacities and enthalpy as functions of temperature for each species.

The calorically imperfect model has been widely used in previous CFD research for high-speed and high-temperature flows to achieve more accurate computations. For ex-

ample, Lampart et al. [44] implemented this model in a 3D RANS (Reynolds-Averaged Navier–Stokes) solver and validated it on low-pressure turbine simulations, comparing results with both experimental data and the calorically perfect model. They demonstrated that flow properties such as pressure, temperature, enthalpy, and mass flow rate were more accurately captured by the calorically imperfect model, with better agreement to experimental data than the calorically perfect model. In addition, May et al. [45] investigated a DG solver for high-speed flows using a calorically imperfect model. In their study, flows around a diamond-shaped airfoil were solved using a p_2 approximation. The analysis focused on the drag coefficient around the airfoil as a function of the number of mesh elements, and the results were compared with reference values. Their findings confirmed that the solutions computed with the calorically imperfect model converge to the reference values as the number of elements increases, thereby validating the accuracy of the model. Jiang et al. [46] also studied a high-order scheme under a calorically imperfect assumption to enhance hypersonic heating predictions. They used a fifth-order WENO scheme [47] for their computations. The benchmark test for stagnation point prediction on a blunt body compared the computational error using both calorically imperfect and perfect gas models. A wide range of flow conditions, with Mach numbers ranging from 5 to 10, was considered in this test. Jiang et al. [46] demonstrated that the error in the calorically imperfect model was significantly smaller than that in the calorically perfect model. Moreover, as the Mach number increased, the disparity between the two models grew, highlighting the superiority of the calorically imperfect model, particularly at high Mach numbers where high-temperature effects become more prominent and cannot be ignored. Overall, the calorically imperfect model has been successfully implemented in CFD solvers and used for high-speed and high-temperature flows, such as hypersonic and combusting scenarios, leading to more accurate predictions of complex flow dynamics.

There are a limited number of articles focusing on the use of the DG method for multi-species gas simulations. For example, Trojak et al. [48] recently introduced a positivity-preserving numerical stabilization approach for the high-order DG method applied to multi-

species flow. However, their study only employed the calorically perfect model. Similarly, Andrés et al. [49] introduced an entropy-stable discretization using a DG method for the calorically perfect multi-species plasmas. Lv et al. [50] focused on the development of a DG method for chemically reacting multi-species flows, and Du et al. [51] developed an oscillation-free DG method for solving the multi-species chemically reacting flows, but both studies restricted their approach to the calorically perfect gas model for simplicity. Moreover, Luo et al. [52] proposed a high-order DG method for the numerical simulation of compressible multi-species flows, but they also used constant specific heats, meaning that the gas was treated as calorically perfect. These studies highlight a current gap in research regarding the application of the DG method with a calorically imperfect gas model. Therefore, implementing a multi-species model within a DG solver that includes the calorically imperfect assumption for high-temperature flows would be a valuable contribution to the CFD field.

1.4 Thesis Objective

The primary goal of this research is to develop a multi-species high-order code with the eventual objective of advancing the field of entropy-stability, which ensures that the total entropy of the discrete system is conserved for dissipation-free problems, thereby respecting the second law of thermodynamics [53]. However, implementing a reacting model and a multi-species entropy-stable scheme in a DG solver presents significant challenges due to their inherent complexity. As an initial step towards achieving this goal, the multi-species model and the calorically imperfect model are implemented in PHiLiP (Parallel High-Order Library for PDEs through hp-adaptive Discontinuous Galerkin methods) [54], an in-house DG solver developed by the McGill Computational Aerodynamics Group. This implementation marks a contribution to the code development framework within the research group, as PHiLiP previously supported only a mono-species model and a calorically perfect model. Additionally, conducting relevant test cases using the newly developed calorically imperfect

multi-species DG solver contributes to the CFD field, given the current scarcity of studies that utilize these models within a DG framework.

The following outline presents the key steps for developing these models in a DG solver: (1) understanding the fundamentals of the non-reacting multi-species model and the calorically imperfect model, including algorithms to compute multi-species convective fluxes from multi-species conservative solutions under the calorically imperfect assumption; (2) implementing these models in a DG solver; (3) testing the code through verification; (4) testing the implemented models through validation.

This thesis is structured as follows: Firstly, in Chapter 2, we will review the methodology of DG computations, the multi-species Euler equations using the calorically imperfect model, and the error evaluations. We will introduce the dimensional and nondimensional equations and summarize all newly implemented functions in PHiLiP. Secondly, in Chapter 3, we will provide the results and analysis of the computations for verification. We conducted three verification tests using the DG solver with the implemented models, including one-dimensional and two-dimensional tests where the multi-species model is important, as well as a test with high-temperature conditions where the calorically imperfect model is crucial. Following verification tests, validation tests are summarized in Chapter 4. Two numerical tests are run, and their results are compared with previous studies using the multi-species model to check the applicability of the implemented models. Finally, Chapter 5 will present the conclusion of this research, and the possible future work will be discussed.

Chapter 2

Methodology

This chapter provides a brief overview of fundamental numerical methods central to this work. The first section describes the Discontinuous Galerkin (DG) method employed in this study. The second part covers the multi-species Euler equations using the calorically imperfect model, presented in both dimensional and nondimensional forms, along with relevant computational algorithms. The third section details the error estimations used to evaluate the order of accuracy.

2.1 Discontinuous Galerkin Method

The Discontinuous Galerkin (DG) method is a high-order numerical technique for solving differential equations [55]. The method combines the high-order accuracy of Finite Element Methods (FEM) [56] with the flexibility of permitting discontinuities at cell interfaces, a feature characteristic of Finite Volume Methods (FVM) [57]. The DG method is an advantageous high-order numerical scheme known for its high parallel computing efficiency, which helps reduce computational time compared to other schemes [58]. This efficiency arises from its discontinuous nature at element boundaries similar to FVM. Additionally, the DG method is similar to the FEM as it approximates the solution within each cell as a linear combination of basis functions. This enables the DG method to achieve high-order solutions

without requiring a large stencil, ensuring accurate approximate solutions across the entire domain that depend only on neighboring cells. Considering these advantages of balancing high-order accuracy and computational cost, the DG method is employed in PHiLiP [54], the in-house CFD solver developed by the Computational Aerodynamics Group at McGill University. The developed calorically imperfect multi-species model has been implemented within this framework.

The Discontinuous Galerkin (DG) method was first introduced by Reed and Hill [59] with applications to the stationary neutron transport problem. Later, Cockburn and Shu [60–62] demonstrated that these methods can serve as powerful computational tools for solving systems of conservation laws. The DG method has since been expanded to address a wide range of fluid problems, including advection-diffusion cases [63], the Euler equations [64], and the Navier-Stokes equations [65]. The fundamental principles of the DG method will be summarized below [55, 66, 67].

Consider the initial value problem,:

$$\mathcal{R}(u(x, t), t) = \frac{\partial u(x, t)}{\partial t} + \nabla \cdot \mathcal{F}(u(x, t)) - S(u(x, t)) = 0 \quad (2.1)$$

where u represents the flow solution, $\mathcal{R}(u(x, t), t)$ denotes the residual, $\mathcal{F}(u)$ is the flux vector, and $S(u)$ is the source term. Equations of this form include the Euler equations used in this work. The computational task involves finding a discrete solution u_h such that $\mathcal{R}_h(u_h(x, t), t) = 0$. Consider a discrete numerical solution $u_h(x, t)$ that approximates the solution u to Equation 2.1 over the computational domain Ω_h . The computational domain Ω with the boundary Γ consists of non-overlapping elements $\Omega_k \in \Omega_h$ with the boundary $\partial\Omega_k$. Note that Ω_h is the collection of Ω_k . Each element has a local solution denoted as $u_h^k(x, t)$. In the DG method, the solution is represented by the direct sum of local solutions over each element as follows:

$$u_h(x, t) = \bigoplus_{\Omega_k \in \Omega_h} u_h^k(x, t). \quad (2.2)$$

The local solution on each element is discretized to be in space of polynomial with degree p_k and can be expressed as:

$$u_h^k(x) = \sum_{i=1}^{N(p_k)} u_i^k(t) \phi_{h,i}(x), \forall x \in \Omega_k, \quad (2.3)$$

where $\phi_{h,i}$ represents the i -th basis function within the element k and $N(p_k)$ denotes the number of basis functions. To determine the solution, the residual $\mathcal{R}(u(x, t), t)$ from Equation 2.1 must be orthogonal to arbitrary test functions ψ . In the DG method, test functions are chosen as the basis functions used to represent the solution, thus we consider $\psi = \phi$. Multiplying Equation 2.1 by a basis function ϕ_h and performing integration by parts over each element yields:

$$\int_{\Omega_k} \phi_h \frac{\partial u_h}{\partial t} d\Omega - \int_{\Omega_k} \nabla \phi_h \cdot \mathcal{F}(u_h) d\Omega - \int_{\Omega_k} \phi_h S(u_h) d\Omega + \int_{\partial\Omega_k} \phi_h \hat{\mathcal{F}}(u_h^+, u_h^-, n) d\Gamma = 0, \quad (2.4)$$

where superscripts $+$ and $-$ denote respectively the trace of the functions interior and exterior to the element's face Ω_k . This formulation provides the weak form of the DG method employed throughout this study. In this context, the term $\hat{\mathcal{F}}$ denotes the numerical flux function, which ensures the conservation across interfaces. For flux computations in this research, the Lax-Friedrichs flux method [68] is employed, and Gauss-Lobatto (GL) points [69] are used for the integration. The reader is advised to refer to [70] for a thorough derivation of the DG method employed within PHiLiP.

2.2 Calorically Imperfect Multi-Species Euler Equations

This section provides a summary of the Euler equations for multi-species gas in both dimensional and nondimensional forms. The convective fluxes are derived from the conservative solutions, and the corresponding algorithms are detailed here.

2D Euler equations for multi-species gas [31–35] is given by:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} = 0 \quad (2.5)$$

where

$$\mathbf{Q} = \begin{pmatrix} \tilde{\rho} \\ \tilde{\rho}u \\ \tilde{\rho}v \\ \tilde{\rho}\tilde{E} \\ \rho_s \end{pmatrix}, \mathbf{E} = \begin{pmatrix} \tilde{\rho}u \\ \tilde{\rho}u^2 + \tilde{P} \\ \tilde{\rho}uv \\ \tilde{\rho}u\tilde{H} \\ \rho_s u \end{pmatrix}, \mathbf{F} = \begin{pmatrix} \tilde{\rho}v \\ \tilde{\rho}uv \\ \tilde{\rho}v^2 + \tilde{P} \\ \rho v\tilde{H} \\ \rho_s v \end{pmatrix}, s = 0 : n_{species} - 2 \quad (2.6)$$

in a dimensional form. In Equations 2.5 and 2.6, the first equation represents the conservation of mixture mass, while the second and third equations describe the conservation of mixture momentum. The fourth equation gives the conservation of mixture energy, and the remaining equations address the conservation of mass for each individual species. The following equations apply to all species, indexed from 0 to $n_{species} - 1$, and similarly apply to mixture properties. For example, terms denoted as $(\cdot)_s$ can be substituted with $(\tilde{\cdot})$.

To advance the conservative solutions to the next time step, the fluxes are computed based on the solutions of the multi-species Euler equations at the current time step. As shown in Equation 2.6, the convective fluxes $\mathbf{E}$ and $\mathbf{F}$ are calculated using the mixture density $\tilde{\rho}$, velocities u and v , mixture pressure $\tilde{P}$, mixture specific enthalpy $\tilde{H}$, and species densities ρ_s for $s = 0 : n_{species} - 1$.

The ratio of specific heat is defined as:

$$\gamma = \frac{C_p}{C_v}, \quad (2.7)$$

with the specific heats related to the gas constant by:

$$R = C_p - C_v. \quad (2.8)$$

In a calorically perfect model, where C_v is independent of temperature, the specific internal energy is given by:

$$e = C_v T. \quad (2.9)$$

The equation of state is expressed as:

$$P = \rho R T. \quad (2.10)$$

Using Equations 2.7, 2.8, 2.9, and 2.10, the pressure is derived by the following equation:

$$P = \rho(\gamma - 1)(E - k), \quad (2.11)$$

where E is the specific total energy and k is the kinetic energy, given in a two-dimensional domain by:

$$k = \frac{1}{2}(u^2 + v^2). \quad (2.12)$$

Since the conservative variables in $\mathbf{Q}$ include specific total energy and velocities as seen in Equation 2.6, the pressure P can be directly derived from $\mathbf{Q}$ using Equation 2.11 in a calorically perfect model. The specific total enthalpy H is then calculated from E , P , and ρ using the following Equation:

$$H = E + \frac{P}{\rho}. \quad (2.13)$$

Once P and H are obtained, fluxes are computed, allowing the solution to advance to the next time step.

In contrast, for a calorically imperfect model, the mixture pressure cannot be directly obtained from the mixture energy because Equation 2.11 no longer holds, as C_p varies with temperature. In this case, a different approach is required. First, the temperature is computed from the specific total energy using a root-finding method, and then the pressure is derived from the equation of state. This section provides a comprehensive summary of the equations and algorithms needed to compute convective fluxes, as well as temperature, mix-

ture pressure, and mixture total enthalpy to solve the multi-species Euler equations under the calorically imperfect assumption.

2.2.1 Dimensional Form

Mass fraction is a measure used in chemistry and fluid dynamics to describe the proportion of a particular component in a mixture. It is defined as:

$$Y_s = \frac{\rho_s}{\tilde{\rho}}, \quad (2.14)$$

where Y_s is the mass fraction of species s , ρ_s is the density of species s , and $\tilde{\rho}$ is the mixture density. The mixture property $\tilde{\sigma}$ (σ can be replaced with $\rho, P, E, e, H, h, C_p, C_v, \gamma, R, M$, and a) is obtained using mass fraction as follows:

$$\tilde{\sigma} = \sum_{s=1}^{n_{species}} Y_s \sigma_s, \quad (2.15)$$

where σ_s represents the property of species s . This definition allows us to calculate the overall properties of the mixture based on the properties and proportions of the individual species. For example, the mixture density $\tilde{\rho}$, pressure $\tilde{P}$, and internal energy $\tilde{E}$ can all be derived from the corresponding properties of the individual species and their mass fractions within the mixture. This approach is crucial in multi-species CFD simulations. Similarly, the species property is computed using mass fraction and mixture property as:

$$\sigma_s = Y_s \tilde{\sigma}. \quad (2.16)$$

This relationship indicates that the property of a specific species can be determined by its mass fraction and the corresponding mixture property.

In terms of energy relations, specific kinetic energy is given by:

$$k = \frac{1}{2} (u^2 + v^2), \quad (2.17)$$

where u and v are the velocities in each direction. Specific species' total energy is the sum of specific species' internal energy and specific kinetic energy calculated in Equation 2.17, and it is expressed as:

$$E_s = e_s + \frac{1}{2}(u^2 + v^2) = e_s + k, \quad (2.18)$$

where E_s is the specific species' total energy of species s . Using this specific species' total energy, species pressure, and species' density, we can compute the specific species' total enthalpy by:

$$H_s = E_s + \frac{P_s}{\rho_s}, \quad (2.19)$$

where P_s denotes the species pressure, and it is calculated from the equation of state:

$$P_s = \rho_s R_s T, \quad (2.20)$$

where R_s is the gas constant of species s , and T is the temperature.

The temperature, in the calorically imperfect model, is computed using a root-finding algorithm applied to the equation for specific internal energy. There are two primary methods to obtain species specific internal energy. One method to obtain species specific internal energy is using Equation 2.18. Another method involves integrating with respect to temperature, as follows:

$$e_s(T) = \int_{T_{ref}}^T C_{v,s} dT + e_{f,s}, \quad (2.21)$$

where $e_s(T)$ is the specific internal energy of species s at temperature T , T_{ref} is reference temperature, $C_{v,s}$ is specific heat at constant volume for species s , and $e_{f,s}$ is specific energy of formation of species, s . Similarly, using integration with respect to temperature, specific enthalpy of species s is given by [71]:

$$h_s(T) = \int_{T_{ref}}^T C_{p,s} dT + h_{f,s}, \quad (2.22)$$

where $h_s(T)$ denotes the specific enthalpy of species s at temperature T , $h_{f,s}$ is specific enthalpy of formation of species s . $e_{f,s}$ has the following equation using $h_{f,s}$:

$$e_{f,s} = h_{f,s} - R_s T_{ref}, \quad (2.23)$$

where R_s is the gas constant of species, s . Its relationship to specific heat capacities is given by the following equation:

$$R_s = C_{p,s} - C_{v,s}, \quad (2.24)$$

where $C_{p,s}$ and $C_{v,s}$ are specific heat for species s at constant pressure and at constant volume, respectively. We obtain the following equation by substituting Equations 2.22, 2.23 and 2.24 into Equation 2.21:

$$\begin{aligned} e_s(T) &= \int_{T_{ref}}^T (C_{p,s} - R_s) dT + e_{f,s} \\ &= \int_{T_{ref}}^T C_{p,s} dT - R_s(T - T_{ref}) + h_{f,s} - R_s T_{ref} \\ &= h_s(T) - h_{f,s} - R_s(T - T_{ref}) + h_{f,s} - R_s T_{ref} \\ &= h_s(T) - R_s T. \end{aligned} \quad (2.25)$$

When modeling multi-species gas, it is common practice to compute thermodynamic properties of individual species using curve fits [72]. The NASA Coefficients and Properties (CAP) program [1, 2] provides polynomial curve fits for the specific heat at constant pressure $C_{p,s}$ and specific enthalpy h_s of each species s as functions of temperature:

$$\frac{C_{p,s}(T)}{R_s} = a_{1,s}T^{-2} + a_{2,s}/T + a_{3,s} + a_{4,s}T + a_{5,s}T^2 + a_{6,s}T^3 + a_{7,s}T^4, \quad (2.26)$$

$$\frac{h_s(T)}{R_s T} = -a_{1,s}T^{-2} + a_{2,s}(\ln T)/T + a_{3,s} + a_{4,s}T/2 + a_{5,s}T^2/3 + a_{6,s}T^3/4 + a_{7,s}T^4/5 + b_{1,s}/T, \quad (2.27)$$

where the coefficients $a_{i,s}$ and $b_{j,s}$ vary with temperature. These polynomials use a reference pressure of $P_{ref} = 1 \times 10^5$ Pa and a reference temperature of $T_{ref} = 298.15$ K. Additionally,

the program provides the specific enthalpy of formation $h_{f,s}$ of species s given at T_{ref} . Equations 2.26 and 2.27 allow for accurate computation of thermodynamic properties of gases, particularly for high-temperature simulations where the calorically imperfect model is necessary.

The energy of molecules is composed of four parts of internal energy [73]:

$$e_s(T) = e_{t,s}(T) + e_{r,s}(T) + e_{v,s}(T) + e_{e,s}(T), \quad (2.28)$$

where $e_{t,s}(T)$ is the translational energy due to the random motion and collision of the gas molecules, $e_{r,s}(T)$ is rotational energy of the molecule about its axes, $e_{v,s}(T)$ is the vibrational energy from molecular vibrations, and $e_{e,s}(T)$ is the energy due to electronic excitation of species s at temperature T [74]. In low-temperature conditions, vibrational and electronic effects are negligible, and these components are omitted in the calorically perfect model. However, for high-temperature or high-speed flows, the contributions of vibrational and electronic energies become significant, making it essential to use a calorically imperfect model. The NASA Coefficients and Properties (CAP) program is based on the calorically imperfect assumption. It builds on the work of McBride and Gordon [75], who utilized statistical mechanics to include vibrational and electronic effects in addition to translational and rotational energies. Therefore, when using the NASA CAP polynomials for calculating specific enthalpy $h_s(T)$ (as in Equation 2.27), the four components of internal energy (as expressed in Equation 2.28) are considered. Consequently, the NASA CAP polynomials are well-suited for modeling calorically imperfect gases.

Equations 2.26 and 2.27 are rewritten as:

$$C_{p,s}(T) = (a_1 T^{-2} + a_2/T + a_3 + a_4 T + a_5 T^2 + a_6 T^3 + a_7 T^4) R_s \quad (2.29)$$

$$h_s(T) = (-a_1 T^{-2} + a_2(\ln T)/T + a_3 + a_4 T/2 + a_5 T^2/3 + a_6 T^3/4 + a_7 T^4/5 + b_1/T) R_s T. \quad (2.30)$$

Given that we have two methods to compute specific internal energy in Equations 2.18 and 2.25, we can compute temperature implicitly by solving the following equation:

$$\tilde{e}(T) = \sum_{s=1}^{n_{species}} Y_s c_s(T) = \sum_{s=1}^{n_{species}} Y_s (h_s(T) - R_s T). \quad (2.31)$$

2.2.2 NonDimensional Form

One significant reason to employ nondimensional properties in Computational Fluid Dynamics (CFD) is to facilitate comparisons with benchmark solutions or to calibrate against key dimensionless parameters [76]. In dimensional form, the variables and coefficients in the equations can vary widely in magnitude. Such disparities can lead to ill-conditioning of the Jacobian matrices formed to solve the linear system at each nonlinear iteration. Nondimensionalization typically normalizes these variables, bringing them within a similar range of values. This reduction in the range of values improves the conditioning of the system by ensuring that the numerical algorithms operate on values that are neither too large nor too small, reducing the possibility of round-off and subtractive cancellation errors and thus improving numerical stability. This research aligns the method of nondimensionalization with the approach of Masatsuka [77]. The nondimensional quantities used in this study are as follows (nondimensionalization also applies to mixture properties; for example, $()_s$ can be

replaced with $\tilde{()})$:

$$\begin{aligned}
\rho^* &= \frac{\rho}{\rho_{ref}}, \quad u^* = \frac{u}{u_{ref}}, \quad v^* = \frac{v}{u_{ref}}, \quad P^* = \frac{P}{\rho_{ref} u_{ref}^2}, \quad x^* = \frac{x}{L_{ref}}, \quad y^* = \frac{y}{L_{ref}}, \quad t^* = \frac{t}{L_{ref}/u_{ref}}, \\
\rho_s^* &= \frac{\rho_s}{\rho_{ref}}, \quad k^* = \frac{k}{u_{ref}^2}, \quad P_s^* = \frac{P_s}{\rho_{ref} u_{ref}^2}, \quad E_s^* = \frac{E_s}{u_{ref}^2}, \quad H_s^* = \frac{H_s}{u_{ref}^2}, \quad e_s^* = \frac{e_s}{u_{ref}^2}, \quad h_s^* = \frac{h_s}{u_{ref}^2}, \\
R_s^* &= \frac{R_s}{R_{ref}}, \quad C_{p,s}^* = \frac{C_{p,s}}{R_{ref}}, \quad C_{v,s}^* = \frac{C_{v,s}}{R_{ref}}, \quad \gamma_s^* = \frac{\gamma_s}{\gamma_{ref}}, \\
T_{ref}^* &= \frac{T_{ref}}{T_{ref}} = 1, \quad e_{tr,s}^* = \frac{e_{tr,s}}{u_{ref}^2}, \quad e_{v,s}^* = \frac{e_{v,s}}{u_{ref}^2}, \quad e_{e,s}^* = \frac{e_{e,s}}{u_{ref}^2}, \quad e_{f,s}^* = \frac{e_{f,s}}{u_{ref}^2}, \\
a_s^* &= \frac{a_s}{u_{ref}}, \quad M_s^* = M_s, \quad Y_s^* = Y_s, \quad tol^* = tol, \quad err^* = err, \quad dim^* = dim, \quad vel2^* = \frac{vel2}{u_{ref}^2}, \\
L^* &= \frac{L}{L_{ref}}, \quad N^* = N
\end{aligned} \tag{2.32}$$

Using nondimensionalizations in Equation 2.32, the dimensional multi-species Euler equations and related equations described in the previous section are nondimensionalized as follows. The nondimensional multi-species Euler equations (2D) are:

$$\frac{\partial \mathbf{Q}^*}{\partial t^*} + \frac{\partial \mathbf{E}^*}{\partial x^*} + \frac{\partial \mathbf{F}^*}{\partial y^*} = 0 \tag{2.33}$$

where

$$\mathbf{Q}^* = \begin{pmatrix} \tilde{\rho}^* \\ \tilde{\rho}^* u^* \\ \tilde{\rho}^* v^* \\ \tilde{\rho}^* \tilde{E}^* \\ \rho_s^* \end{pmatrix}, \quad \mathbf{E}^* = \begin{pmatrix} \tilde{\rho}^* u^* \\ \tilde{\rho}^* (u^*)^2 + \tilde{P}^* \\ \tilde{\rho}^* u^* v^* \\ \tilde{\rho}^* u^* \tilde{H}^* \\ \rho_s^* u^* \end{pmatrix}, \quad \mathbf{F}^* = \begin{pmatrix} \tilde{\rho}^* v^* \\ \tilde{\rho}^* u^* v^* \\ \tilde{\rho}^* (v^*)^2 + \tilde{P}^* \\ \rho^* v^* \tilde{H}^* \\ \rho_s^* v^* \end{pmatrix}, \quad s = 0 : n_{species} - 2 \tag{2.34}$$

The equations introduced in the previous section will now be expressed using the nondimensionalization outlined in Equation 2.32. For example, the nondimensional version of

Equation 2.19 is given by:

$$H_s^* u_{ref}^2 = E_s^* u_{ref}^2 + \frac{P_s^* \rho_{ref} u_{ref}^2}{\rho_s^* \rho_{ref}}. \quad (2.35)$$

After canceling out the nondimensional terms, Equation 2.35 simplifies to:

$$H_s^* = E_s^* + \frac{P_s^*}{\rho_s^*}. \quad (2.36)$$

This resulting equation no longer includes reference values and resembles the dimensional form presented in Equation 2.19. Similarly, Equations 2.14, 2.15, 2.16, 2.17, 2.18, 2.24, and 2.28 can be used by simply substituting dimensional properties by nondimensional properties.

