Predictive modeling of hypersonic flows in chemical nonequilibrium : physics-based and data-driven approaches

Clément Scherding

SORBONNE UNIVERSITÉ

École doctorale Sciences Mécaniques, Acoustique, Électronique et Robotique

THÈSE DE DOCTORAT
SPÉCIALITÉ MÉCANIQUE

Predictive modeling of hypersonic flows in chemical nonequilibrium: physics-based and data-driven approaches

---

Modélisation prédictive des écoulements hypersoniques hors-équilibre chimique: approches physiques et basées sur les données

Presentée par **Clément SCHERDING**

pour obtenir le grade de Docteur de Sorbonne Université

Sous la direction de Taraneh SAYADI, Chargée de Recherche
Georgios RIGAS, Maître de Conférence
Denis SIPP, Directeur de Recherche
Peter J. SCHMID, Professeur

à l'Institut Jean Le Rond ∂'Alembert, Sorbonne Université, CNRS, UMR 7190

Soutenue le 29/09/2023 devant le jury composé de :

M. Thierry MAGIN	Professeur	Rapporteur
M. Karthik DURAISAMY	Professeur	Rapporteur
Mme. Paola CINELLA	Professeure	Présidente du Jury
M. Corentin LAPEYRE	Chargé de Recherche	Examinateur
Mme. Taraneh SAYADI	Chargée de Recherche	Directrice de Thèse
M. Georgios RIGAS	Maître de Conférence	Co-encadrant
M. Denis SIPP	Directeur de Recherche	Co-encadrant
M. Peter J. SCHMID	Professeur	Co-encadrant

Remerciements

Je tiens à remercier, tout d'abord, le Centre National de la Recherche Scientifique de m'avoir permis de réaliser cette thèse dans le cadre du CNRS - Imperial College London joint PhD program. Mes remerciements vont également à l'Institut Jean le Rond d'Alembert dans son ensemble qui m'a accueilli dans ses locaux sur le campus de Jussieu.

Je remercie Corentin Lapeyre et Karthik Duraisamy pour leurs commentaires et questions qui ont été d'une grande précision et ont ouvert la voie à de nombreuses extensions possibles de ce travail.

Je suis extrêmement reconnaissant aux membres de mon comité de suivi, Paola Cinella et Thierry Magin, qui m'ont suivi tout au long de ces trois dernières années et qui ont su aiguiller mes recherches.

Je remercie chaleureusement tous mes directeurs de thèse, Denis Sipp, Georgios Rigas, Peter Schmid et Taraneh Sayadi. Comme le soulignait Peter, ce n'est pas toujours facile d'être le seul soldat au sein d'une armée de généraux, mais cela m'a profondément stimulé et contribué à l'amélioration de mon travail et je vous en remercie.

Merci à Denis pour son intérêt constant et ses critiques constructives à l'égard des résultats que je présentais. Merci à Peter et George de m'avoir accompagné lors du Stanford Center for Turbulence Research Summer Program 2022. Je ne saurais dire combien de problèmes nous avons résolus ensemble et ce travail n'en serait pas là aujourd'hui sans votre aide. Merci également de m'avoir fait découvrir vos universités respectives, je garderai toujours un très bon souvenir de ces expériences internationales. Enfin, merci à Taraneh Sayadi qui, dès notre première réunion, a su diriger avec brio cet ambitieux projet de recherche.

Je remercie tous mes collègues doctorants de d'Alembert avec qui j'ai travaillé et passé beaucoup de bons moments, que ce soit au labo, en conférence ou au bar: Camille, Cécile, Laureline, Yutao, Mikhail, Ariadni, Ludo, Simon, Rodolphe, Samy, Patricia, Cécilia, Camilla, Amine, Gonzalo, Quentin, Cécile et Franck, merci à vous ! Merci tout particulièrement à Serena, Tomas et Alejandro. Nous formions une magnifique équipe (de recherche) et j'en garderai de très bon souvenirs.

Je remercie Germain, Arthur, Luc, Julien et tous mes amis de Centrale de m'avoir fait relativiser, voire oublier mes petits problèmes de doctorant.

Je remercie également mes amis d'Alsace, notamment Tony, pour avoir toujours été là quand un retour aux sources était nécessaire.

Merci à mes grands-mères, Théa et Madeleine, à qui je dois beaucoup. Merci à mes parents, Karin et Christian, et à ma sœur, Mélanie, d'avoir toujours été à mes côtés.

Amélie, merci pour ton soutien infaillible et ton amour qui m'ont porté jusqu'ici. Je te dédie cette thèse, bien que ces quelques mots ne suffisent pas à exprimer pleinement toute la gratitude et l'amour que je ressens pour toi.