However, in Equations 2.20, 2.21, 2.22, 2.23, 2.25, and 2.31, certain reference values remain even after the nondimensionalization process described in Equation 2.32. For example, the equation of state (Equation 2.20) transforms into its nondimensional form as follows:

$$P_s^* = \rho_s^* R_s^* T^* \left(\frac{R_{ref} T_{ref}}{u_{ref}^2} \right) = \rho_s^* R_s^* T^* \left(\frac{1}{\gamma_{ref} M_{ref}^2} \right), \quad (2.37)$$

where ρ_s^* is nondimensional density for species s , R_s^* is nondimensional gas constant for species s , T^* is nondimensional temperature, R_{ref} is reference gas constant, T_{ref} is reference temperature, u_{ref} is reference velocity, γ_{ref} is reference specific heat ratio, and M_{ref} is the reference Mach number. Hence, some reference values are still present in the nondimensionalized equation. The integration form of nondimensional specific internal energy for species s is derived from Equations 2.21 and 2.32, and is expressed as:

$$e_s^*(T^*) = \left(\frac{R_{ref} T_{ref}}{u_{ref}^2} \right) \int_{T_{ref}^*}^{T^*} C_{v,s}^* dT^* + e_{f,s}^*, \quad (2.38)$$

where T_{ref}^* is nondimensional reference temperature, $C_{v,s}^*$ is nondimensional specific heat constant at constant pressure, and $e_{f,s}^*$ is nondimensional specific energy of formation of species s . Using Equation 2.32, specific enthalpy of species s in Equation 2.22 is non-denationalized

as:

$$h_s^*(T^*) = \left(\frac{R_{ref} T_{ref}}{u_{ref}^2} \right) \int_{T_{ref}^*}^{T^*} C_{p,s}^* dT^* + h_{f,s}^*, \quad (2.39)$$

where $h_s^*(T)$ denotes nondimensional specific enthalpy of species s at nondimensional temperature T^* , $h_{f,s}^*$ is nondimensional specific enthalpy of formation of species s . Based on Equation 2.23, $e_{f,s}^*$ has the following equation using $h_{f,s}^*$:

$$e_{f,s}^* = h_{f,s}^* - \left(\frac{R_{ref} T_{ref}}{u_{ref}^2} \right) R_s^* T_{ref}^*, \quad (2.40)$$

where R_s^* is the gas constant of species s . We obtain the following equation about nondimensional internal energy for species s by substituting Equations 2.39, 2.40, and nondimensional version of 2.24 into Equation 2.38:

$$e_s^*(T^*) = h_s^*(T^*) - \left(\frac{R_{ref} T_{ref}}{u_{ref}^2} \right) R_s^* T^* \quad (2.41)$$

Nondimensional specific enthalpy in Equation 2.41 is derived from Equation 2.27:

$$h_s^*(T^*) = \frac{h_s(T)}{R_{ref}}, \quad (2.42)$$

where the NASA CAP [1] [2] database is used in the computation of dimensional specific enthalpy $h_s(T)$ and the dimensional temperature in Equation 2.42 is derived from nondimensional temperature as:

$$T = T_{ref} T^*, \quad (2.43)$$

where T is dimensional temperature, T_{ref} is reference temperature, and T^* is nondimensional temperature. Hence, $e_s^*(T^*)$ on the left hand side of Equation 2.41 are obtained using Equations 2.42 and 2.43.

The Nondimensional and mixture version of Equation 2.18 gives the following relationship:

$$\tilde{e}^* = \tilde{E}^* + k^*, \quad (2.44)$$

where $\tilde{e}^*$ is the nondimensional mixture specific internal energy, $\tilde{E}^*$ is the nondimensional mixture specific total energy, and k^* is the nondimensional kinetic energy. On the other hand, $\tilde{e}^*(T^*)$ is derived by summing $e_s^*(T^*)$ in Equation 2.41 over all species:

$$\begin{aligned}\tilde{e}^*(T^*) &= \sum_{s=1}^{n_{species}} Y_s^* e_s^*(T^*) \\ &= \sum_{s=1}^{n_{species}} Y_s^* \left[h_s^*(T^*) - \left(\frac{R_{ref} T_{ref}}{u_{ref}^2} \right) R_s^* T^* \right],\end{aligned}\tag{2.45}$$

where Y_s^* represents the mass fraction of species s . Therefore, by combining Equations 2.44 and 2.45, the following equation holds:

$$\tilde{E}^* + k^* = \sum_{s=1}^{n_{species}} Y_s^* \left[h_s^*(T^*) - \left(\frac{R_{ref} T_{ref}}{u_{ref}^2} \right) R_s^* T^* \right]\tag{2.46}$$

In Equation 2.46, nondimensional temperature is implicitly computed using the Newton-Raphson method. Note that the derivative of $h_s^*(T^*)$ with respect to T^* is required to perform the Newton-Raphson update, and it is derived from Equation 2.39 as:

$$(h_s^*(T^*))' = \left(\frac{R_{ref} T_{ref}}{u_{ref}^2} \right) C_{p,s}^*(T^*),\tag{2.47}$$

where the nondimensional specific heat at constant pressure for species s , $C_{p,s}^*(T^*)$, is given by:

$$C_{p,s}^*(T^*) = \frac{C_{p,s}(T)}{R_{ref}},\tag{2.48}$$

where the dimensional specific heat at constant pressure for species s , $C_{p,s}(T)$, is computed using the NASA CAP polynomials as given in Equation 2.29. The dimensional temperature T is obtained using the relation in Equation 2.43.

2.2.3 Algorithm

To implement a calorically imperfect multi-species model to a calorically perfect mono-species discontinuous Galerkin (DG) solver, we need to introduce functions to compute nondi-

mensional multi-species convective fluxes $\mathbf{E}^*$ and $\mathbf{F}^*$ including $(\tilde{\rho}^*, u^*, v^*, \tilde{P}^*, \tilde{H}^*, \text{ and } \rho_s^*)$ from nondimensional multi-species conservative solutions ($\mathbf{Q}^*$) under calorically imperfect assumptions. Nondimensional multi-species conservative solutions yield the nondimensional multi-species convective fluxes when all main functions denoted as f_{M_i} are computed. The supporting functions denoted as f_{S_j} are used for the pre-processing or the post-processing.

In Algorithm 1, the nondimensional mixture density $\tilde{\rho}^*$ is extracted as the first element of $\mathbf{Q}^*$. Algorithm 2, calculates the nondimensional velocities u^* and v^* by dividing the nondimensional mixture momentum by $\tilde{\rho}^*$ obtained in Algorithm 1. The squared sum of these velocities is computed in Algorithm 3, and the nondimensional kinetic energy k^* is evaluated in Algorithm 4. Nondimensional mixture total energy $\tilde{E}^*$ is directly computed from conservation of mixture energy in $\mathbf{Q}^*$ using Algorithm 5. The nondimensional species density ρ_s^* is also directly obtained from conservation of species mass in Algorithm 6. Algorithm 7 calculates the mass fraction of species s , using $\tilde{\rho}^*$ and ρ_s^* , while mixture property is obtained in Algorithm 8. Dimensional temperature T is given in Algorithm 9 using reference temperature T_{ref} and nondimensional temperature T^* . We then compute the nondimensional species gas constants R_s^* in Algorithm 10. The NASA CAP database provides $C_{p,s}^*$, $C_{v,s}^*$, and h_s^* as functions of temperature in Algorithms 11, 12, and 13, respectively. Specific internal energy for species s is determined in Algorithm 14 using temperature. However, we need to find the nondimensional temperature T^* through a root-finding algorithm in Algorithm 15. After computing the nondimensional mixture gas constant $\tilde{R}^*$ from R_s^* and Y_s^* in Algorithm 16, the nondimensional mixture pressure $\tilde{P}$ can be calculated using the equation of state in Algorithm 17. Using $\tilde{P}$, the nondimensional mixture total enthalpy $\tilde{H}^*$ is obtained in Algorithm 18. Finally, the nondimensional multi-species convective fluxes are computed in Algorithm 19 using $\tilde{\rho}^*, u^*, v^*, \tilde{P}^*, \tilde{H}^*, \text{ and } \rho_s^*$.

The primitive solutions are given as:

$$\mathbf{P}^* = \begin{pmatrix} \tilde{\rho}^* \\ u^* \\ v^* \\ \tilde{P}^* \\ Y_s^* \end{pmatrix}, s = 0 : n_{species} - 2; \quad (2.49)$$

where $\tilde{\rho}^*$ is the nondimensional mixture density, u^* and v^* denote nondimensional velocities, $\tilde{P}^*$ represent nondimensional mixture pressure, and Y_s^* is the mass fraction for species s . The algorithms of all implemented functions to compute calorically imperfect multi-species gas are summarized below.

Algorithm 1 Compute mixture density $\tilde{\rho}^*$: f_{M_1}

Input: Conservative solutions: $\mathbf{Q}^*$

Output: Mixture density: $\tilde{\rho}^*$

- 1: Compute mixture density: $\tilde{\rho}^* = \mathbf{Q}^*[0]$
 - 2: Return: $\tilde{\rho}^*$
-

Algorithm 2 Compute velocities u^* and v^* : f_{M_2}

Input: Conservative solutions: $\mathbf{Q}^*$

Output: Velocities: u^* and v^*

- 1: Compute mixture density: $\tilde{\rho}^* = \text{compute_mixture_density}(\mathbf{Q}^*)$
 - 2: Compute velocities: $u^* = \frac{\mathbf{Q}^*[1]}{\tilde{\rho}^*}$ and $v^* = \frac{\mathbf{Q}^*[2]}{\tilde{\rho}^*}$
 - 3: Return: u^* and v^*
-

Algorithm 3 Compute squared velocities $vel2^*$: f_{M_3}

Input: Conservative solutions: $\mathbf{Q}^*$

Output: Squared velocities: $vel2^*$

- 1: Compute velocities: u^* and $v^* = \text{compute_velocities}(\mathbf{Q}^*)$
 - 2: Compute squared velocities: $vel2^* = (u^*)^2 + (v^*)^2$
 - 3: Return: $vel2^*$
-

Algorithm 4 Compute specific kinetic energy k^* : f_{M_4}

Input: Conservative solutions: $\mathbf{Q}^*$ **Output:** Specific kinetic energy: k^*

- 1: Compute squared velocities: $vel2 = \text{compute_squared_velocities}(\mathbf{Q}^*)$
 - 2: Compute specific kinetic energy: $k^* = \frac{1}{2}vel2^*$
 - 3: Return: k^*
-

Algorithm 5 Compute mixture specific total energy $\tilde{E}^*$: f_{M_5}

Input: Conservative solutions: $\mathbf{Q}^*$ **Output:** Mixture specific total energy: $\tilde{E}^*$

- 1: Compute mixture density: $\tilde{\rho}^* = \text{compute_mixture_density}(\mathbf{Q}^*)$
 - 2: Compute mixture specific total energy: $\tilde{E}^* = \frac{\mathbf{Q}^*[n_{state}-1]}{\tilde{\rho}^*}$
 - 3: Return: $\tilde{E}^*$
-

Algorithm 6 Compute species densities ρ_s^* : f_{M_6}

Input: Conservative solutions: $\mathbf{Q}^*$ **Output:** Species densities: $\rho_s^*, s = 0 : n_{species} - 1$

- 1: Compute mixture density: $\tilde{\rho}^* = \text{compute_mixture_density}(\mathbf{Q}^*)$
 - 2: Set summation: $sum^* = 0.0$
 - 3: Compute $(0 : n_{species} - 2)$ th species densities and their sum:
 - 4: **for** $i = n_{state} : n_{state} + (n_{species} - 2)$ **do**
 - 5: $s = i - n_{state}$
 - 6: $\rho_s^* = \mathbf{Q}^*[i]$
 - 7: $sum^* += \rho_s^*$
 - 8: **end for**
 - 9: Compute $n_{species} - 1$ th species density: $\rho_s^* = \tilde{\rho}^* - sum^*$
 - 10: Return $\rho_s^*, s = 0 : n_{species} - 1$
-

Algorithm 7 Compute mass fractions: Y_s^* : f_{M_7}

Input: Conservative solutions: $\mathbf{Q}^*$ **Output:** Mass fraction: Y_s^* , $s = 0 : n_{species} - 1$

- 1: Compute mixture density $\tilde{\rho}^*$: $\tilde{\rho}^* = \text{compute_mixture_density}(\mathbf{Q}^*)$
 - 2: Compute species densities ($s = 0 : n_{species} - 1$): $\rho_s^* = \text{compute_species_density}(\mathbf{Q}^*)$
 - 3: Compute mass fractions ($s = 0 : n_{state} - 1$) th
 - 4: **for** $s = 0 : n_{species} - 1$ **do**
 - 5: Compute mass fractions for $s = 0 : n_{species} - 1$ th species: $Y_s^* = \frac{\rho_s^*}{\tilde{\rho}^*}$
 - 6: **end for**
 - 7: Return: Y_s^* , $s = 0 : n_{species} - 1$
-

Algorithm 8 Compute mixture from species: property_s^* : f_{M_8}

Input: Mass fractions and species values: Y_s^* and property_s^* , $s = 0 : n_{species} - 1$ **Output:** Mixture value: property^*

- 1: Set summation: $\text{sum}^* = 0.0$
 - 2: **for** $s = 0 : n_{species} - 1$ **do**
 - 3: $\text{sum}^* += Y_s^* \text{property}_s^*$
 - 4: **end for**
 - 5: Return: $\text{property}^* = \text{sum}^*$
-

Algorithm 9 Compute dimensional temperature T : f_{M_9}

Input: Temperature: T^* **Output:** Dimensional temperature: T

- 1: Compute dimensional temperature: $T = T^* T_{ref}$
 - 2: Return: T
-

Algorithm 10 Compute species gas constants R_s^* : $f_{M_{10}}$

Input: Universal gas constant: R_u^* **Output:** Species gas constants: R_s^* , $s = 0 : n_{species} - 1$

- 1: **for** $s = 0 : n_{species} - 1$ **do**
 - 2: Compute species gas constants:
 $R_s = \frac{R_u}{MW_s}$
 $R_s^* = \frac{R_s}{R_{ref}}$
 - 3: **end for**
 - 4: Return R_s^* , $s = 0 : n_{species} - 1$
-

Algorithm 11 Compute species specific heat at constant pressure $C_{p,s}^*$: f_{M11}

Input: Temperature: T^* **Output:** Species specific heat at constant pressure: $C_{p,s}^*, s = 0 : n_{species} - 1$

- 1: Compute dimensional temperature: $T = \text{compute_dimensional_temperature}(T^*)$
 - 2: **for** $s = 0 : n_{species} - 1$ **do**
 - 3: Compute dimensional species specific heat at constant pressure using NASA CAP:
 $C_{p,s} = f_{NASA-CAP_{C_{p,s}}}(T)R_s$
 - 4: Compute nondimensional specific heat at constant pressure: $C_{p,s}^* = \frac{C_{p,s}}{R_{ref}}$
 - 5: **end for**
 - 6: Return $C_{p,s}^*, s = 0 : n_{species} - 1$
-

Algorithm 12 Compute species specific heat at constant volume $C_{v,s}^*$: f_{M12}

Input: Temperature: T^* **Output:** Species specific heat at constant volume $C_{v,s}^*, s = 0 : n_{species} - 1$

- 1: **for** $s = 0 : n_{species} - 1$ **do**
 - 2: Compute species specific heat at constant pressure: $C_{p,s}^* = \text{compute_species_Cp}(T^*)$
 - 3: Compute species specific heat at constant volume: $C_{v,s}^* = C_{p,s}^* - R_s^*$
 - 4: **end for**
 - 5: Return: $C_{v,s}^*, s = 0 : n_{species} - 1$
-

Algorithm 13 Compute species specific enthalpy h_s^* : f_{M13}

Input: Temperature: T^* **Output:** Species specific enthalpy: $h_s^*, s = 0 : n_{species} - 1$

- 1: Compute dimensional temperature: $T = \text{compute_dimensional_temperature}(T^*)$
 - 2: **for** $s = 0 : n_{species} - 1$ **do**
 - 3: Compute dimensional species specific enthalpy using NASA CAP: $h_s = f_{NASA-CAP_{h_s}}(T)R_sT$
 - 4: Compute nondimensional species specific enthalpy: $h_s^* = \frac{h_s}{u_{ref}^2}$
 - 5: **end for**
 - 6: Return $h_s^*, s = 0 : n_{species} - 1$
-

Algorithm 14 Compute species specific internal energy e_s^* : f_{M14}

Input: Temperature: T^* **Output:** Species specific internal energy: $e_s^*, s = 0 : n_{species} - 1$

- 1: **for** $s = 0 : n_{species} - 1$ **do**
 - 2: Compute species specific enthalpy using: $h_s^* = \text{compute_species_specific_enthalpy}(T^*)$
 - 3: Compute species specific internal energy: $e_s^* = h_s^* - \frac{R_{ref}T_{ref}}{u_{ref}^2}R_s^*T^*$
 - 4: **end for**
 - 5: Return $e_s^*, s = 0 : n_{species} - 1$
-

Algorithm 15 Compute temperature T^* : f_{M15}

Input: Conservative solutions: Q^* **Output:** Temperature: T^*

- 1: Compute mixture specific total energy using:
 $\tilde{E}^* = \text{compute_mixture_specific_total_energy}(Q^*)$
 - 2: Compute specific kinetic energy: $k^* = \text{compute_specific_kinetic_energy}(Q^*)$
 - 3: Compute mass fraction: $Y_s^* = \text{compute_mass_fraction}(Q^*)$
 - 4: Compute mixture density: $\tilde{\rho}^* = \text{compute_mixture_density}(Q^*)$
 - 5: Compute mixture gas constant: $\tilde{R}^* = \text{compute_mixture_gas_constant}(Q^*)$
 - 6: Compute initial mixture pressure: $\tilde{P}_0^* = \tilde{\rho}^*(\gamma_{ref}^* - 1)(\tilde{E}^* - k^*)$
 - 7: Set the initial guess temperature: $T_0^* = \frac{\tilde{P}_0^*}{\tilde{\rho}^* \tilde{R}^*}(\gamma_{ref}^* M_{ref}^2)$
 - 8: **while** $err^* > tol^*$ **do**
 - 9: 1): $f^*(T_n^*)$
 - 10: Compute Newton-Raphson function $f^*(T_n^*)$ ($= \tilde{e}^*(T_n^*) - (\tilde{E}^* - k^*)$) in the following way:
 - 11: Compute species specific internal energy at T_n^* :
 - 12: **for** $i = 0 : n_{species} - 1$ **do**
 - 13: Compute species specific internal energy at T_n^* :
 $e_s^*(T_n^*) = \text{compute_species_specific_internal_energy}(T_n^*)$
 - 14: **end for**
 - 15: Compute mixture specific internal energy:
 $\tilde{e}^*(T_n^*) = \text{compute_mixture_from_species}(Y_s^*, e_s^*(T_n^*))$
 - 16: Compute Newton-Raphson function: $f^* = \tilde{e}^*(T_n^*) - (\tilde{E}^* - k^*)$
 - 17: 2): $f^{*'}(T_n^*)$
 - 18: Compute Newton-Raphson derivative function $f^{*'}(T_n^*)$ ($= \tilde{C}_v^*(T_n^*)$) in the following way:
 - 19: **for** $i = 0 : n_{species} - 1$ **do**
 - 20: Compute species specific heat at constant volume at T_n^* :
 $C_{v,s}^*(T_n^*) = \text{compute_species_Cv}(T_n^*)$
 - 21: $\tilde{C}_v^*(T_n^*) = \text{compute_mixture_from_species}(Y_s^*, C_{v,s}^*(T_n^*))$
 - 22: **end for**
 - 23: Compute Newton-Raphson derivative function: $f^{*'} = \tilde{C}_v^*(T_n^*)$
 - 24: 3): Main part
 - 25: Newton-Raphson Method: $T_{n+1}^* = T_n^* - \frac{f^{*'}(T_n^*)}{f^*(T_n^*)}$
 - 26: Compute error: $err = |T_{n+1}^* - T_n^*|$
 - 27: Update temperature: $T_n^* = T_{n+1}^*$
 - 28: **end while**
 - 29: Return: $T^* = T_{n+1}^* = T_n^*$
-

Algorithm 16 Compute mixture gas constant $\tilde{R}^*$: f_{M14}

Input: Conservative solutions: $\mathbf{Q}^*$

Output: Mixture gas constant: $\tilde{R}^*$

- 1: Compute mass fraction: $Y_s^* = \text{compute_mass_fraction}(\mathbf{Q}^*)$
 - 2: Compute mixture gas constant: $\text{compute_mixture_from_species}(Y_s^*, R_s^*)$
 - 3: Return: $\tilde{R}^*$
-

Algorithm 17 Compute mixture pressure $\tilde{P}^*$: f_{M17}

Input: Conservative solutions: $\mathbf{Q}^*$

Output: Mixture pressure: $\tilde{P}^*$

- 1: Compute mixture density: $\tilde{\rho}^* = \text{compute_mixture_density}(\mathbf{Q}^*)$
 - 2: Compute mixture gas constant: $\tilde{R}^* = \text{compute_mixture_gas_constant}(\mathbf{Q}^*)$
 - 3: Compute temperature: $T^* = \text{compute_temperature}(\mathbf{Q}^*)$
 - 4: Compute mixture pressure: $\tilde{P}^* = \tilde{\rho}^* \tilde{R}^* T^* \frac{1}{\gamma_{ref} M_{ref}^2}$
 - 5: Return: $\tilde{P}^*$
-

Algorithm 18 Compute mixture specific total enthalpy $\tilde{H}^*$: f_{M18}

Input: Conservative solutions: $\mathbf{Q}^*$

Output: Mixture specific total enthalpy: $\tilde{H}^*$

- 1: Compute mixture specific total energy: $\tilde{E}^* = \text{compute_mixture_specific_total_energy}(\mathbf{Q}^*)$
 - 2: Compute mixture pressure: $\tilde{P}^* = \text{compute_mixture_pressure}(\mathbf{Q}^*)$
 - 3: Compute mixture density: $\tilde{\rho}^* = \text{compute_mixture_density}(\mathbf{Q}^*)$
 - 4: Compute mixture specific total enthalpy: $\tilde{H}^* = \tilde{E}^* + \frac{\tilde{P}^*}{\tilde{\rho}^*}$
 - 5: Return: $\tilde{H}^*$
-

Algorithm 19 Compute convective flux $\mathbf{E}^*$ and $\mathbf{F}^*$: f_{M19}

Input: Conservative solutions: $\mathbf{Q}^*$ **Output:** Convective flux: $\mathbf{E}^*$ and $\mathbf{F}^*$

```
1: Compute mixture density:  $\tilde{\rho}^* = \text{compute\_mixture\_density}(\mathbf{Q}^*)$ 
2: Compute velocities:  $u^*$  and  $v^*$  ( $= v_{\text{velocity}}$ )  $= \text{compute\_velocities}(\mathbf{Q}^*)$ 
3: Compute mixture pressure:  $\tilde{P}^* = \text{compute\_mixture\_pressure}(\mathbf{Q}^*)$ 
4: Compute mixture specific total enthalpy:
    $\tilde{H}^* = \text{compute\_mixture\_specific\_total\_enthalpy}(\mathbf{Q}^*)$ 
5: Compute species density:  $\rho_s^* = \text{compute\_species\_density}(\mathbf{Q}^*)$ 
6: for  $i = 0 : \text{dim} - 1$  do
7:   A): Mixture density equation
8:    $\mathbf{f}_{lux}^*[0][\text{dim}] = \mathbf{Q}^*[i + 1]$ 
9:   B): Compute momentum equation
10:  for  $j = 0 : \text{dim} - 1$  do
11:     $\mathbf{f}_{lux}^*[1 + j][i] = \tilde{\rho}^* v_{\text{velocity}}^*[i] v_{\text{velocity}}^*[j]$ 
12:  end for
13:   $\mathbf{f}_{lux}^*[1 + j][i] += \tilde{\rho}^*$ 
14:  C): Energy equation
15:   $\mathbf{f}_{lux}^*[n_{\text{state}} - 1][i] = \tilde{\rho}^* v_{\text{velocity}}^*[i] \tilde{H}^*$ 
16:  D) Species density equation
17:  for  $k = n_{\text{state}} : n_{\text{state}} + (n_{\text{species}} - 2)$  do
18:     $s = k - n_{\text{state}}$ 
19:     $\mathbf{f}_{lux}^*[k][i] = \rho_s^*[s] v_{\text{velocity}}^*[i]$ 
20:  end for
21: end for
22:  $\mathbf{E}^* = \mathbf{f}_{lux}^*[:, 0]$ ,  $\mathbf{F}^* = \mathbf{f}_{lux}^*[:, 1]$ 
23: Return:  $\mathbf{E}^*$  and  $\mathbf{F}^*$ 
```

Algorithm 20 Convert primitive to conservative $\mathbf{Q}^*$: $f_{S_{20}}$

Input: Primitive solutions: $\mathbf{P}^*$

Output: Species densities: $\mathbf{Q}^*$

```

1: A) Mixture density
2: Compute mixture density:  $\tilde{\rho}^* = \mathbf{P}^*[0]$ 
3:  $\mathbf{Q}^*[0] = \mathbf{P}^*[0]$ 
4: B) Momentum
5: for  $i = 0 : \dim - 1$  do
6:   Compute velocities:  $vel^*[i] = \mathbf{P}^*[i + 1]$ 
7:   Compute squared velocities:  $vel2^* = vel^*[i]vel^*[i]$ 
8:    $\mathbf{Q}^*[i + 1] = \tilde{\rho}^*vel^*[i]$ 
9: end for
10: C) Energy
11:  $sum^* = 0.0$ 
12: for  $i = \dim + 2 : n_{species} - 2$  do
13:    $s = i - (\dim + 2)$ 
14:   Compute  $(0 : n_{species} - 2)$  th species densities:  $\rho_s^* = \mathbf{P}^*[i]$ 
15:   Compute summation:  $sum^* += \rho_s^*$ 
16: end for
17: Compute  $(n_{state} - 1)$  th species density:  $\rho_s^* = \tilde{\rho}^* - sum^*$ 
18: for  $i = 0 : n_{species} - 1$  do
19:   Compute mass fractions:  $Y_s^* = \frac{\rho_s^*}{\rho^*}$ 
20: end for
21: for  $i = 0 : n_{species} - 1$  do
22:   Compute mixture gas constant:  $\tilde{R}^* = \text{compute\_mixture\_from\_species}(Y_s^*, R_s^*)$ 
23: end for
24: Compute temperature:  $T^* = \frac{\tilde{P}^*}{\tilde{\rho}^* \tilde{R}^*} \frac{u_{ref}^2}{R_{ref} T_{ref}}$ 
25: Compute specific kinetic energy:  $k^* = \frac{1}{2} vel2^*$ 
26: for  $i = 0 : n_{species} - 1$  do
27:   Compute species specific enthalpy:  $h_s^* = \text{compute\_species\_specific\_enthalpy}(T^*)$ 
28:   Compute species specific internal energy:  $e_s^* = h_s^* - R_s^* T^*$ 
29:   Compute species specific total energy:  $E_s^* = e_s^* + k^*$ 
30: end for
31: Compute mixture specific total energy:  $\tilde{E}^* = \text{compute\_mixture\_from\_species}(Y_s^*, E_s^*)$ 
32: D) Species densities
33: for  $s = 0 : n_{species} - 2$  do
34:    $\mathbf{Q}^*[n_{state} - 1 + s] = \rho_s^*$ 
35: end for
36: Return:  $\mathbf{Q}^*$ 

```

Algorithm 21 Compute species specific heat ratio γ_s^* : $f_{S_{21}}$

Input: Conservative solutions: $\mathbf{Q}^*$

Output: Species specific heat ratio: γ_s^* , $s = 0 : n_{species} - 1$

- 1: Compute temperature: $T^* = \text{compute_temperature}(\mathbf{Q}^*)$
 - 2: **for** $s = 0 : n_{species} - 1$ **do**
 - 3: Compute species specific heat at constant pressure: $C_{p,s}^* = \text{compute_species_Cp}(T^*)$
 - 4: Compute species specific heat at constant volume: $C_{v,s}^* = \text{compute_species_Cv}(T^*)$
 - 5: Compute species specific heat ratio: $\gamma_s^* = \frac{C_{p,s}^*}{C_{v,s}^*}$
 - 6: **end for**
 - 7: Return: γ_s^* , $s = 0 : n_{species} - 1$
-

Algorithm 22 Compute species speed of sound a_s^* : $f_{S_{22}}$

Input: Conservative solutions: $\mathbf{Q}^*$

Output: Species specific speed of sound: a_s^* , $s = 0 : n_{species} - 1$

- 1: Compute temperature: $T^* = \text{compute_temperature}(\mathbf{Q}^*)$
 - 2: **for** $s = 0 : n_{species} - 1$ **do**
 - 3: Compute species specific heat ratio: $\gamma_s^* = \text{compute_species_gamma}(T^*)$
 - 4: Compute species speed of sound: $a_s^* = \sqrt{\gamma_s^* R_s^* T^*}$
 - 5: **end for**
 - 6: Return: a_s^* , $s = 0 : n_{species} - 1$
-

2.3 Error Estimation

In order to rigorously test the implemented numerical method, one should create convergence plots. A convergence plot is a tool used to assess the accuracy of a numerical solution by examining how the error decreases as the spatial discretization of the domain is refined. It provides a visual and quantitative way to evaluate the effectiveness of discretization schemes and to ensure that the solution is converging appropriately as the mesh is refined. Error norms measure the difference between the numerical and the exact solution. The most often used norms are [78]:

L_1 error norm

$$\epsilon_1 = \|u_{numerical} - u_{exact}\|_1 = \int_{\Omega} |u_{numerical} - u_{exact}| d\Omega \quad (2.50)$$

L_2 error norm

$$\epsilon_2 = \|u_{numerical} - u_{exact}\|_2 = \left(\int_{\Omega} |u_{numerical} - u_{exact}|^2 d\Omega \right)^{1/2} \quad (2.51)$$

L_∞ error norm

$$\epsilon_\infty = ||u_{numerical} - u_{exact}||_\infty = \max_{\Omega} (|u_{numerical} - u_{exact}|) \quad (2.52)$$

These norms help quantify the accuracy of the numerical method and its convergence properties, providing a comprehensive evaluation of the implemented models. The over-integration approach precisely computes L_1 and L_2 norms by involving additional Gauss quadrature points, ensuring accurate error measurement and reliable convergence analysis. To create convergence plots, one needs the exact solution of the problem. Therefore, convergence plots are based on data from numerical solutions of problems that have well-known analytic solutions [78]. In our research, since the analytic solutions are unavailable in the verification test cases, we will run the case for one cycle to use the initial solutions as the exact solutions. This approach allows us to rigorously test the accuracy and convergence of the numerical method despite the absence of explicit analytic solutions. By introducing the ratio r of coarse to fine mesh element spacing, the observed order of accuracy p becomes [79]:

$$p = \frac{\ln\left(\frac{\epsilon_{p,2}}{\epsilon_{p,1}}\right)}{\ln(r)}, \quad (2.53)$$

where $\epsilon_{p,1}$ and $\epsilon_{p,2}$ indicate the L_p error norm for the fine mesh and coarse mesh, respectively.

Chapter 3

Verification

This section presents a series of test cases to confirm the applicability of the multi-species model and the calorically imperfect model. Verification is demonstrated in one-dimensional and two-dimensional domains. A high-temperature test is conducted in addition to the low-temperature test to demonstrate the applicability of the calorically imperfect physics model. The order of accuracy is studied for low and high polynomial orders. The accuracy of the calorically imperfect multi-species Euler equations solver is discussed by comparing the computed order with the theoretical order as well as plots obtained in the previous study.

3.1 1D Low-Temperature Vortex Advection

The accuracy of the multi-species model is studied in a one-dimensional domain through a series of experiments with different polynomial orders and numbers of elements. The test is solved using calorically imperfect multi-species Euler equations. The initial conditions and computational domain of this test case are chosen based on the numerical test by Wang et al. [80].

3.1.1 Problem Statement

The initial conditions of this 3-species (H_2 , O_2 , and N_2) test are given by the following equations:

$$\begin{aligned} T &= T_0 - \frac{(\gamma_0 - 1) \Gamma^2}{8\gamma_0\pi} \left(\frac{1 - r^2}{2} \right), \\ Y_{H_2} &= Y_{H_2,0} - \frac{2\pi a_1}{\gamma_0\Gamma} \exp\left(\frac{1 - r^2}{2} \right), \\ Y_{O_2} &= Y_{O_2,0} - \frac{2\pi a_2}{\gamma_0\Gamma} \exp\left(\frac{1 - r^2}{2} \right), \\ Y_{N_2} &= 1 - (Y_{H_2} + Y_{O_2}), \end{aligned} \tag{3.1}$$

where $T_0 = 300$ K, $\Gamma = 50$, $\gamma_0 = 1.4$, $Y_{H_2,0} = 0.01277$, $Y_{O_2,0} = 0.101$, $a_1 = 0.005$, $a_2 = 0.03$, and $r = x - x_0$; where x_0 is the centre of the physical domain. The uniform initial mixture pressure and velocity are $P = 101325$ Pa and $u = 100$ m/s, respectively. Temperature and mass fractions are given by a Gaussian profile in Equation 3.1. Mixture density is obtained from temperature and mixture pressure using the equation of state. The computational domain length is $L = 10$ m and the center of the domain is located at $x_0 = 5$ m. Periodic boundary conditions are applied such that the flow advects through the domain every 0.1 s.

The initial conditions are visualized in Figure 3.1. Gaussian profiles are depicted in Figures 3.1a, 3.1b, and 3.1c. The temperature ranges from 250 K to 300 K, as seen in Figure 3.1c, where both calorically perfect and imperfect models can be applied. Plots after 0.1 cycle (0.01 s) are shown in Figures 3.2 and 3.3 to confirm the advection of the vortex.

3.1.2 Results

L_1 , L_2 , and L_∞ error norms of mixture density and mass fraction of hydrogen are computed after 1 cycle ($t = 0.1$ s) by comparing results against the initial conditions, which are considered as the exact solutions. To demonstrate the capability of computing different polynomial orders, tests are conducted for $p = 2$ and $p = 5$, aligned with Wang et al. [80]. The number of elements is increased from $N = 2$ to 64 for the $p = 2$ test and to $N = 16$ for the $p = 5$ test until reasonable convergence of the error norm slope is reached. The ratio

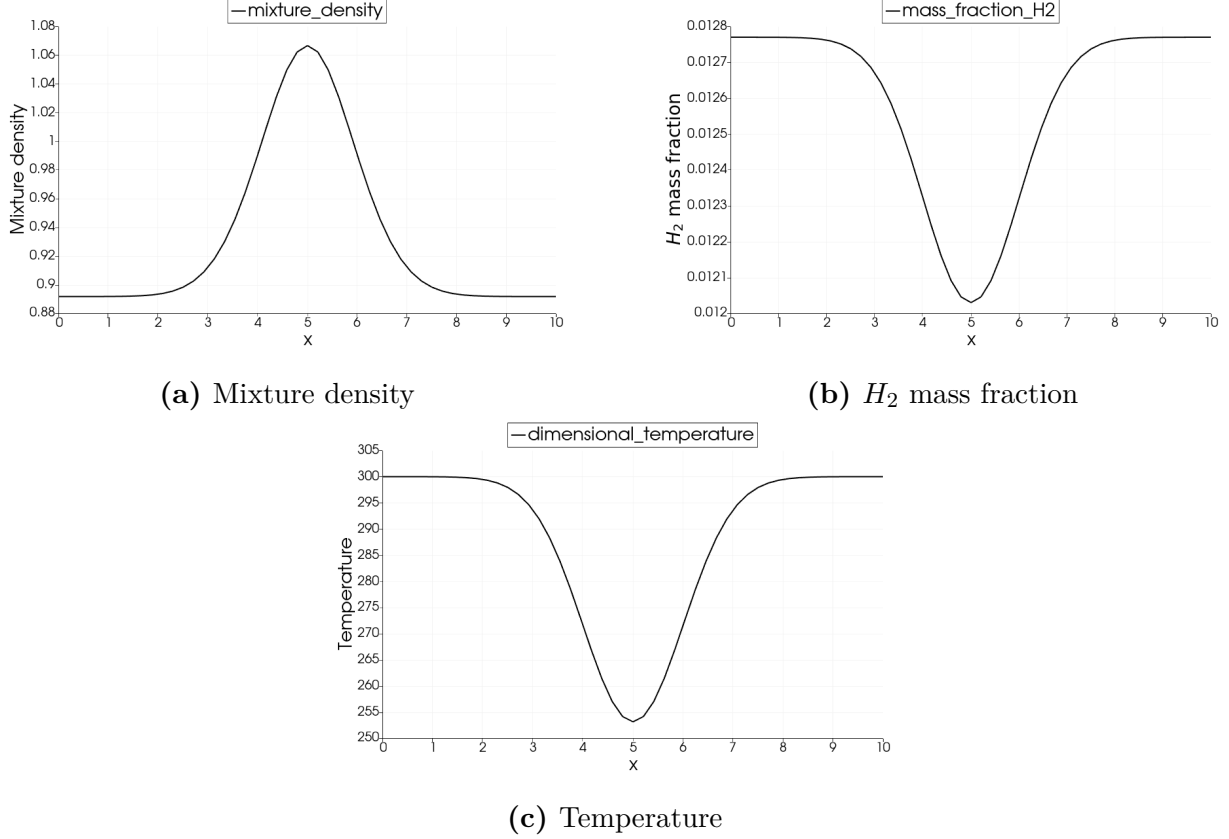
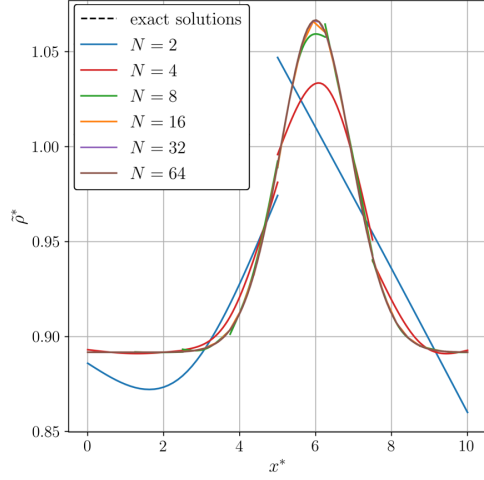


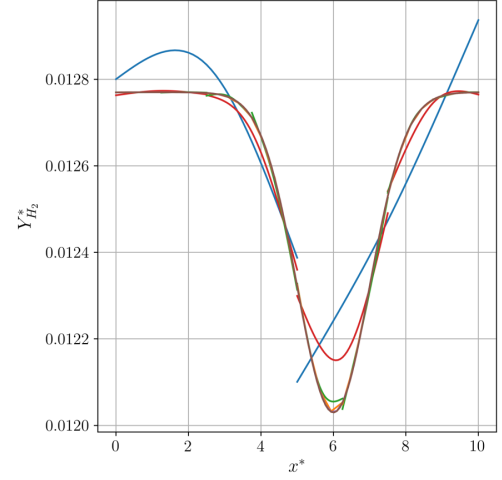
Figure 3.1: Profiles of mixture density, H_2 mass fraction, and temperature at the initial time for the 1D low-temperature vortex advection problem.

of coarse to fine mesh element spacing is set to $r = 2$ in both polynomial order cases. To minimize temporal errors, a very small time step, $\Delta t = 1 \times 10^{-6}$ s, is used for all grid sizes.

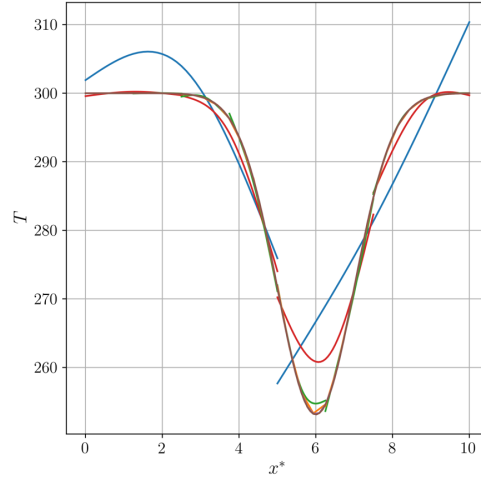
In Figure 3.2, flow profiles are drawn using a $p2$ approximation with varying number of elements. When the number of elements is small ($N = 2$ and $N = 4$), the flow profiles deviate significantly from the exact solution, and discontinuities between neighbouring elements, which are inherent to the discontinuous Galerkin method are clearly visible. As N increases, the results converge toward the exact solution. For $N \geq 32$, the resolution appears sufficient, as the results closely match the exact solution. Figure 3.3 illustrates the results using a $p5$ approximation. When $N = 2$, the results exhibit fluctuations, making them unsuitable for simulations. However, for $N \geq 4$, the computations yield accurate results. Comparing



(a) Mixture density



(b) H_2 mass fraction

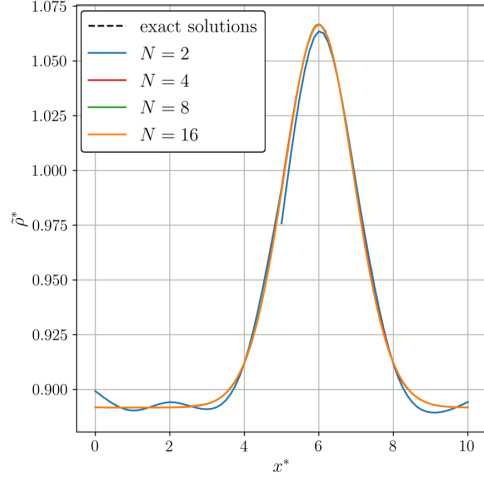


(c) Temperature

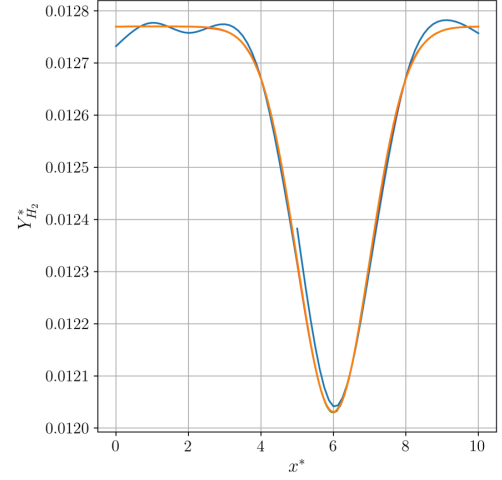
Figure 3.2: Profiles of mixture density, H_2 mass fraction, and temperature after 0.1 cycle using a $p2$ approximation for the 1D low-temperature vortex advection problem.

Figures 3.2 and 3.3, it is evident that a higher polynomial order provides more accurate results for the same number of elements.

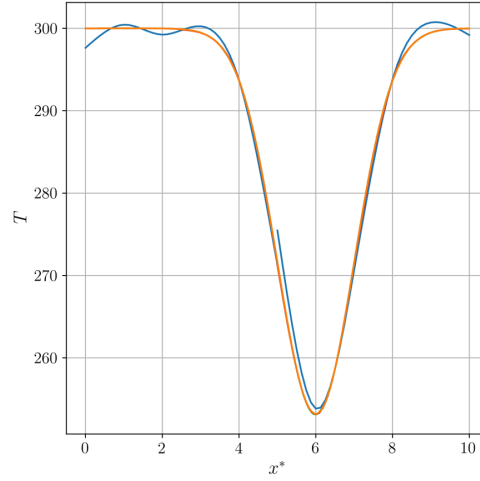
Figure 3.4 compares results using the same number of DOFs but at two different polynomial degrees. Results with the $p2$ approximation show some deviation from the exact solution, whereas those using $p5$ are very close to the exact solutions. This indicates that a higher polynomial degree yields more accurate results at equivalent DOFs, highlighting the advantage of higher-order computations.