Modélisation prédictive des écoulements hypersoniques hors-équilibre chimique: approches physiques et basées sur les données

Résumé: Dans cette thèse, nous nous intéressons aux écoulements hypersoniques. Ces derniers sont d'un grand intérêt pour de nombreuses applications aérospatiales et constituent un élément essentiel des avancées technologiques dans ce domaine. En première partie, nous étudions numériquement l'interaction complexe entre la thermodynamique, la chimie et la mécanique des fluides dans les conditions extrêmes rencontrées aux vitesses hypersoniques. La prise en compte des effets thermochimiques hors-équilibre repose sur des modèles de gaz complexes. A cet effet, les propriétés gazeuses sont donc initialement modélisées à l'aide du logiciel Mutation++. Cependant, l'utilisation de ce type de modèle augmente considérablement le coût des calculs. Dans la deuxième partie, nous présentons donc une nouvelle technique d'apprentissage automatique pour extraire de manière adaptative un modèle d'ordre réduit de la thermochimie d'un mélange de gaz. Les états thermodynamiques sont projetés dans un espace à faible dimension et regroupés pour identifier les régions présentant différents niveaux d'équilibre thermochimique. Ensuite, un modèle de substitution est construit entre l'espace de projection et l'espace de sortie à l'aide de réseaux de fonctions à base radiale. La méthode est validée sur des simulations de couches limites hypersoniques et des interactions onde de choc - couche limite avec des réactions chimiques hors-équilibre. Remplacer Mutation++ par le modèle d'ordre réduit améliore les performances du code de calcul jusqu'à 70% tout en maintenant la précision globale. Grâce aux capacités d'apprentissage en ligne, la généralisation et la prédictibilité du modèle sont démontrées lorsque les données d'apprentissage ne sont que partiellement disponibles. La méthode peut être facilement portée à d'autres domaines reposant sur des évaluations de fonctions coûteuses en haute dimension.

Mots-clés: Aérodynamique hypersonique, effets hors-équilibre thermochimique, modèles d'ordre réduit, techniques d'apprentissage machine, apprentissage actif, modèles de substitution

Predictive modeling of hypersonic flows in chemical nonequilibrium: physics-based and data-driven approaches

Abstract: Hypersonic flows are of great interest in a wide range of aerospace applications and are a critical component of many technological advances. In this thesis, we first numerically study the complex interplay between thermodynamics, chemistry and fluid mechanics in the extreme conditions encountered at hypersonic speeds. Accurate simulations of these flows in thermochemical (non)-equilibrium rely on detailed thermochemical gas models. Thus, the gas properties of the reactive mixture are initially modeled using the open-source Mutation++ library. While accurately capturing the underlying aerothermochemistry, the library dramatically increases the cost of such calculations. In the second part, a novel model-agnostic machine-learning technique is presented which allows an adaptive extraction of a reduced-order thermochemical model of a gas mixture from the library. The states are embedded in a low-dimensional space and clustered to identify regions with different levels of thermochemical (non)-equilibrium. Then, a surrogate surface from the reduced cluster-space to the output space is generated using radial-basis-function networks. The method is validated and benchmarked on simulations of hypersonic flat-plate boundary layers and shock-wave boundary layer interaction with finite-rate chemistry. Substituting Mutation++ with the light-weight, machine-learned alternative improves the performance of the solver by up to 70% while maintaining overall accuracy. Thanks to the online learning capabilities of the technique, generalization and predictivity of the model are demonstrated when training data is only partially available. The method can be readily ported to other domains relying on expensive, high-dimensional function evaluations.

Keywords: Hypersonic aerodynamics, thermochemical nonequilbrium effects, reduced-order models, machine learning, active learning, surrogate modeling

Contents

List of Figures

List of Tables

List of Acronyms

CFD Computational Fluid Dynamics

DNS Direct Numerical Simulation

LES Large Eddy Simulation

ROM Reduced-Order Model

FOM Full-Order Model

RBF Radial Basis Functions Network

PCA Principal Component Analysis

POD Proper Orthogonal Decomposition

PLS Partial Least-Squares

PCE Polynomial Chaos Expansion

SVD Singular Value Decomposition

SBLI Shock-Wave Boundary Layer Interaction

JISC Jet in High-Speed Crossflow

RRHO Rigid-Rotor Harmonic oscillator

CPG Calorically Perfect Gas

TPG Thermally Perfect Gas

CNEQ Chemical Nonequilibrium

TCNEQ Thermochemical Nonequilibrium

LTE Local Thermodynamic Equilibrium

AE Auto-Encoder

IO-E Input/Output-Encoder

KDE Kernel Density Estimation

LMS Least Mean Squares Filters

RLS Recursive Least Squares

EKF Extended Kalman Filters

VKI Von Karman Institute for Fluid Dynamics

LuT Look-up Table

ISAT In Situ Adaptive Tabulation

CFL Courant-Friedrichs–Lewy Condition

LAD Local Artificial Diffusivity

RMS Root-Mean-Squared

PSD Power Spectrum Density

RAN Resource-Allocating Network

Introduction

The revival of interest in complex orbital and interplanetary missions, as well as the pursuit of commercial hypersonic flight have pushed the boundaries of vehicle design to unprecedented speeds and extreme conditions. Consequently, there is a pressing requirement for accurate and reliable design of atmospheric re-entry and hypersonic cruise vehicles. In this context, hypersonic boundary layers have emerged as a significant research and engineering challenge. As a result, there is an increasing necessity for in-depth investigations in the field of hypersonic aerothermochemistry to tackle these challenges effectively [Bertin & Cummings 2003, Schmisseur 2015, Leyva 2017, Theofilis *et al.* 2022].

In fact, an object flying at hypersonic speed is surrounded by an extremely complex flow environment. In addition to the classical compressible flow features, the dissipation of massive amount of kinetic energy within the boundary layer introduces highly energetic gas states. These high temperatures induce chemical reactions and vibrational relaxation phenomena, which evolve in a flow dominated by extremely short time scales. At the molecular level, there is no guarantee that the collisions are sufficiently frequent for the energy exchange and chemical process to reach equilibrium. Thus, the composition and properties of the gas can vary in space and time, and the flow departs from thermal and/or chemical equilibrium (TCNEQ). These nonequilibrium effects can significantly influence the flow behavior, heat transfer, turbulence and chemical kinetics in hypersonic environments [Holden 1986, Leyva 2017]. A schematic of various thermochemical nonequilibrium effects within a generic hypersonic flows over a re-entry vehicle is presented in Figure 1.

It is notoriously difficult and expensive to collect reliable experimental and in-flight data for hypersonic flows [Schneider 2008]. In ground facilities, most experimental studies are carried out at low temperatures to achieve hypersonic speeds, thereby eliminating potential high-enthalpy effects [Lu 2002]. Hence, the need for precise and dependable numerical simulations to accurately characterize and predict the flow environment becomes increasingly crucial. Conventional computational fluid dynamics (CFD) codes and tools often fall short in capturing the intricate and complex nature of hypersonic flows in nonequilibrium, mostly due to the absence of models capable of handling the multiphysics phenomena involved. In fact, computational work on hypersonic boundary layers focuses mainly on calorically or thermally perfect gases [Martin 2007, Duan *et al.* 2010, Duan *et al.* 2011, Franko & Lele 2013, Hildebrand *et al.* 2018, Fu *et al.* 2021].

However, in recent years, there has been increasing studies to understand the nature and impact of thermochemical nonequilibrium in hypersonic flows. The numerical simulation of the complex interplay between thermodynamics, chemistry and fluid mechanics in these extreme conditions is a difficult modeling challenge and relies on detailed thermochemical gas models. Early work

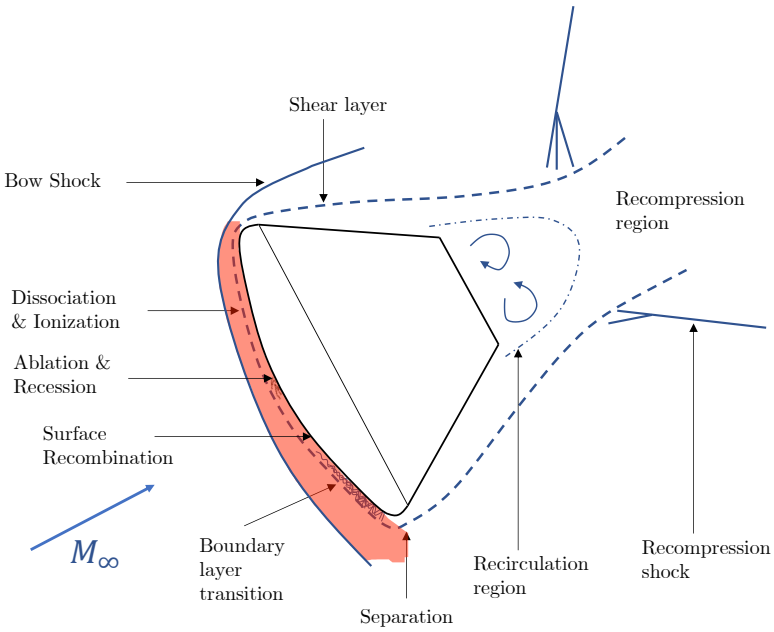


Figure 1: Schematic of a generic re-entry vehicle flying at hypersonic speed with a non-exhaustive list of various thermochemical nonequilibrium effects, adapted from [Scanlon *et al.* 2015].