(a) Mixture density



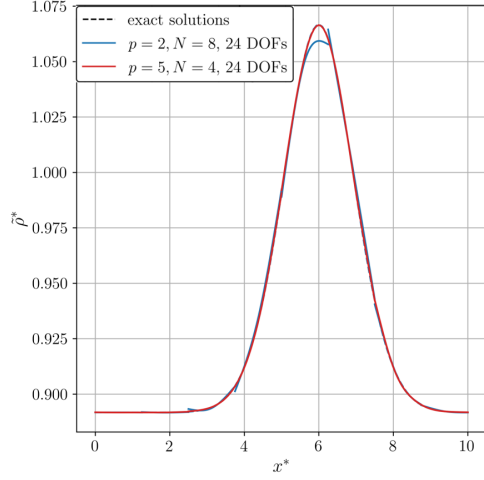
(b) H_2 mass fraction



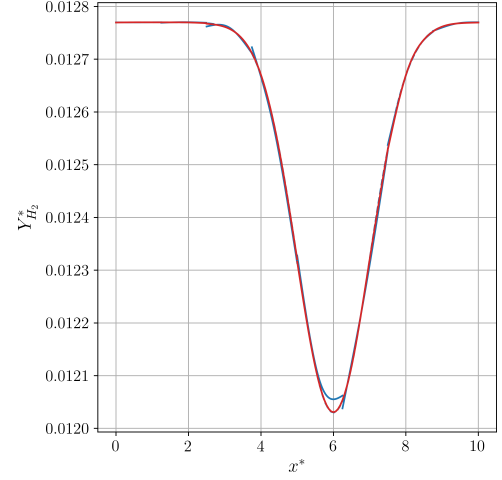
(c) Temperature

Figure 3.3: Profiles of mixture density, H_2 mass fraction, and temperature after 0.1 cycle using a $p5$ approximation for the 1D low-temperature vortex advection problem.

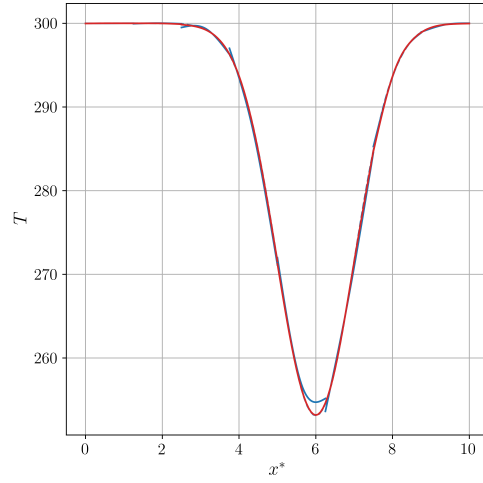
Wang et al. [80] propose using a very small time step, $\Delta t = 1 \times 10^{-6}$ s. To validate this choice, an additional test was conducted with a time step one-order magnitude higher. Figure 3.5 compares results obtained using $\Delta t = 1 \times 10^{-6}$ and $\Delta t = 1 \times 10^{-5}$. To assess the impact of varying the time step, the finest grid, corresponding to the largest number of elements (DOFs = 192 with $p = 2$ and $N = 64$), was used, as this would ensure that the temporal discretization error from the time-stepping scheme is smaller than the lower bound



(a) Mixture density



(b) H_2 mass fraction



(c) Temperature

Figure 3.4: Profiles of mixture density, H_2 mass fraction, and temperature after 0.1 cycle using $p2$ and $p5$ approximations using different grid sizes such that DOFs are equivalent for the 1D low-temperature vortex advection problem.

of the spatial discretization errors for the range of grids and polynomial orders investigated. The results from both time steps are very similar, indicating $\Delta t = 1 \times 10^{-6}$ is appropriate.

Figures 3.6, 3.7, and 3.8 show the L_1 , L_2 , and L_∞ error norms of $\tilde{\rho}^*$ and $Y_{H_2}^*$, as well as the theoretical slopes with respect to L^*/N^* . According to these figures, the error norm plots obtained from the $p = 2$ test and the $p = 5$ test closely match the predicted convergence rate as L^*/N^* decreases (N^* increases) for both test cases. This indicates that the correct

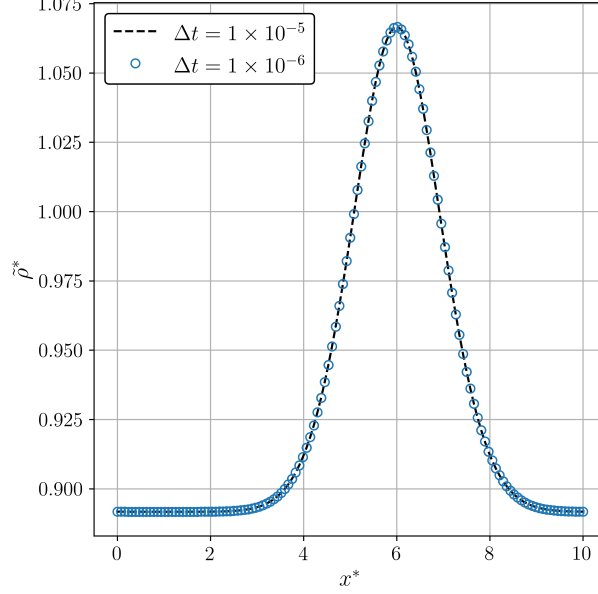


Figure 3.5: Profiles of mixture density after 0.1 cycle comparing results by $\Delta t = 1 \times 10^{-6}$ and $\Delta t = 1 \times 10^{-5}$ using a $p2$ approximation with $N = 64$ for the 1D low-temperature vortex advection problem.

order of accuracy is achieved from low to high polynomial degrees. Higher polynomial degree computations yield better slopes compared to lower polynomial degree computations because the total number of degrees of freedom (DOFs) per element is larger with a smaller truncation error which scales as $\mathcal{O}(\Delta x^{p+1})$, where Δx is the reference mesh element spacing. DOFs per element is defined as $(p + 1)^d$; where d is the dimension of the problem; as such there are 6 DOFs in $p5$ computations and 3 for $p2$.

The same test has been conducted by Wang et al. [80], and their plots are reproduced in Figures 3.6 and 3.8 for the L_1 and L_∞ error norms, respectively. In Figures 3.6 and 3.8, error plots from the study by Wang et al. approach the convergence orders, but the plots obtained in this study align more closely with expected orders. Additionally, this study requires fewer elements than that of Wang et al. to achieve the same error magnitude.

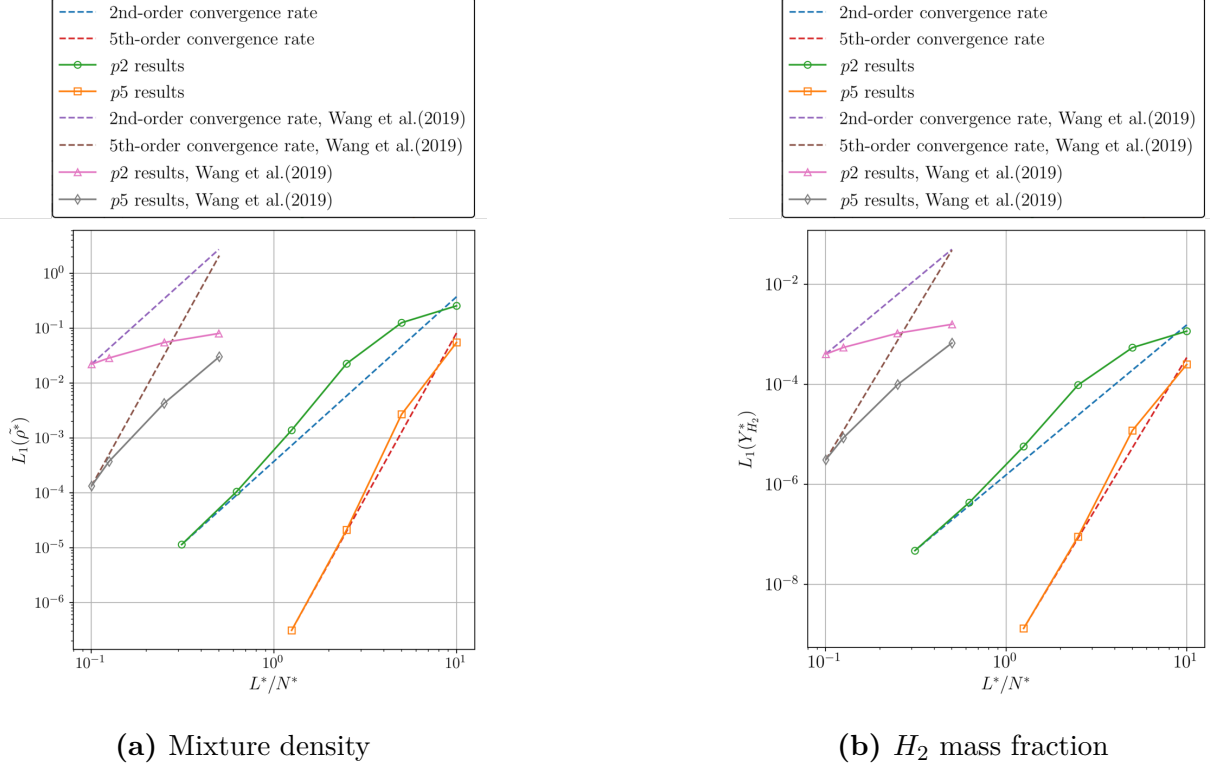


Figure 3.6: L_1 error norms of mixture density and H_2 mass fraction with a different number of elements for the 1D low-temperature vortex advection problem including results by Wang et al. [80]. Blue dot line: 2nd-order convergence rate; red dot line: 5th-order convergence rate; green solid line with circle symbol: p_2 results; orange solid line with square symbol: p_5 results; purple dot line: 2nd-order convergence rate by Wang et al.; brown dot line: 5th-order convergence rate by Wang et al.; pink solid line with triangle symbol: p_2 results by Wang et al.; gray solid line with diamond symbol: p_5 results by Wang et al.

3.2 1D High-Temperature Vortex Advection

Thermal properties exhibit non-linear relationships with temperature at high gas temperatures, while linear relationships are observed at lower temperatures. This research implements a calorically imperfect model in a DG solver to accurately simulate high-temperature flows by capturing the non-linear relationships between thermal properties and temperature observed in real gases. To demonstrate the applicability of the calorically imperfect model within the multi-species DG solver, the order of accuracy is studied under high-temperature conditions.

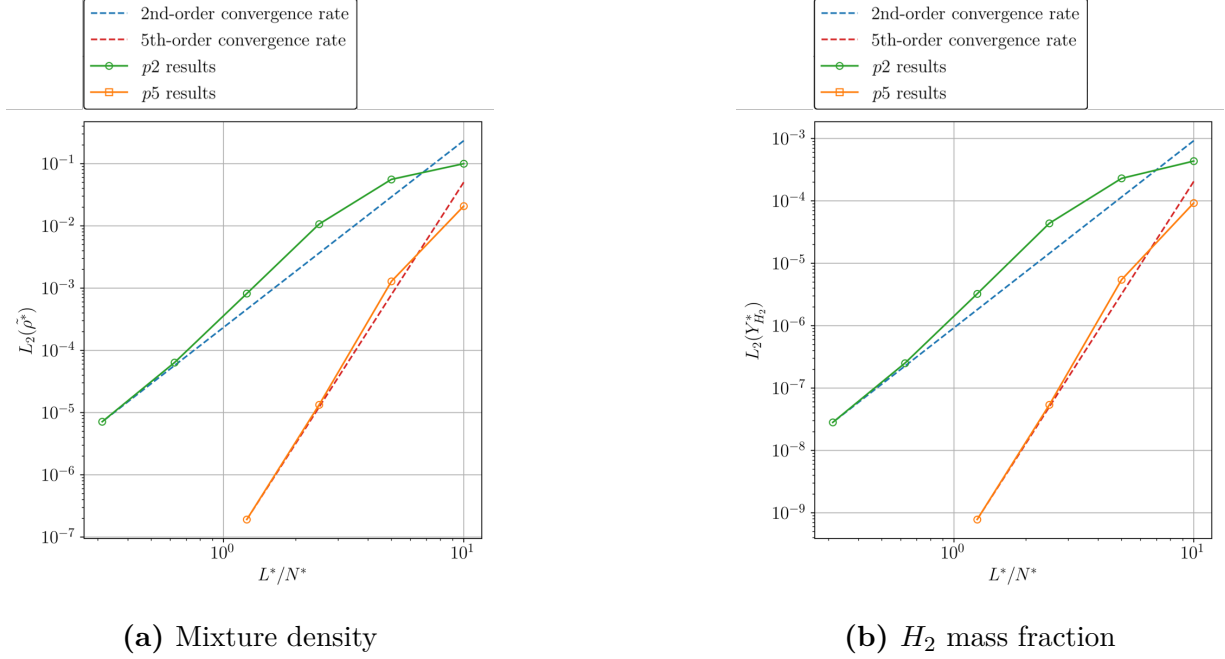


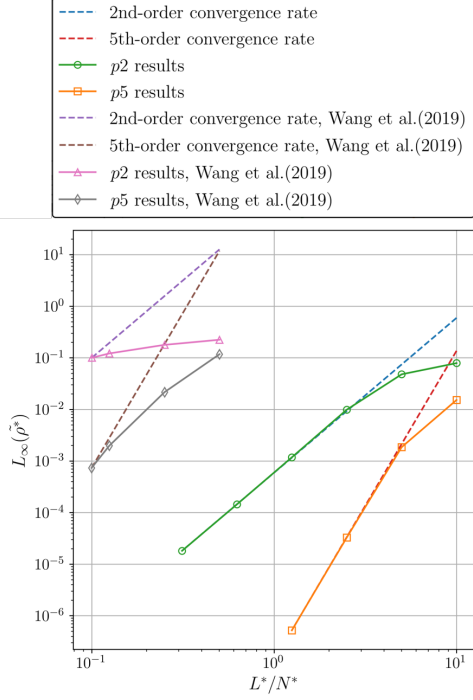
Figure 3.7: L_2 error norms of mixture density and H_2 mass fraction with a different number of elements for the 1D low-temperature vortex advection problem. Blue dot line: 2nd-order convergence rate; red dot line: 5th-order convergence rate; green solid line with circle symbol: $p2$ results; orange solid line with square symbol: $p5$ results.

3.2.1 Problem Statement

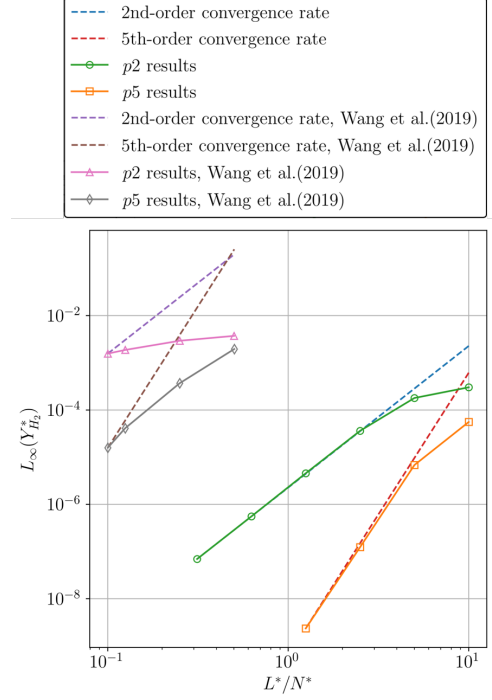
A 3-species model (H_2 , O_2 , and N_2) is used, with all test conditions aligned with the 1D low-temperature vortex advection problem, except for the temperature. In this test, the temperature is deliberately multiplied by 5 compared to that in Equation 3.1 to evaluate the applicability of the calorically imperfect model under high-temperature conditions. The initial conditions are visualized in Figure 3.9, where the temperature ranges from 1250 K to 1500 K, as shown in Figure 3.9c. The flow profiles after 0.1 cycle (0.01 s) are presented in Figure 3.10 to confirm the advection of the vortex.

3.2.2 Results

Figures 3.11, 3.12, and 3.13 present the L_1 , L_2 , and L_∞ error norms of $\tilde{\rho}^*$ and T^* as functions of L^*/N^* for $p = 2$ and $p = 5$ tests, respectively. These figures also include the theoretical



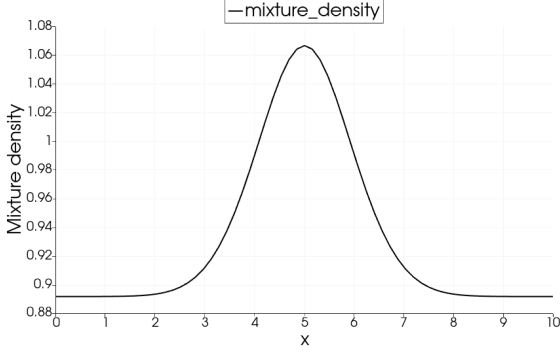
(a) Mixture density



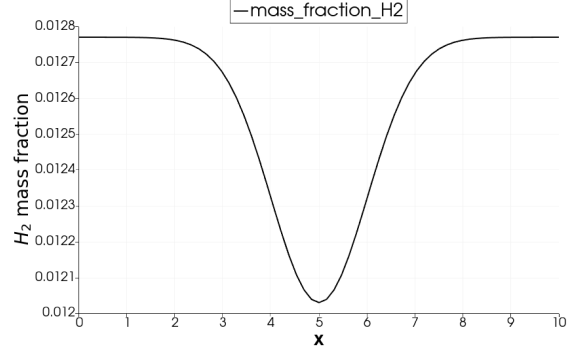
(b) H_2 mass fraction

Figure 3.8: L_∞ error norms of mixture density and H_2 mass fraction with different number of elements for the 1D low-temperature vortex advection problem including results by Wang et al. [80]. Blue dot line: 2nd-order convergence rate; red dot line: 5th-order convergence rate; green solid line with circle symbol: $p2$ results; orange solid line with square symbol: $p5$ results; purple dot line: 2nd-order convergence rate by Wang et al.; blown dot line: 2nd-order convergence rate by Wang et al.; pink solid line with triangle symbol: $p2$ results by Wang et al.; gray solid line with diamond symbol: $p5$ results by Wang et al.

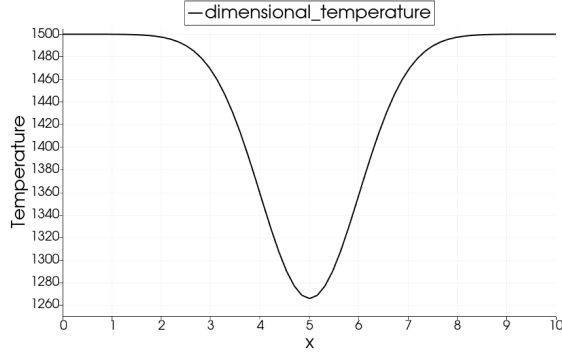
slopes corresponding to the polynomial orders used. The error norms of temperature are evaluated in this test to assess the applicability of the calorically imperfect model where temperature is a key property. According to Figures 3.11, 3.12, and 3.13, the L_1 , L_2 , and L_∞ error norms of both $\tilde{\rho}^*$ and T^* closely match the expected convergence rate as L^*/N^* decreases for both polynomial orders. This indicates that the correct convergence rate is achieved for both low- and high-order approximations within the multi-species Euler equations solver with the calorically imperfect model.



(a) Mixture density



(b) H_2 mass fraction



(c) Temperature

Figure 3.9: Profiles of mixture density, H_2 mass fraction, and temperature at the initial time for the 1D high-temperature vortex advection problem.

3.3 2D Vortex Advection

In addition to the one-dimensional studies, the accuracy of the multi-species model is evaluated in a two-dimensional domain. Error evaluations are conducted by varying polynomial orders and element counts across several numerical experiments. The test conditions are based on the 1D low-temperature vortex advection problem by Wang et al. [80], but have been expanded to two dimensions.

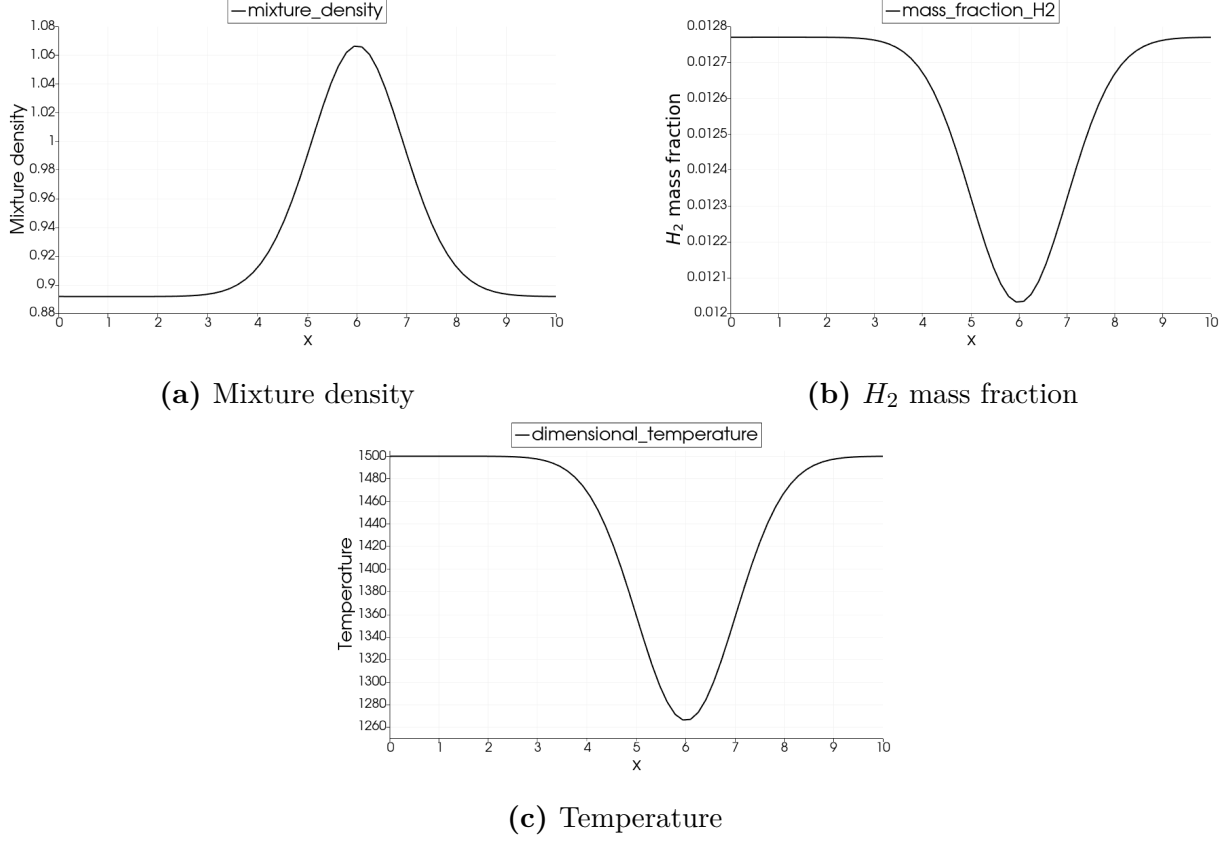


Figure 3.10: Profiles of mixture density, H_2 mass fraction, and temperature after 0.1 cycle for the 1D high-temperature vortex advection problem.

3.3.1 Problem Statement

A 2-species model (H_2 and O_2) is used, and the initial conditions are given by the following equations:

$$\begin{aligned}
 T &= T_0 - \frac{(\gamma_0 - 1) \Gamma^2}{8\gamma_0\pi} \left(\frac{1 - r^2}{2} \right), \\
 Y_{H_2} &= Y_{H_2,0} - \frac{2\pi a_1}{\gamma_0 \Gamma} \exp \left(\frac{1 - r^2}{2} \right), \\
 Y_{O_2} &= 1 - Y_{H_2},
 \end{aligned} \tag{3.2}$$

where $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$, represents the distance from the centre of the physical domain at $x_0 = y_0 = 5$ m. The uniform velocities are $u = v = 100$ m/s. The domain extends $L = 10$ m in each direction, with periodic boundary conditions applied to all boundaries. To minimize temporal errors, a very small time step, $\Delta t = 10^{-6}$ s, is used for all grid sizes,

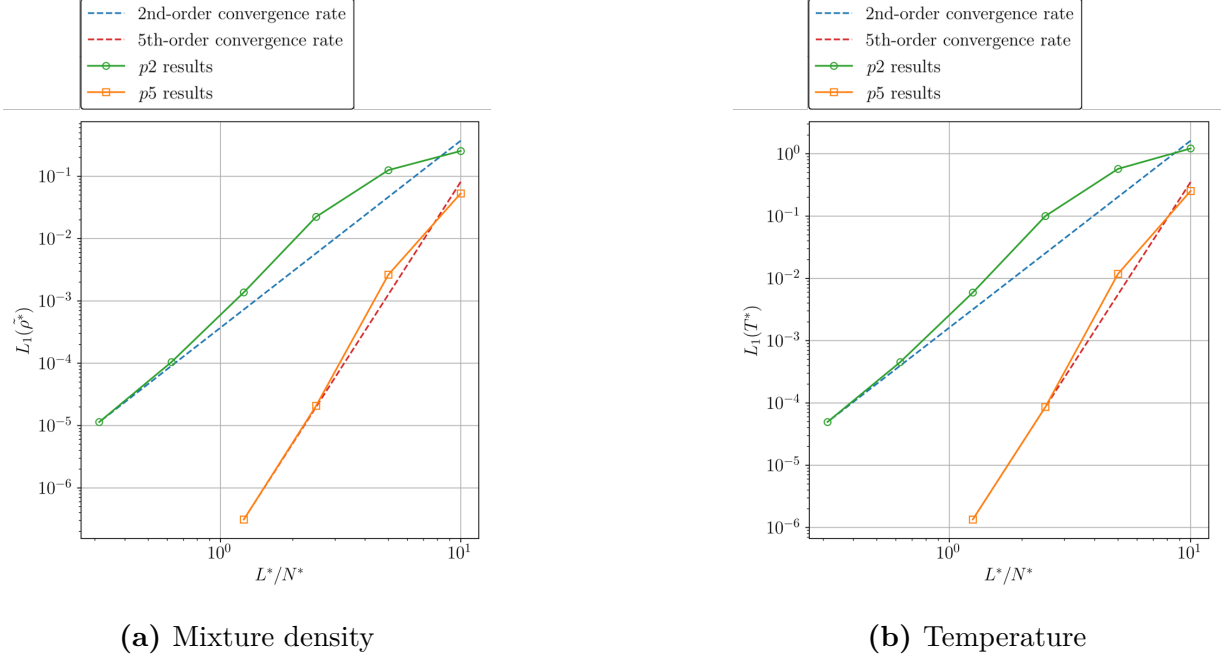


Figure 3.11: L_1 error norms of mixture density and temperature with different number of elements for the 1D high-temperature vortex advection problem. Blue dot line: 2nd-order convergence rate; red dot line: 5th-order convergence rate; green solid line with circle symbol: $p2$ results; orange solid line with square symbol: $p5$ results.

consistent with the 1D vortex advection problems. However, this small time step significantly increases the computational cost for two-dimensional simulations. To manage this, the error norms are evaluated after 0.1 cycle (after 0.01 s) by comparing the exact solutions obtained by shifting the initial conditions by 1 m in both the x and y directions. The initial conditions are illustrated in Figure 3.14, which shows the contours of $\tilde{\rho}^*$, $Y_{H_2}^*$, and T . The contours after 0.1 cycle (0.01 s) are presented in Figure 3.15 to illustrate the advection of the vortex.

3.3.2 Results

The L_1 , L_2 , and L_∞ error norms of $\tilde{\rho}^*$ and $Y_{H_2}^*$ are computed after 0.1 cycle ($t = 0.01$ s) by comparing results against the exact solutions. To verify the effectiveness of low- and high-order computations, tests are conducted for $p = 1, 2$, and 3 approximations.

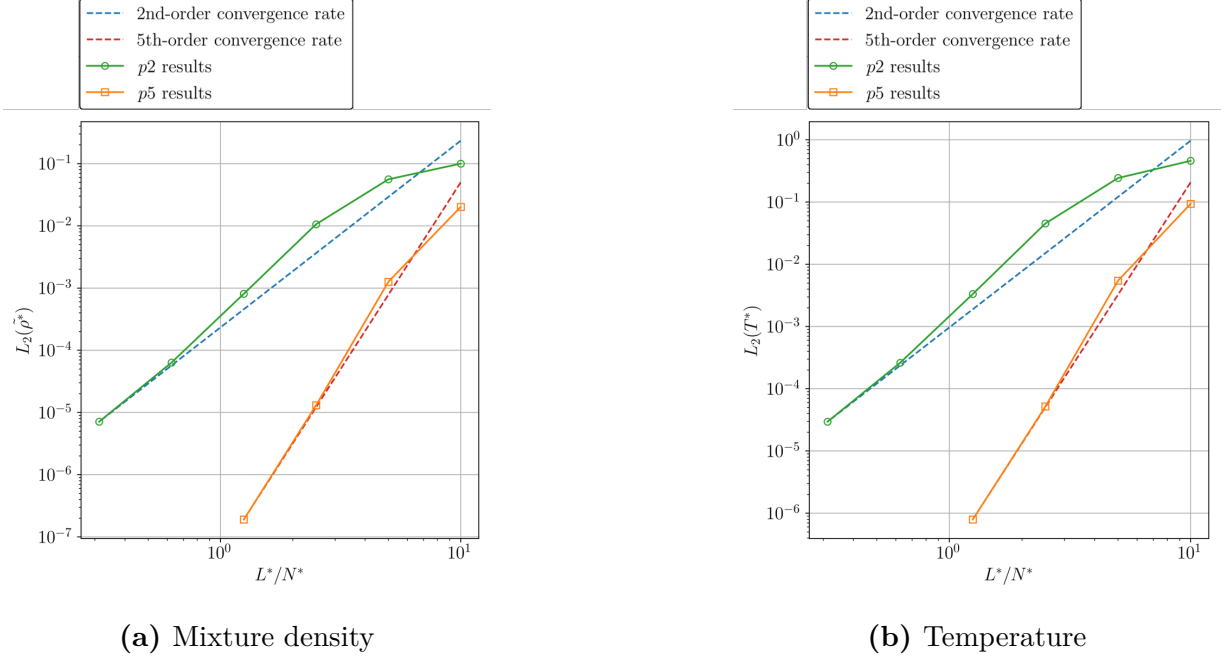
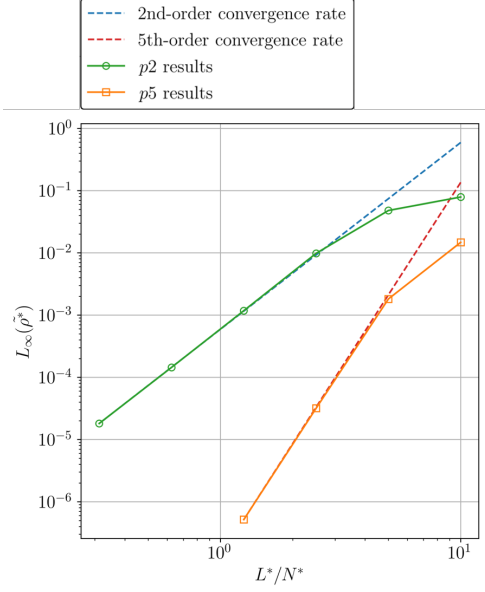


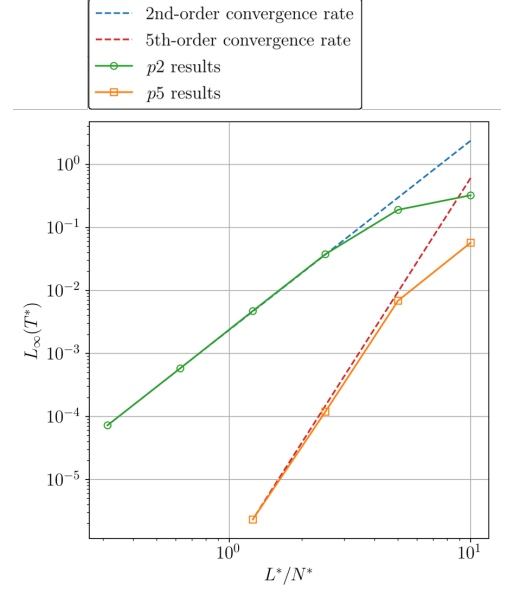
Figure 3.12: L_2 error norms of mixture density and temperature with different number of elements for the 1D high-temperature vortex advection problem. Blue dot line: 2nd-order convergence rate; red dot line: 5th-order convergence rate; green solid line with circle symbol: $p2$ results; orange solid line with square symbol: $p5$ results.

Figure 3.16 compares the cross-section of mixture density, mass fraction of H_2 , and temperature along the $y^* = x^*$ axis using the same number of DOFs per dimension ($N = 64$) but with two different polynomial degrees, $p = 1$ and $p = 3$. The results are plotted against the distance from $(x^*, y^*) = (0, 0)$, calculated as $\sqrt{(x^*)^2 + (y^*)^2}$. The figures show that the $p1$ approximation exhibits some deviation from the exact solution, whereas the $p3$ approximation closely matches the exact solutions. This indicates that using a higher polynomial degree produces more accurate results for equivalent DOFs in two-dimensional computations, highlighting the advantages of higher-order methods.

Figures 3.17, 3.18, and 3.19 show the L_1 , L_2 , and L_∞ error norms of $\tilde{\rho}^*$ and $Y_{H_2}^*$ with respect to L^*/N^* for $p = 1, 2$, and 3 . These figures also include the theoretical convergence rates for each polynomial order. According to these figures, the L_1 and L_2 error norms for both $\tilde{\rho}^*$ and $Y_{H_2}^*$ closely match the theoretical slopes across all polynomial orders. In contrast, the L_∞ error norm tends towards the expected slopes. However, in this study, the



(a) Mixture density



(b) Temperature

Figure 3.13: L_∞ error norms of mixture density and temperature with different numbers of elements for the 1D high-temperature vortex advection problem. Blue dot line: 2nd-order convergence rate; red dot line: 5th-order convergence rate; green solid line with circle symbol: $p2$ results; orange solid line with square symbol: $p5$ results.

L_∞ error norm illustrated in Figure 3.19 show better convergence rates than those reported in the previous study by Wang et al. [80] (see in Figure 3.8). Overall, the error norm plots demonstrate the predicted slopes and show a better convergence rate compared to the previous study. This confirms that the implemented multi-species model shows the right order of accuracy in two-dimensional computations.

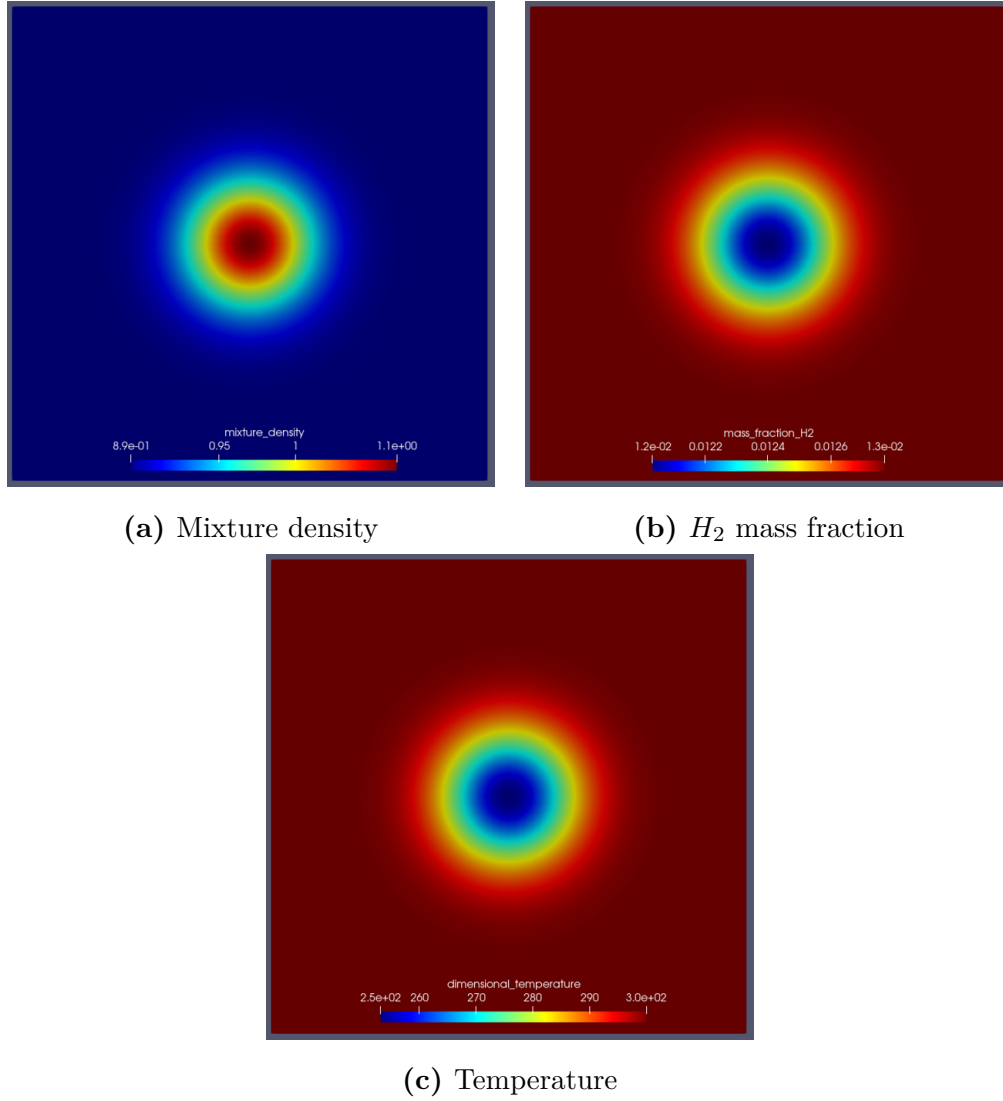
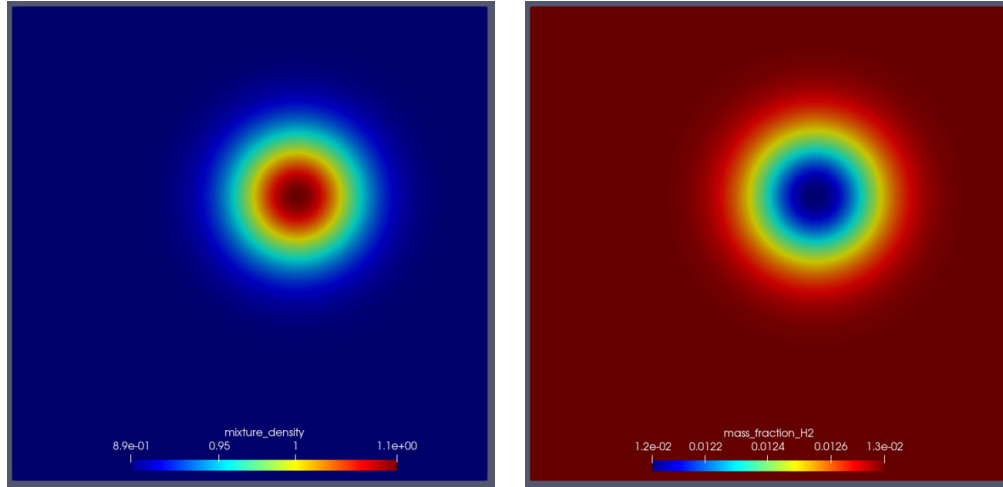
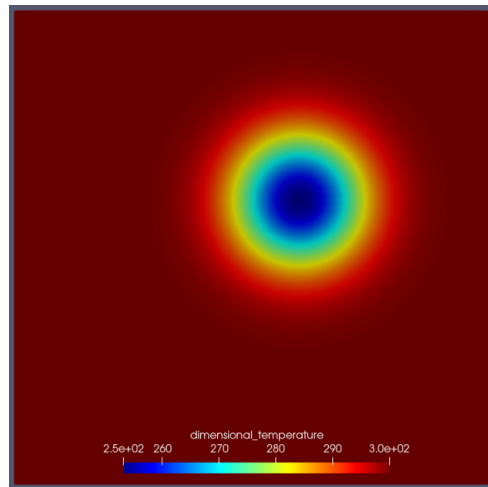


Figure 3.14: Contours of mixture density, H_2 mass fraction, and temperature at the initial time for the 2D vortex advection problem.



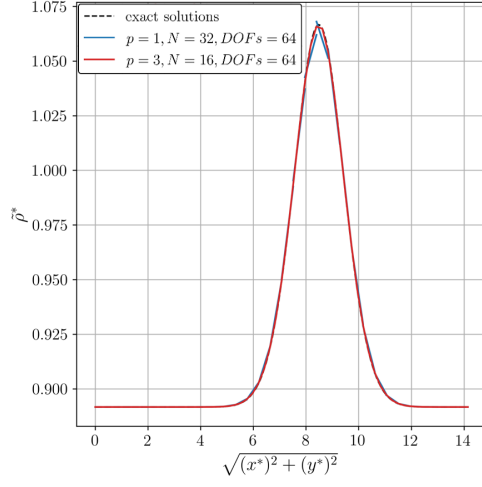
(a) Contours of $\tilde{\rho}^*$

(b) H_2 mass fraction

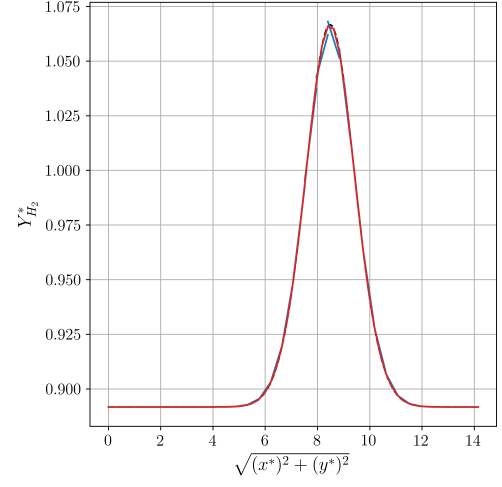


(c) Temperature

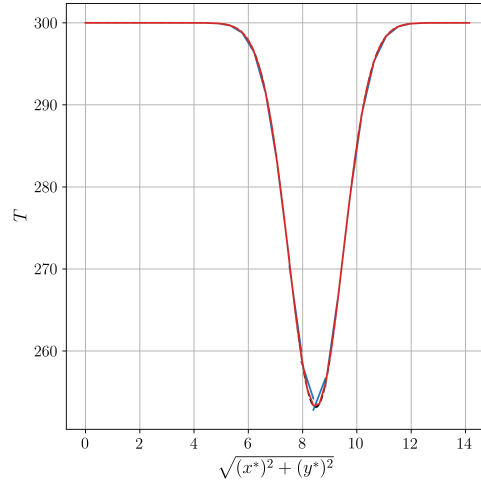
Figure 3.15: Contours of mixture density, H_2 mass fraction, and temperature after 0.1 cycle for the 2D vortex advection problem.



(a) Mixture density

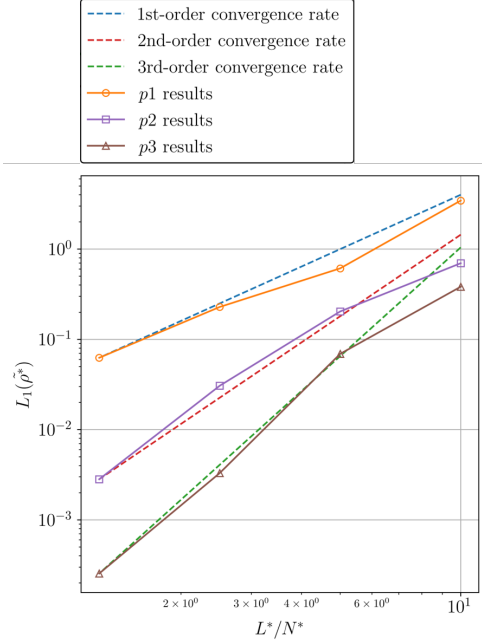


(b) H_2 mass fraction

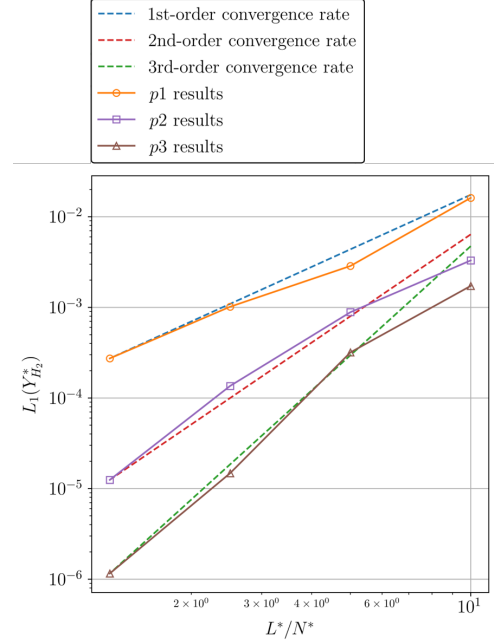


(c) Temperature

Figure 3.16: Profiles of mixture density, H_2 mass fraction, and temperature at $y^* = x^*$ axis after 0.1 of a cycle using $p1$ and $p3$ approximations using different grid sizes such that DOFs are equivalent for the 2D low-temperature vortex advection problem.

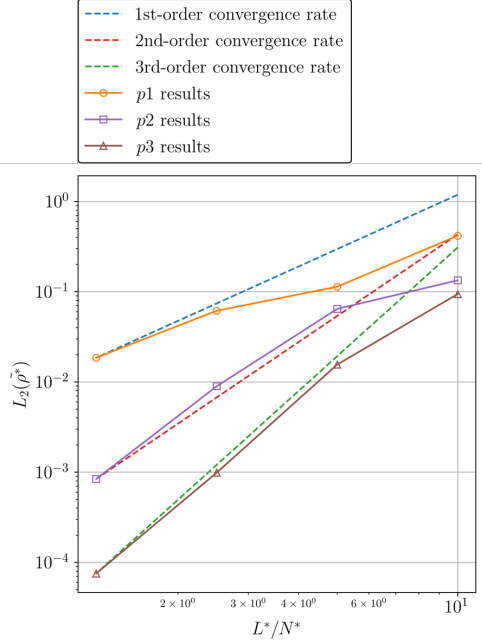


(a) Mixture density

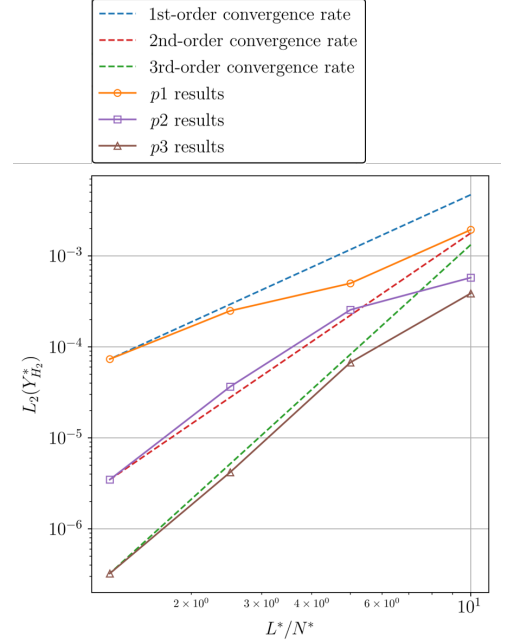


(b) H_2 mass fraction

Figure 3.17: L_1 error norm of mixture density and H_2 mass fraction with a different number of elements for the 2D vortex advection problem. Blue dot line: 1st-order convergence rate; red dot line: 2nd-order convergence rate; green dot line: 3rd order convergence rate; orange solid line with circle symbol: $p1$ results; purple solid line with square symbol: $p2$ results; brown solid line with triangle symbol: $p3$ results.

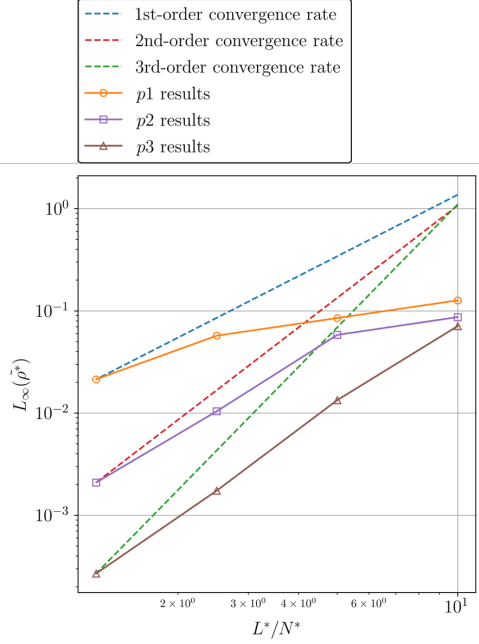


(a) Mixture density

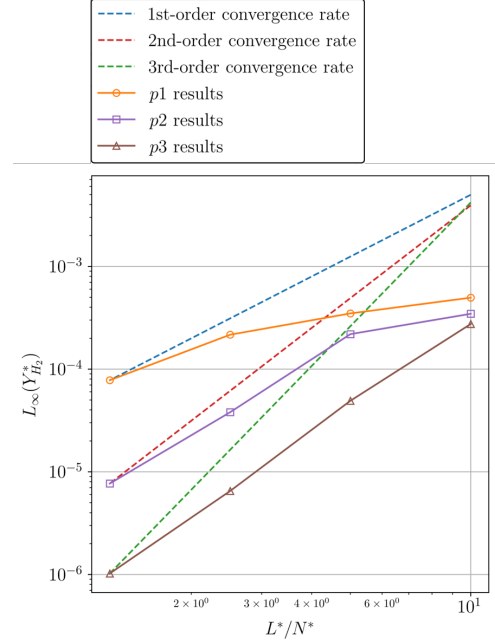


(b) H_2 mass fraction

Figure 3.18: L_2 error norm of mixture density and H_2 mass fraction with a different number of elements for the 2D vortex advection problem. Blue dot line: 1st-order convergence rate; red dot line: 2nd-order convergence rate; green dot line: 3rd order convergence rate; orange solid line with circle symbol: $p1$ results; purple solid line with square symbol: $p2$ results; brown solid line with triangle symbol: $p3$ results.



(a) Mixture density



(b) H_2 mass fraction

Figure 3.19: L_∞ error norm of mixture density and H_2 mass fraction with different number of elements for the 2D vortex advection problem. Blue dot line: 1st-order convergence rate; red dot line: 2nd-order convergence rate; green dot line: 3rd order convergence rate; orange solid line with circle symbol: $p1$ results; purple solid line with square symbol: $p2$ results; brown solid line with triangle symbol: $p3$ results.

Chapter 4

Validation

Validation in CFD involves comparing simulation results against experimental data to assess and reduce modeling uncertainties [81]. However, due to the scarcity of experimental benchmark tests for non-reactive multi-species models, finding suitable experimental data for validation is challenging. Therefore, an alternative approach is to compare the simulation results with those from previously conducted numerical experiments. This approach also ensures the validation of numerical models within a CFD solver, even in the absence of extensive experimental data.

4.1 2D Isentropic Euler Vortex Advection

The isentropic vortex test case, first introduced by Shu et al. [82], is frequently used as a validation case for numerical solvers. This test was extended to a multi-species scenario by Trojak et al. [48], and simulated under low-temperature conditions using a calorically perfect model. In this research, it is further extended to a calorically imperfect test. Since low-temperature conditions are used in both tests, the results should be similar, allowing for a comparative investigation. The 2D isentropic Euler vortex advection problem is solved using a calorically imperfect multi-species Euler equations solver. By comparing results

in this study with those from Trojak et al. [48], we validate implementations of models developed in this study.

4.1.1 Problem Statement

The two-species model (N_2 and O_2) is used, and the initial conditions are given by the following equations:

$$\begin{aligned}
\rho_{N_2}^* &= \left(1 - \frac{(\gamma_{N_2}^* - 1)\beta^2 M_{ref}^2 \exp(2f)}{8\pi^2} \right)^{1/(\gamma_{N_2}^* - 1)}, \\
\rho_{O_2}^* &= \left(1 - \frac{(\gamma_{O_2}^* - 1)\beta^2 M_{ref}^2 \exp(2f)}{8\pi^2} \right)^{1/(\gamma_{O_2}^* - 1)}, \\
u^* &= U_0 + \frac{\beta y^*}{2\pi R^*} \exp(f), \\
v^* &= V_0 - \frac{\beta x^*}{2\pi R^*} \exp(f), \\
\tilde{P}^* &= \frac{1}{\tilde{\gamma}^* M_{ref}^2} \tilde{\rho}^* \tilde{\gamma}^*, \\
f &= \frac{1 - (x^* - x_0)^2 - (y^* - y_0)^2}{2R^{*2}}.
\end{aligned} \tag{4.1}$$

The domain is defined as $\Omega^* = [x_0 - L^*, x_0 + L^*] \times [y_0 - L^*, y_0 + L^*]$. Given that the domain is centered at the origin $x_0 = y_0 = 0$ and has an extent of $L^* = 10$, the physical domain becomes $\Omega^* = [-10, 10] \times [-10, 10]$. This ensures a sufficiently large area to observe the vortex and advection phenomena. The nondimensional parameter $\beta = 13.5$ quantifies the strength of the vortex, with a radius of $R^* = 1.5$. The advection velocities $U_0 = 0$ and $V_0 = 1$ imply that there is no movement in the x -direction but in the y -direction. The reference Mach number is $M_{ref} = 0.4$. This setup is appropriate for numerical simulations of vortices with advection effects, ensuring a comprehensive analysis of the flow characteristics within the defined computational domain.

4.1.2 Results

The solution is advanced to 1.0 cycle ($t^* = 20$), after which the flow velocities are reversed. The test is conducted with different numbers of elements, specifically $N = 16, 32$, and 64 in each direction, employing a $p = 4$ approximation for higher-order accuracy, aligned with the approach used by Trojak et al. [48]. Additionally, a test using a lower-order $p = 2$ approximation is conducted for comparison purposes. The evolution of the vortex roll-up with time advancement is illustrated in Figure 4.1, showing the density contours of N_2 . The contours after 1.0 cycle obtained in this study (see Figure 4.1f) closely resemble those obtained by Trojak et al. [48], validating the implemented models in this work.

In addition, Figure 4.2 presents the contours of $\rho_{N_2}^*$ after the flow reversal, comparing the results with varying mesh sizes and polynomial degrees. Figures 4.2a and 4.2b show the results with a small number of DOFs, Figures 4.2c and 4.2d depict the results with a medium number of DOFs, and Figures 4.2e and 4.2f illustrate the results with a large number of DOFs. These figures show that the computations with a medium and large number of DOFs capture the vortex roll-up well, in contrast to the results with a small number of DOFs. At all resolutions, the roll-up phenomenon is resolved better in $p4$ computations than in $p2$ computations when the number of DOFs is similar. The differences are particularly noticeable between Figures 4.2c and 4.2d at medium resolutions.

Figure 4.3 shows $\rho_{N_2}^*$ after the flow reversal for all mesh sizes, including the results in this study and those from the previous study by Trojak et al. [48], both using a $p4$ approximation. In all resolutions, $\rho_{N_2}^*$ exhibits oscillations, indicating the presence of the roll-up vortex, as seen in Figure 4.2. The zoomed-in figure on the left highlights that the results at medium resolutions ($N = 40$ and $N = 32$) and higher resolutions ($N = 80$ and $N = 64$) show similar plots. Moreover, at these resolutions, results closely match those of Trojak et al. [48]. However, at low resolutions ($N = 40$ and $N = 32$), plots differ significantly from those at medium and high resolutions. In this study, the plots show different results from those of Trojak et al. [48] near the centre of the domain (between $y^* = -2.5$ and $y^* = 2.5$), although the trend remains similar, as seen in the zoomed-in figure on the right. Compared with the

previous study, results in this work reasonably match those of Trojak et al. [48], with better agreement at higher resolutions.

Figure 4.4 compares results with two polynomial degrees, $p = 2$ and $p = 4$, with a similar number of DOFs. According to the zoomed-in figure on the left, the results from a medium number of DOFs ($p = 2, N = 64$ and $p = 4, N = 32$) and a large number of DOFs ($p = 2, N = 128$ and $p = 4, N = 64$) show similar plots, while the results from a small number of DOFs ($p = 2, N = 32$ and $p = 4, N = 16$) differ slightly. In the zoomed-in figure on the right, the results using a large number of DOFs show similar plots between $p2$ and $p4$ approximations and the vortex roll-up are well-captured even at the centre of the domain, where the phenomenon is otherwise less clearly observed.

Overall, the solver demonstrates consistent results across low- to high-order computations, closely matching the data from the previous study. These findings validate the correct implementation of the models.

4.2 2D Fuel Droplet Advection

This test, introduced by Ma et al. [83], is designed to evaluate the performance of the numerical schemes for multi-species flows. It involves a two-dimensional scenario in which a fuel species droplet is advected through an inert gas under low-temperature conditions using a calorically perfect model. In this research, it is extended to a calorically imperfect case. Given that low-temperature conditions are employed in both scenarios, the results should be comparable, facilitating a comparative analysis. The test is solved using the calorically imperfect multi-species Euler equations solver. We validate the implementations by comparing results in this work with those from Ma et al. [83].

4.2.1 Problem Statement

The 2-species model (iso-octane as fuel and N_2 as ambient gas) is used, with initial mass fractions given by:

$$\begin{aligned} Y_{fuel} &= 1.0 - 0.5(1 + \tanh(\alpha(r - r_0))), \\ Y_{N_2} &= 1 - Y_{fuel}, \end{aligned} \tag{4.2}$$

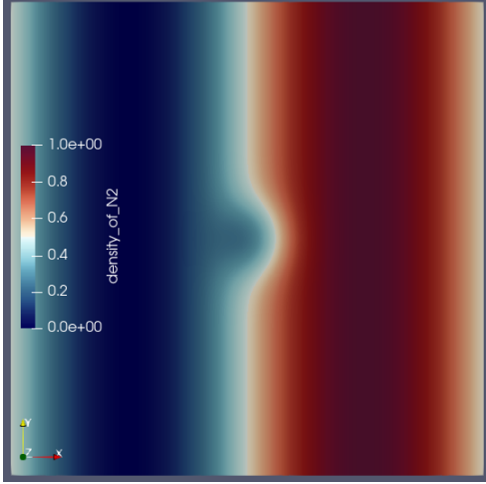
where the computational domain extends $L = 1$ mm in each direction, and periodic boundary conditions are applied to all boundaries. This mass fraction is obtained from the numerical results by Ma et al. [83], corresponding to the Spray A operating point [84]. The center of the domain is located at $(x, y) = (x_0, y_0) = (0.5, 0.5)$ mm. The variable $r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$ mm represents the distance from the center of the fuel droplet, and $r_0 = 1/\pi$ mm defines its radius. The parameter α controls the intensity of the transition between the fuel droplet and the ambient gas. Specifically, $\alpha = 23$ for $r < r_0$ (within the fuel droplet) and $\alpha = 60$ for $r \geq r_0$ (outside of the fuel droplet). In the entire domain, the uniform velocities $u = 50, v = 0$ m/s, temperature $T = 573$ K, and mixture pressure $P = 600$ kPa are used, where iso-octane is used as the fuel droplet species. The mixture density is calculated from the mixture pressure and temperature using the equation of state.

4.2.2 Results

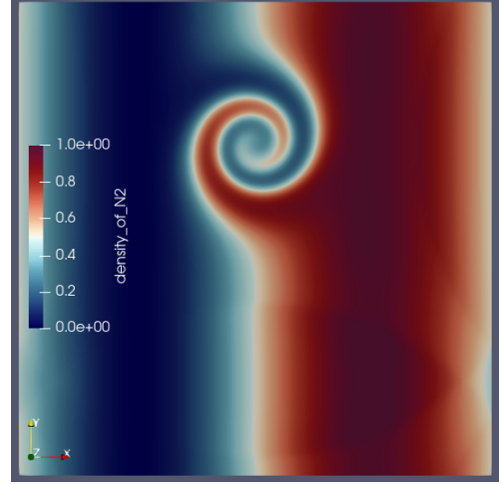
The fuel droplet moves positively in the x -direction, passes through the periodic boundary, and returns to its original position after 1.0 cycle at $t = 0.02$ ms. The test case described here uses a mesh with $N = 32$ elements in each direction and an approximation order of $p = 2$. Figure 4.5 illustrates the evolution of the flow field of Y_{fuel}^* over time. For comparison purposes, additional tests are also performed using high-order computations ($N = 32$ and $p = 5$) with the same initial temperature, as well as a scenario with a temperature five times higher while maintaining the approximation at $p = 2$.

Figure 4.6 presents the plots of Y_{fuel}^* at $y^* = 0.5$ after 1.0 cycle, including various test scenarios. These plots compare the initial conditions, results from the previous study by Ma

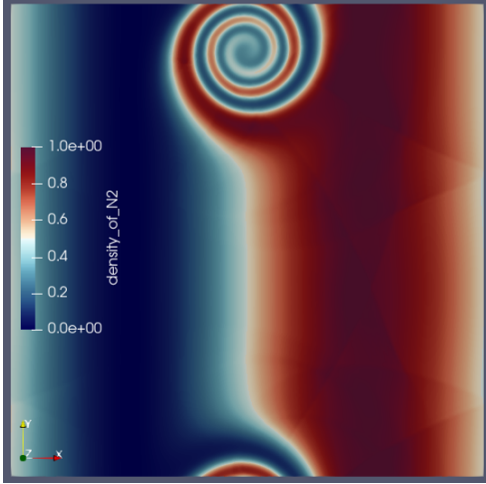
et al. [83], results using the $p = 2$ approximation at a lower temperature, results using a higher polynomial degree with the same temperature, and results at a higher temperature using the same polynomial degree. This figure demonstrates that the advected vortex returns precisely to the original location after one cycle. According to the zoomed-in view, the results using $p2$ approximation exhibit some differences compared to other cases, while the $p5$ results are very similar to both the initial conditions and the data from Ma et al. [83]. Across varying temperatures and polynomial degrees, the results in this study closely match the mass fraction data from the previous study throughout the domain. This comparison validates the implementations of the models.



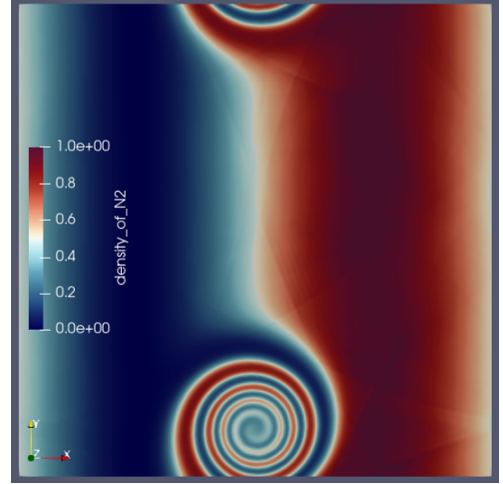
(a) At initial time



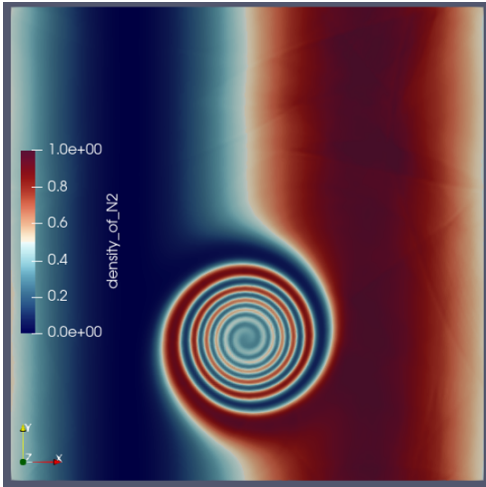
(b) After 0.2 Cycle



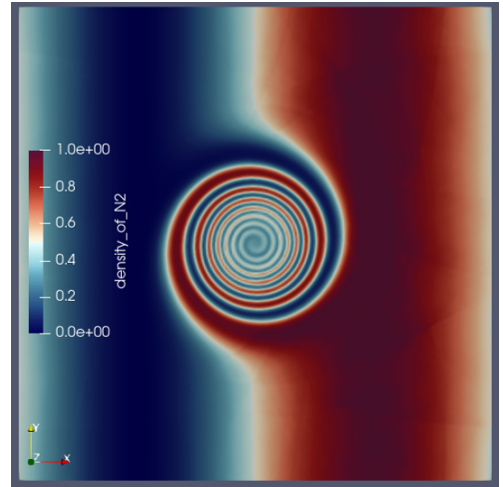
(c) After 0.4 cycle



(d) After 0.6 cycle

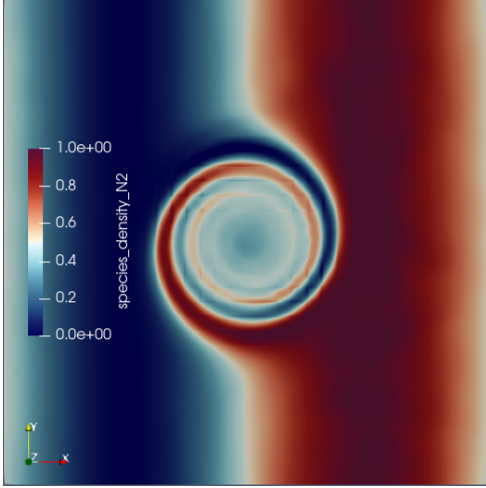


(e) After 0.8 cycle

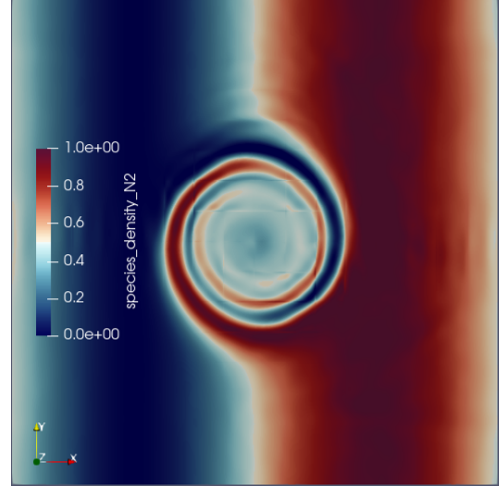


(f) After 1.0 cycle

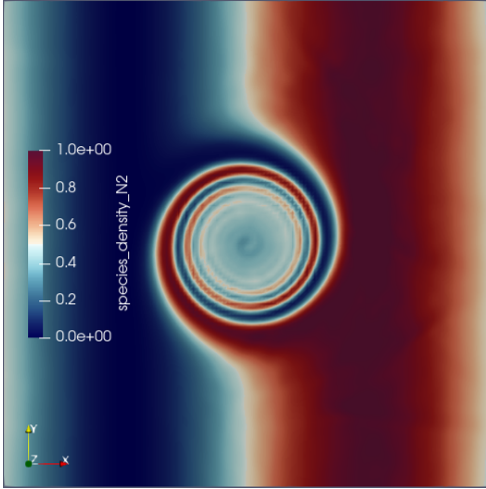
Figure 4.1: Contours of N_2 density with the time progress using a p_4 approximation for the 2D isentropic Euler vortex advection problem.



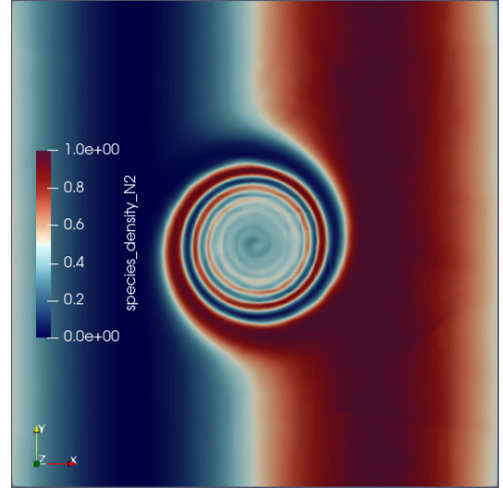
(a) $p2, N = 32, DOFs = 96$



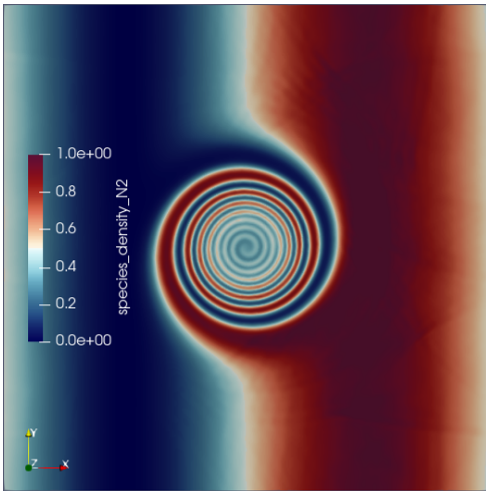
(b) $p4, N = 16, DOFs = 80$



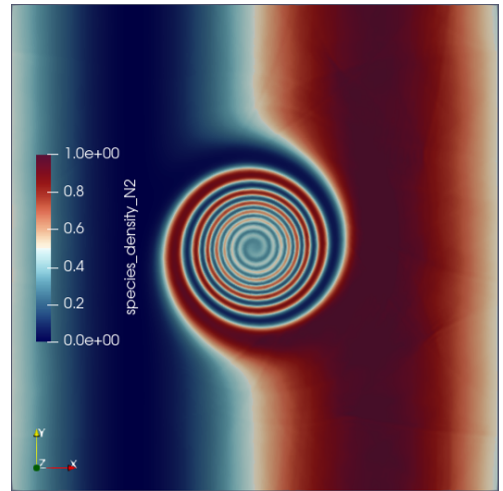
(c) $p2, N = 64, DOFs = 192$



(d) $p4, N = 32, DOFs = 160$



(e) $p2, N = 128, DOFs = 384$



(f) $p4, N = 64, DOFs = 320$

Figure 4.2: Contours of N_2 density after flow reversal with the grid refinement for the 2D isentropic Euler vortex advection problem.

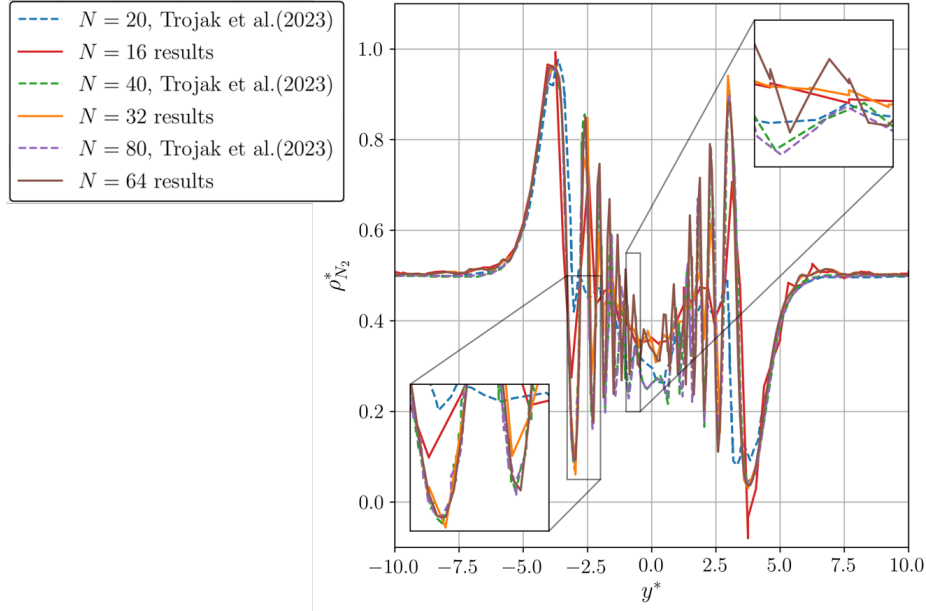


Figure 4.3: Comparison of the cross-section of N_2 density at $x^* = 0$ after flow reversal using a $p4$ approximation. Blue dot line: $N = 20$ by Trojak et al.; red solid line: $N = 16$; green dot line: $N = 40$ by Trojak et al.; orange solid line: $N = 32$; purple dot line: $N = 80$ by Trojak et al.; brown solid line: $N = 64$.

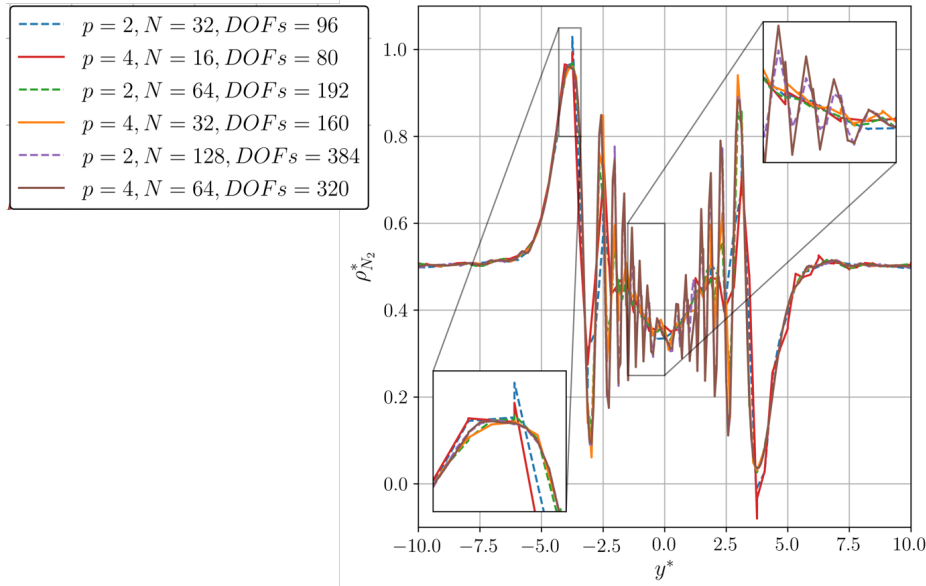
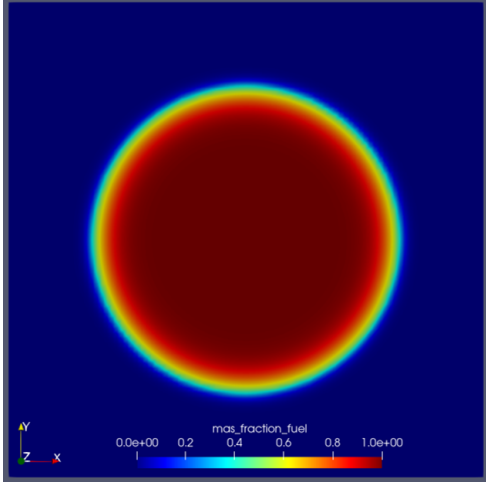
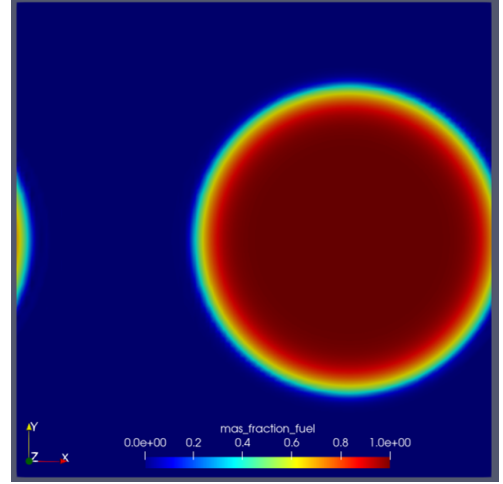


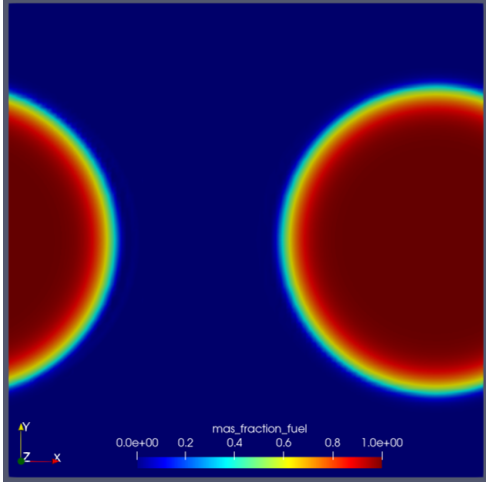
Figure 4.4: Comparison of the cross-section of N_2 density at $x^* = 0$ after flow reversal. Blue dot line: $p = 2, N = 32, DOFs = 96$; red solid line: $p = 4, N = 16, DOFs = 80$; green dot line: $p = 2, N = 64, DOFs = 192$; orange solid line: $p = 4, N = 32, DOFs = 160$; purple dot line: $p = 2, N = 128, DOFs = 384$; brown solid line: $p = 4, N = 64, DOFs = 320$.



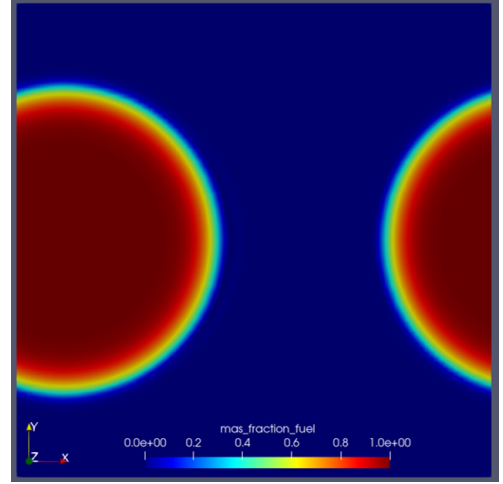
(a) At initial time



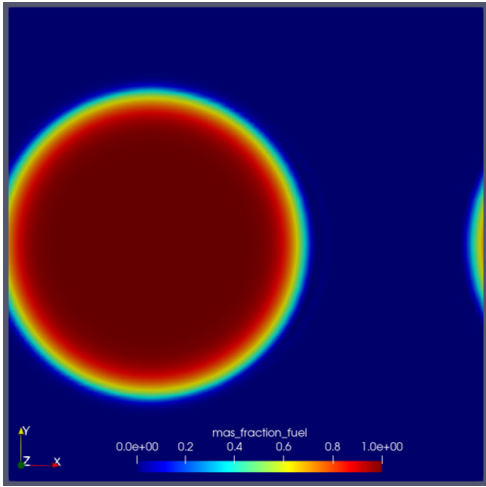
(b) After 0.2 Cycle



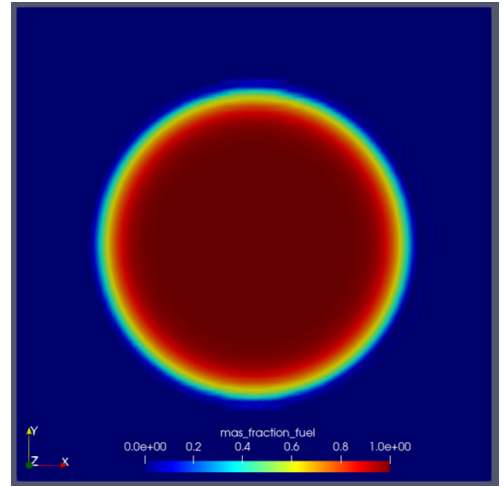
(c) After 0.4 cycle



(d) After 0.6 cycle



(e) After 0.8 cycle



(f) After 1.0 cycle

Figure 4.5: Contours of fuel mass fraction with time progress for the 2D fuel drop advection problem ($p = 3$ and $N = 32$).

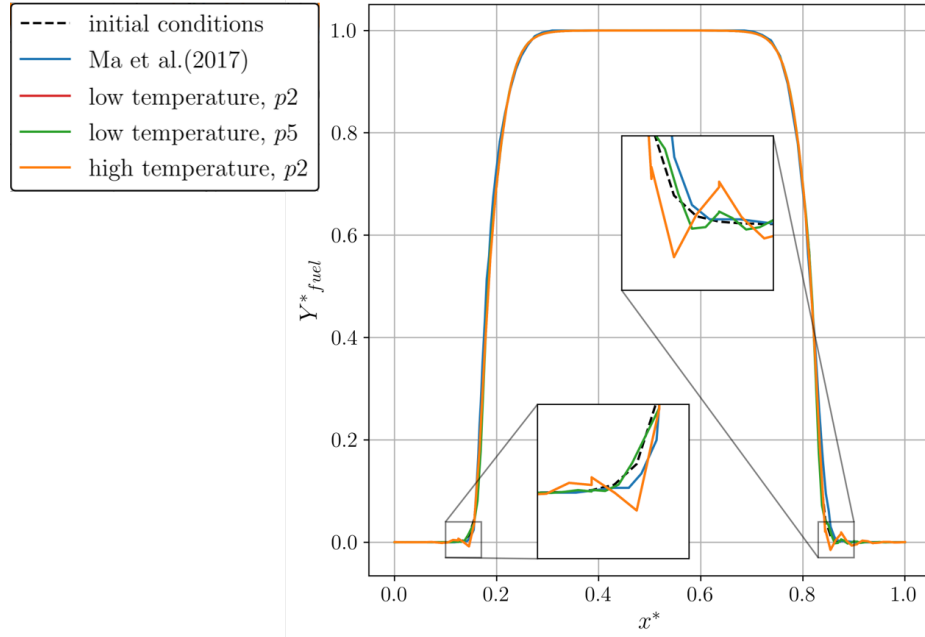


Figure 4.6: Comparison of the cross-section of fuel mass fraction at $y^* = 0.5$ after flow reversal for the 2D fuel droplet advection problem. Black dot line: the initial conditions; blue solid line: results by Ma et al.; red solid line: p_2 results with low-temperature; p_5 results with low-temperature; orange solid line: p_2 results with high-temperature.

Chapter 5

Conclusion

In this study, we have successfully implemented the multi-species model and the calorically imperfect model within a DG solver, thereby enhancing its capability to accurately simulate complex flow phenomena. The integration of these models allows the solver to handle both low-temperature and high-temperature conditions effectively. The implemented models were tested through a series of numerical experiments conducted in both one-dimensional and two-dimensional domains. These tests included scenarios with high-temperature conditions to evaluate the performance of the calorically imperfect model. The results from these experiments demonstrated that the DG solver, enhanced by the multi-species and calorically imperfect models, achieves the expected convergence rates for both low-order and high-order computations. Furthermore, the computed order of accuracy in this work shows improved alignment with theoretical expectations compared to previous studies. This indicates that the models and implementation not only meet but exceed the performance benchmarks established by earlier research.

In addition to verifying the accuracy of the solver, extensive validation tests were conducted to further assess the applicability of the multi-species model. These validation tests, based on established numerical experiments from other researchers, were extended from the calorically perfect framework to incorporate the calorically imperfect model. The results obtained from these validation tests closely align with previously reported data, thereby

confirming the successful implementation of both the calorically imperfect and multi-species models within the Euler equations solver using the DG method.

The primary contribution of this work is the successful integration of the multi-species model and the calorically imperfect model into the PHiLiP framework. This enhancement enables the solver to accurately simulate both non-reacting mixing and high-temperature scenarios, overcoming the limitations of the previous mono-species and calorically perfect models. By addressing these limitations, our work lays a solid foundation for future developments in PHiLiP, particularly for implementing reacting models and multi-species entropy-stable schemes.

5.1 Future Work

Several avenues for future work arise from the developments presented in this thesis. Further numerical testing is essential to thoroughly evaluate the capabilities of the implemented models. While the current tests have been performed within two-dimensional domains and with two-species and three-species models, extending these tests to three-dimensional domains and incorporating a larger number of species would provide a more comprehensive assessment of the models under diverse conditions. Moreover, incorporating limiters into the multi-species solver would allow for the exploration of multi-species shock-involved cases, where complex and nonlinear phenomena are prevalent. Future work could include investigating multi-species shock tube problems [83, 85] and air-helium shock bubble interactions [31, 85], both of which present challenging dynamics and complex flow patterns. These cases would serve as valuable benchmarks for further validation.

Extending the governing equations from multi-species Euler equations to multi-species Navier-Stokes equations would allow for testing viscous flows. Although the Navier-Stokes equations solver is already available in PHiLiP, it would require further implementation of viscous fluxes under multi-species assumptions. Moreover, this thesis has focused on non-reacting flows but it would be beneficial for PHiLiP to implement the reacting model to

enhance the capability to simulate combustng flows. For a reaction model, a very small time step or CFL number would be necessary to capture reacting phenomena, and a mass diffusion model would be required to capture the distribution of species particles. Lastly, it would be valuable to consider multi-species version of the entropy stable scheme. This scheme has been studied and implemented in PHiLiP for mono-species computations. Since there are limited number of studies for the multi-species entropy stable scheme, it would be beneficial to extend this scheme from mono-species to multi-species scenarios.

5.1.1 3D Test Case (Multi-Species Inviscid Taylor-Green Vortex)

The Taylor-Green vortex, first studied by Taylor et al. [86], has become a popular benchmark case over the years due to its simplicity in geometry and the complex flow phenomena it represents [87]. While it is extensively employed to validate computational codes for mono-species flow, relatively few studies have explored its applicability to multi-species gas dynamics. For example, Abdelsamie et al. [88] applied a Direct Numerical Simulation (DNS) solver to study interactions of a Taylor-Green vortex with both non-reacting and reacting hydrogen-oxygen flames. This demonstrates the potential of the Taylor-Green vortex as a benchmark for multi-species flows, particularly in exploring complex interactions such as chemical reactions in vortical structures. Similarly, Trojak et al. [48] explored the multi-species Taylor-Green vortex flow using a DG solver. However, their analysis was limited to a two-species system with identical specific heat constants. This simplification allowed direct comparisons with mono-species solutions, but it did not explore the broader complexity of multi-species dynamics.

In this study, despite current limitations in the number of governing equations available in the existing multi-species solver, the inviscid Taylor-Green vortex flow is tested considering two different species to assess the three-dimensional computational capabilities of the solver. Additionally, this flow is extended to include a calorically imperfect assumption, underscoring its potential for advancing research in high-temperature and chemically reacting flows within three-dimensional domains.

The two-species model comprising N_2 and O_2 is employed, with the initial state defined by the following equations:

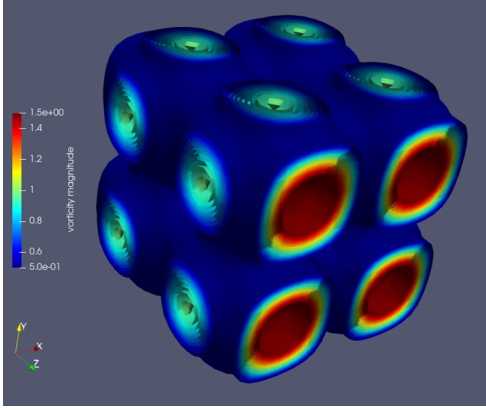
$$\begin{aligned}
\tilde{\rho}^* &= 1, \\
u^* &= \sin(x) \cos(y) \cos(z), \\
v^* &= \cos(x) \sin(y) \cos(z), \\
w^* &= 0, \\
\tilde{P}^* &= \frac{1}{\gamma_{ref} M_{ref}^2} + \frac{1}{16} (\cos(2x) + \cos(2y)) (\cos(2z) + 2),
\end{aligned} \tag{5.1}$$

where $\gamma_{ref} = 1.4$ and $M_{ref} = 0.1$. The initial mass fractions are given as:

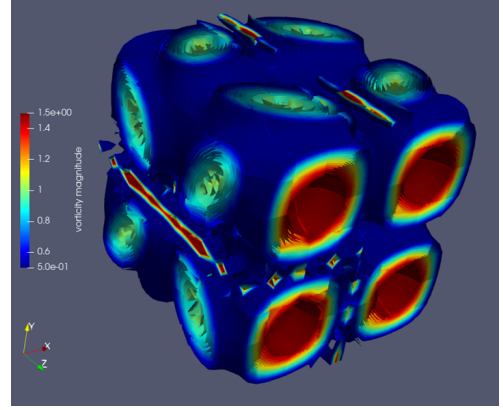
$$\begin{aligned}
Y_{N_2}^* &= \begin{cases} 0.9 & \text{if } x > \pi \text{ and } y > \pi \text{ or } x < \pi \text{ and } y < \pi, \\ 0.1 & \text{otherwise,} \end{cases} \\
Y_{O_2}^* &= 1 - Y_{N_2}^*.
\end{aligned} \tag{5.2}$$

The computational domain is defined as $\Omega^* = [0, 2\pi]^3$, with periodic boundary conditions applied on all boundaries. Tests were conducted with polynomial order $p = 3$ and grid resolution of $N = 4$ and 8 in each direction. An initial small time step $\Delta t^* = 1 \times 10^{-3}$ was employed to avoid numerical instabilities. Although previous mono-species or calorically perfect multi-species studies typically simulate this case up to around $t^* = 15$, the current simulations became unstable at approximately $t^* = 3.9$ for $N = 4$ and $t^* = 2.1$ for $N = 8$ grid resolutions. Instabilities occurred due to the N_2 mass fraction exceeding 1 or the O_2 mass fraction becoming negative, with the temperature also violating the NASA CAP limits. Repeated attempts to reduce the time step, halving it up to $\Delta t^* = 1.25 \times 10^{-4}$, failed to extend the simulation time. In Figure 5.1, the onset of vortex breakup is visible, signifying the transition to more complex flow structures. Concurrently, Figure 5.2 reveals the mixing phenomena, highlighting the diffusion between the two species. This numerical experiment demonstrates the three-dimensional multi-species computational capabilities of the current

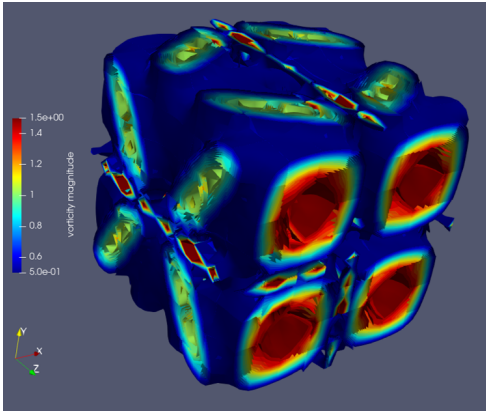
DG solver. However, as future work, alternative initial conditions or limiter strategies are necessary to achieve stable solutions for the entire simulation duration.



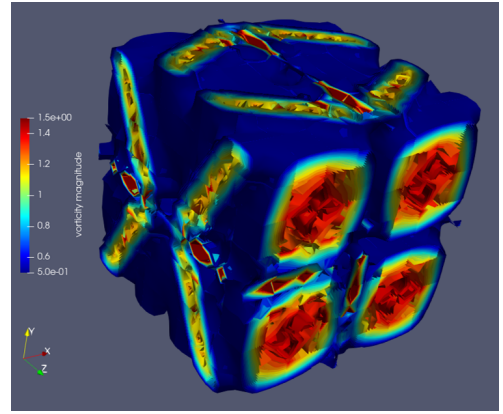
(a) At initial time



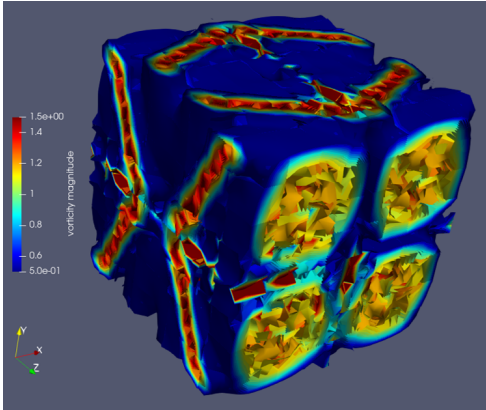
(b) At $t^* = 0.4$



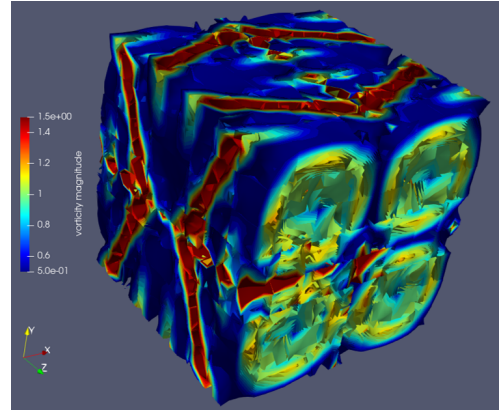
(c) At $t^* = 0.8$



(d) At $t^* = 1.2$

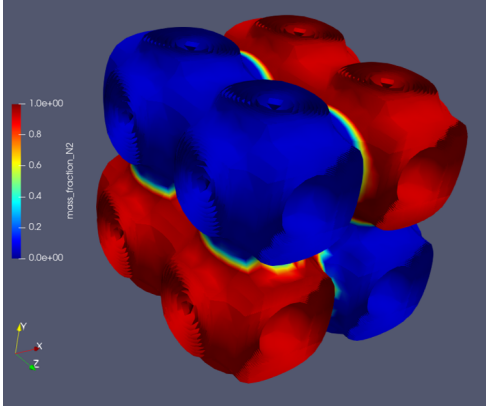


(e) At $t^* = 1.6$

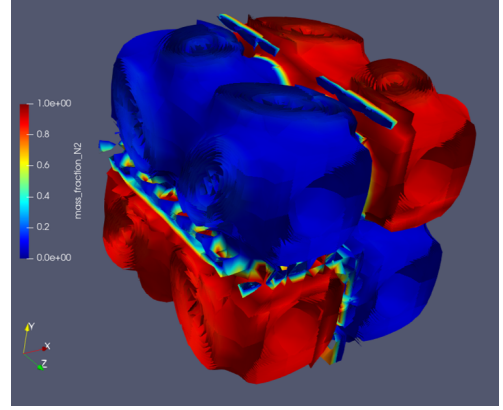


(f) At $t^* = 2.0$

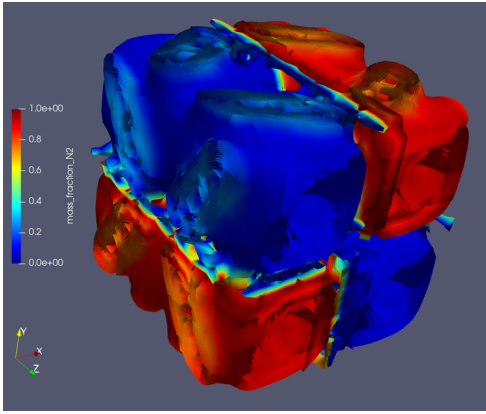
Figure 5.1: Contours of vorticity magnitude with the time progress using a $p = 3$ approximation and $N = 8$ grid resolution in each direction for the multi-species inviscid Taylor-Green vortex problem.



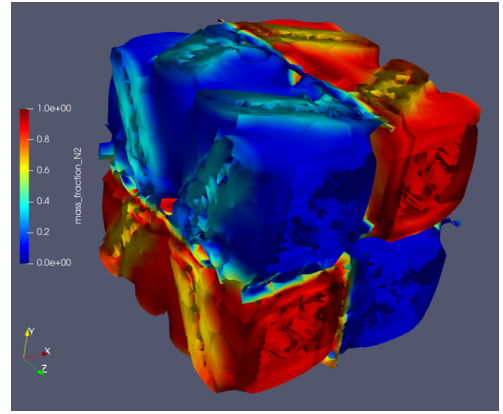
(a) At initial time



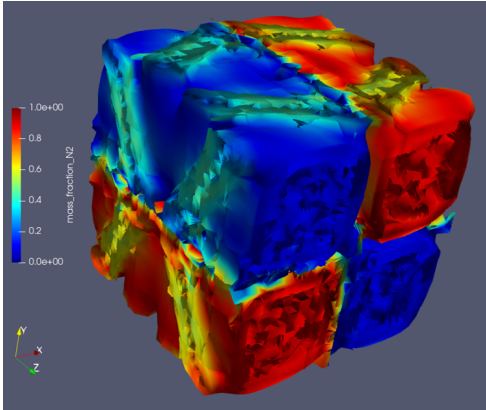
(b) At $t^* = 0.4$



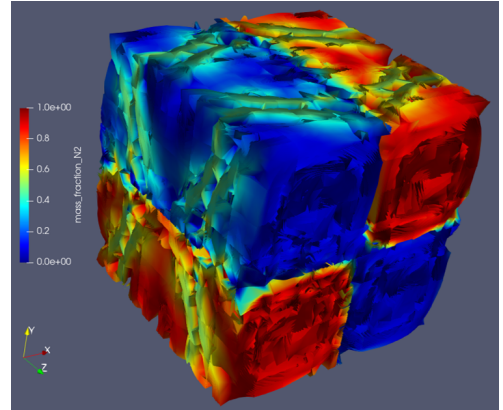
(c) At $t^* = 0.8$



(d) At $t^* = 1.2$



(e) At $t^* = 1.6$



(f) At $t^* = 2.0$

Figure 5.2: Contours of N_2 mass fraction with the time progress using a $p = 3$ approximation and $N = 8$ grid resolution in each direction for the multi-species inviscid Taylor-Green vortex problem.

Bibliography

- [1] Bonnie J McBride. *NASA Glenn coefficients for calculating thermodynamic properties of individual species*. National Aeronautics and Space Administration, John H. Glenn Research Center, 2002.
- [2] Michael J Zehe, Sanford Gordon, and Bonnie J McBride. Cap: A computer code for generating tabular thermodynamic functions from nasa lewis coefficients. Technical report, NASA, 2002.
- [3] Zhijian J Wang, Krzysztof Fidkowski, Rémi Abgrall, Francesco Bassi, Doru Caraeni, Andrew Cary, Herman Deconinck, Ralf Hartmann, Koen Hillewaert, Hung T Huynh, et al. High-order cfd methods: current status and perspective. *International Journal for Numerical Methods in Fluids*, 72(8):811–845, 2013.
- [4] Per-Olof Persson Krzysztof Fidkowski and Ngoc Cuong Nguyen. Special issue: Advances in high-order methods for cfd. *International Journal of Computational Fluid Dynamics*, 37(6):429–429, 2023.
- [5] Chuang-Chao Ye, Peng-Jun-Yi Zhang, Zhen-Hua Wan, Rui Yan, and De-Jun Sun. Accelerating cfd simulation with high order finite difference method on curvilinear coordinates for modern gpu clusters. *Advances in Aerodynamics*, 4(1):7, 2022.
- [6] Chi-Wang Shu. High-order finite difference and finite volume weno schemes and discontinuous galerkin methods for cfd. *International Journal of Computational Fluid Dynamics*, 17(2):107–118, 2003.

- [7] HT Huynh, Zhi J Wang, and Peter E Vincent. High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids. *Computers & fluids*, 98:209–220, 2014.
- [8] Thomas D Dunstan, Nedunchezhian Swaminathan, Ken NC Bray, and R 12Stewart Cant. Geometrical properties and turbulent flame speed measurements in stationary premixed v-flames using direct numerical simulation. *Flow, turbulence and combustion*, 87(2):237–259, 2011.
- [9] Jundika C Kurnia, Agus P Sasmito, and Arun S Mujumdar. Cfd simulation of methane dispersion and innovative methane management in underground mining faces. *Applied Mathematical Modelling*, 38(14):3467–3484, 2014.
- [10] Jerzy Baldyga and John R Bourne. *Turbulent mixing and chemical reactions*. John Wiley & Sons, 1999.
- [11] Dmitry Pashchenko. Comparative analysis of hydrogen/air combustion cfd-modeling for 3d and 2d computational domain of micro-cylindrical combustor. *International Journal of Hydrogen Energy*, 42(49):29545–29556, 2017.
- [12] MM Noor, Andrew P Wandel, and Talal Yusaf. Detail guide for cfd on the simulation of biogas combustion in bluff-body mild burner. In *Proceedings of the 2nd International Conference on Mechanical Engineering Research (ICMER 2013)*. University of Southern Queensland, 2013.
- [13] PA Batrakov, AN Mrakin, and AA Selivanov. Numerical study of the nitrogen oxide formation in non-circular profile furnaces of gas-tube boilers. In *2016 Dynamics of Systems, Mechanisms and Machines (Dynamics)*, pages 1–5. IEEE, 2016.
- [14] Zhi Zhang, Zhenshan Li, and Ningsheng Cai. Reduced-order model of char burning for cfd modeling. *Combustion and Flame*, 165:83–96, 2016.

- [15] Song Wang, Salil Mohan, and Edward L Dreizin. Effect of flow conditions on burn rates of metal particles. *Combustion and Flame*, 168:10–19, 2016.
- [16] Hassan I Kassem, Khalid M Saqr, Hossam S Aly, Mohsin M Sies, and Mazlan Abdul Wahid. Implementation of the eddy dissipation model of turbulent non-premixed combustion in openfoam. *International Communications in Heat and Mass Transfer*, 38(3):363–367, 2011.
- [17] Anna Tsynaeva and Maxim Nikitin. The technology of heat transfer enhancement in channels by means of flow pulsations. In *MATEC Web of Conferences*, volume 86, page 04018. EDP Sciences, 2016.
- [18] A Imren and DC Haworth. On the merits of extrapolation-based stiff ode solvers for combustion cfd. *Combustion and Flame*, 174:1–15, 2016.
- [19] Xinwei Cheng, Hoon Kiat Ng, Suyin Gan, Jee Hou Ho, and Kar Mun Pang. Development and validation of a generic reduced chemical kinetic mechanism for cfd spray combustion modelling of biodiesel fuels. *Combustion and Flame*, 162(6):2354–2370, 2015.
- [20] Dmitry Pashchenko. Thermodynamic equilibrium analysis of combined dry and steam reforming of propane for thermochemical waste-heat recuperation. *International Journal of Hydrogen Energy*, 42(22):14926–14935, 2017.
- [21] DI Pashchenko. Thermochemical recovery of heat contained in flue gases by means of bioethanol conversion. *Thermal Engineering*, 60(6):438–443, 2013.
- [22] Ronald P Fedkiw, Tariq Aslam, Barry Merriman, and Stanley Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of computational physics*, 152(2):457–492, 1999.
- [23] Ronald P Fedkiw, Antonio Marquina, and Barry Merriman. An isobaric fix for the overheating problem in multimaterial compressible flows. *Journal of Computational Physics*, 148(2):545–578, 1999.

- [24] David Adalsteinsson and James A Sethian. A fast level set method for propagating interfaces. *Journal of computational physics*, 118(2):269–277, 1995.
- [25] Stanley Osher, Ronald Fedkiw, and K Piechor. Level set methods and dynamic implicit surfaces. *Appl. Mech. Rev.*, 57(3):B15–B15, 2004.
- [26] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.
- [27] James A Sethian et al. *Level set methods and fast marching methods*, volume 98. Cambridge Cambridge UP, 1999.
- [28] Wei Liu, Li Yuan, and Chi-Wang Shu. A conservative modification to the ghost fluid method for compressible multiphase flows. *Communications in Computational Physics*, 10(4):785–806, 2011.
- [29] Rémi Abgrall. An extension of roe’s upwind scheme to algebraic equilibrium real gas models. *Computers & Fluids*, 19(2):171–182, 1991.
- [30] Vijaya Kumar, M Kampili, Stephan Kelm, K Arul Prakash, and Hans-Josef Allelein. Development and verification of a multi-species gas transport solver. In *Proceedings of the 14th OpenFOAM Workshop, Duisburg, Germany*, pages 23–26, 2019.
- [31] Lei Liang. Simulation of multispecies gas flows using the discontinuous galerkin method. <https://scholarsjunction.msstate.edu/td/3952>. 2012, Theses and Dissertations, Mississippi State University.
- [32] B. Merriman R. P. Fedkiw and S. Osher. High accuracy numerical methods for thermally perfect gasows with chemistry. *Journal of Computational Physics*, 132(2):175–190, 1997.
- [33] J. E. Shepherd J. L. Ziegler, R. Deiterding and D. I. Pullin. An adaptive high-order hybrid scheme for compressive, viscous flows with detailed chemistry. *Journal of Computational Physics*, 230(20):7598–7630, 2011.

- [34] Arnab Chaudhuri, Abdellah Hadjadj, Ashwin Chinnayya, and Sandrine Palerm. Numerical study of compressible mixing layers using high-order weno schemes. *Journal of Scientific Computing*, 47:170–197, 2011.
- [35] Pedro José Martínez Ferrer, Romain Buttay, Guillaume Lehnasch, and Arnaud Mura. A detailed verification procedure for compressible reactive multicomponent navier–stokes solvers. *Computers & Fluids*, 89:88–110, 2014.
- [36] Rolls Royce plc. *The jet engine*. John Wiley & Sons, 2015.
- [37] NASA. X-43a (hyper-x). <https://www.nasa.gov/image-article/x-43a-hyper-x/>. Accessed: 2024-07-12.
- [38] Travis S Taylor. *Introduction to rocket science and engineering*. CRC Press, 2017.
- [39] David W Witte and Kenneth E Tatum. Computer code for determination of thermally perfect gas properties. Technical report, NASA, 1994.
- [40] NASA Glen Research Center. Specific heat capacity calorically imperfect gas. <https://www.grc.nasa.gov/WWW/BGH/realspec.html>.
- [41] BK Hodge and Keith Koenig. Cig: A calorically imperfect compressible flow software element. *Computer Applications in Engineering Education*, 1(6):477–487, 1993.
- [42] Derek J Dalle, Matt L Fotia, and James F Driscoll. Reduced-order modeling of two-dimensional supersonic flows with applications to scramjet inlets. *Journal of Propulsion and Power*, 26(3):545–555, 2010.
- [43] Bonnie J McBride. *Coefficients for calculating thermodynamic and transport properties of individual species*, volume 4513. NASA Langley Research Center, 1993.
- [44] Piotr Lampart, Andrey Rusanov, Sergey Yershov, Stanislaw Marcinkowski, and Andrzej Gardzilewicz. Validation of a 3d rans solver with a state equation of thermally perfect

- and calorically imperfect gas on a multi-stage low-pressure steam turbine flow. *J. Fluids Eng.*, 127(1):83–93, 2005.
- [45] Georg May, Koen Devesse, Ajay Rangarajan, and Thierry Magin. A hybridized discontinuous galerkin solver for high-speed compressible flow. *Aerospace*, 8(11):322, 2021.
- [46] Zhenhua Jiang, Chao Yan, Jian Yu, Feng Qu, and Libin Ma. Effective high-order solver with thermally perfect gas model for hypersonic heating prediction. *Applied Thermal Engineering*, 99:147–159, 2016.
- [47] Robert H Nichols, Robert W Tramel, and Pieter G Buning. Evaluation of two high-order weighted essentially nonoscillatory schemes. *AIAA journal*, 46(12):3090–3102, 2008.
- [48] Will Trojak and Tarik Dzanic. Positivity-preserving discontinuous spectral element methods for compressible multi-species flows. *Computers & Fluids*, page 106343, 2024.
- [49] Andrés M Rueda-Ramírez, Aleksey Sikstel, and Gregor J Gassner. An entropy-stable discontinuous galerkin discretization of the ideal multi-ion magnetohydrodynamics system. *arXiv preprint arXiv:2402.14615*, 2024.
- [50] Yu Lv and Matthias Ihme. Discontinuous galerkin method for multicomponent chemically reacting flows and combustion. *Journal of Computational Physics*, 270:105–137, 2014.
- [51] Jie Du, Yong Liu, and Yang Yang. An oscillation-free bound-preserving discontinuous galerkin method for multi-component chemically reacting flows. *Journal of Scientific Computing*, 95(3):90, 2023.
- [52] Dongmi Luo, Shiyi Li, Weizhang Huang, Jianxian Qiu, and Yibing Chen. A quasi-conservative discontinuous galerkin method for multi-component flows using the non-oscillatory kinetic flux ii: Ale framework. *Journal of Scientific Computing*, 90(1):46, 2022.

- [53] Ayoub Gouasmi, Karthik Duraisamy, and Scott M Murman. Formulation of entropy-stable schemes for the multicomponent compressible euler equations. *Computer Methods in Applied Mechanics and Engineering*, 363:112912, 2020.
- [54] Doug Shi-Dong, Siva Nadarajah, Julien Brillon, Donovan Blais, Keigan MacLean, Pranshu Thakur, Carolyn Pethrick, Alexander Cicchino, and Maya Tatarelli. doughty-shi-dong/philip: Philip version 2.0.0. <https://zenodo.org/records/6600853>.
- [55] Donovan Blais. Goal-oriented adaptive sampling for projection-based reduced-order models. <https://escholarship.mcgill.ca/concern/theses/4f16c7697>. 2022, Master Thesis, McGill University.
- [56] Sandip Mazumder. *Numerical methods for partial differential equations: finite difference and finite volume methods*. Academic Press, 2015.
- [57] Thomas Höhne, Eckhard Krepper, and Ulrich Rohde. Application of cfd codes in nuclear reactor safety analysis. *Science and Technology of Nuclear Installations*, 2010(1):198758, 2010.
- [58] Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.
- [59] William H Reed and Thomas R Hill. Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Lab., N. Mex.(USA), 1973.
- [60] Bernardo Cockburn and Chi-Wang Shu. The local discontinuous galerkin method for time-dependent convection-diffusion systems. *SIAM journal on numerical analysis*, 35(6):2440–2463, 1998.
- [61] Bernardo Cockburn and Chi-Wang Shu. The runge–kutta discontinuous galerkin method for conservation laws v: multidimensional systems. *Journal of computational physics*, 141(2):199–224, 1998.

- [62] Bernardo Cockburn and Chi-Wang Shu. Runge–kutta discontinuous galerkin methods for convection-dominated problems. *Journal of scientific computing*, 16:173–261, 2001.
- [63] Paul Houston, Christoph Schwab, and Endre Süli. Discontinuous hp-finite element methods for advection-diffusion-reaction problems. *SIAM Journal on Numerical Analysis*, 39(6):2133–2163, 2002.
- [64] Ralf Hartmann and Paul Houston. Adaptive discontinuous galerkin finite element methods for the compressible euler equations. *Journal of Computational Physics*, 183(2):508–532, 2002.
- [65] Francesco Bassi and Stefano Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier–stokes equations. *Journal of computational physics*, 131(2):267–279, 1997.
- [66] Keigan MacLean. Unstructured anisotropic all-quad mesh adaptation based on a continuous local error model for the discontinuous galerkin method. <https://escholarship.mcgill.ca/concern/theses/gm80j1332>. 2020, Master Thesis, McGill University.
- [67] Pranshul Thakur and Siva Nadarajah. Adjoint-based goal-oriented implicit shock tracking using full space mesh optimization. *arXiv preprint arXiv:2405.00904*, 2024.
- [68] Peter D Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. In *Selected Papers Volume I*, pages 198–232. Springer, 2005.
- [69] Daniel Arndt, Wolfgang Bangerth, Bruno Blais, Thomas C Clevenger, Marc Fehling, Alexander V Grayver, Timo Heister, Luca Heltai, Martin Kronbichler, Matthias Maier, et al. The deal. ii library, version 9.2. *Journal of Numerical Mathematics*, 28(3):131–146, 2020.
- [70] Doug Shi-Dong. Full-space aerodynamic shape optimization. <https://escholarship.mcgill.ca/concern/theses/k643b688x>. 2024, PhD Thesis, McGill University.

- [71] Peter A Gnoffo. *Conservation equations and physical models for hypersonic air flows in thermal and chemical nonequilibrium*, volume 2867. National Aeronautics and Space Administration, Office of Management, 1989.
- [72] Julien Keith Louis Brillon. Modeling thermochemical nonequilibrium processes and flow field simulations of spark-induced plasma. Master’s thesis, Purdue University, 2020.
- [73] W.G.V. KRUGER and C.H. KRUGER. *Introduction to Physical Gas Dynamics*. New York, 1965.
- [74] Ernst Heinrich Hirschel. Real-gas aerothermodynamic phenomena. *Basics of Aerothermodynamics*, pages 101–133, 2005.
- [75] Bonnie J McBride and Sanford S. Gordon. Computer program for calculating and fitting thermodynamic functions. Technical report, NASA, 1992.
- [76] Heather E Dillon, Ashley Emery, RJ Cochran, and Ann Mescher. Dimensionless versus dimensional analysis in cfd and heat transfer. In *Excerpt from Proceedings of the COMSOL Conference 2010 Boston*, 2010.
- [77] Katate Masatsuka. *I do Like CFD, vol. 1*, volume 1. Lulu. com, 2009.
- [78] Bartosz Górecki. Quickersim cfd lecture notes. https://www.quickersim.com/_ftp/QuickerSim_CFD_course_notes.pdf. Accessed: 2024-07-03.
- [79] Subrahmanya Pavan Kumar Veluri. Code verification and numerical accuracy assessment for finite volume cfd codes. <https://vtechworks.lib.vt.edu/items/6cc361e4-5dcf-48d7-9de2-b12dd31dd741>. 2010, Doctoral Dissertations, Virginia Tech.
- [80] Jian-Hang Wang, Shucheng Pan, Xiangyu Y Hu, and Nikolaus A Adams. Partial characteristic decomposition for multi-species euler equations. *Computers & Fluids*, 181:364–382, 2019.

- [81] Qingyan Chen and Jelana Srebric. A procedure for verification, validation, and reporting of indoor environment cfd analyses. *Hvac&R Research*, 8(2):201–216, 2002.
- [82] Bernardo Cockburn, Chi-Wang Shu, Claes Johnson, Eitan Tadmor, and Chi-Wang Shu. *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, in: *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 1998.
- [83] Peter C Ma, Yu Lv, and Matthias Ihme. An entropy-stable hybrid scheme for simulations of transcritical real-fluid flows. *Journal of Computational Physics*, 340:330–357, 2017.
- [84] G. Bruneaux L. Pickett. Engine combustion network, combustion research facility, sandia national laboratories, livermore, ca. <https://ecn.sandia.gov/diesel-spray-combustion/target-condition/spray-ab>. Accessed: 2024-06-21.
- [85] Ryan F Johnson and Andrew D Kercher. A conservative discontinuous galerkin discretization for the chemically reacting navier-stokes equations. *Journal of Computational Physics*, 423:109826, 2020.
- [86] Geoffrey Ingram Taylor and Albert Edward Green. Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, 158(895):499–521, 1937.
- [87] Jean-Baptiste Chapelier, Marta De La Llave Plata, and Florent Renac. Inviscid and viscous simulations of the taylor-green vortex flow using a modal discontinuous galerkin approach. In *42nd AIAA fluid dynamics conference and exhibit*, page 3073, 2012.
- [88] Abouelmagd Abdelsamie, Ghislain Lartigue, Christos E Frouzakis, and Dominique Thevenin. The taylor-green vortex as a benchmark for high-fidelity combustion simulations using low-mach solvers. *Computers & Fluids*, 223:104935, 2021.