focused on extending the concepts from subsonic and supersonic flow to the hypersonic regime for transition and instability evolution in laminar boundary layers [Malik 1989, Malik *et al.* 1990, Malik & Anderson 1991, Chang *et al.* 1997, Johnson *et al.* 1998, Johnson & Candler 2005]. These studies have demonstrated that the incorporation of chemical heat absorption caused by dissociation within the boundary layer noticeably retards the amplification of disturbances and postpones the transition to turbulence. Direct numerical simulations (DNS) also supported these findings by considering the early stage of transition [Marxen *et al.* 2011a, Marxen *et al.* 2013, Marxen *et al.* 2014b, Duan & Martin 2011] and recently up to fully turbulent boundary layers [Urzay & Di Renzo 2021, Di Renzo & Urzay 2021, Passiatore *et al.* 2021, Passiatore *et al.* 2022, Passiatore *et al.* 2023]. Figure 2 illustrates the potential non-equilibrium effects occurring within a wall-cooled, turbulent hypersonic boundary layer. These studies have identified these effects as cause of order-one change in growth-rate, response behavior and sensitivities even though the effect on the first-order statistics were modest. The work performed at the Von Karman Institute (VKI) [Zanus *et al.* 2017, Miró Miró & Pinna 2017, Miró Miró *et al.* 2018a, Wartemann *et al.* 2018, Miró Miró *et al.* 2018b, Wartemann *et al.* 2019, Miró Miró & Pinna 2019] nuanced these results. The authors concluded that the inadequate modeling of transport phenomena could result in inaccuracies comparable to completely disregarding chemical activity. This wide body of literature led to the comprehensive review of Candler [Candler 2019] on the subject.

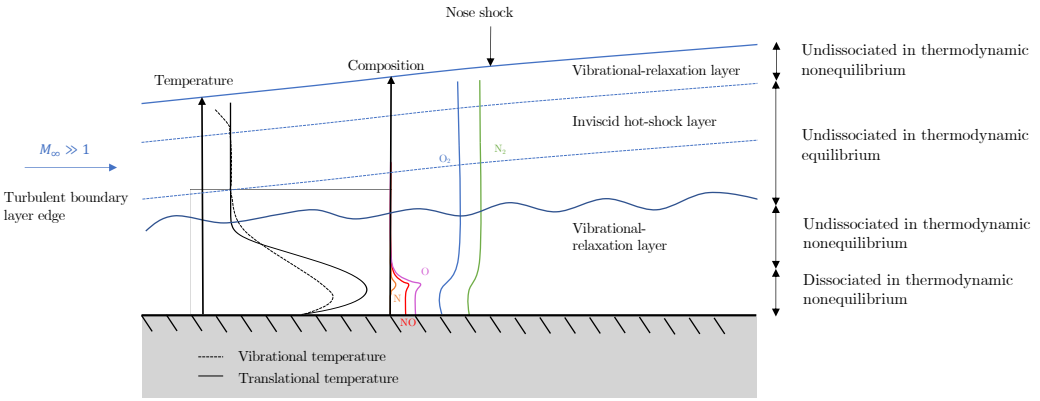


Figure 2: Schematic of a flat-plate turbulent hypersonic boundary layer with potential nonequilibrium effects. Adapted from [Urzay & Di Renzo 2021].

In the continuum regime, the Navier-Stokes equations still hold for hypersonic reacting gases, with some necessary modifications for the gas properties and the addition of extra equations to track a range of species in their inert or ionized forms. Complementing the hydrodynamic state vector by chemical components is a well-established technique, for example in combustion or atmospheric simulations. However, the required modeling of the inter-species interactions, such as dissociation, reaction and recombination for hypersonic applications poses great challenges [Anderson 2019, Candler 2019]. In fact, the validation of these models is arduous since the underlying processes are almost impossible to reproduce in ground facilities. Much of this modeling is accomplished by lookup libraries, which act as repositories of tabulated chemical reactions encountered for a given flow state [Scoggins *et al.* 2020]. When passing state-vector components to the library, amplitudes and time-scales for various forcing terms are returned, appearing as exogeneous inputs to the momentum, energy and species transport equations.

Much effort has gone into these libraries such as Pegase [Bottin *et al.* 1999], Eglib [Ern & Giovangigli 2004], Plato [Munafò *et al.* 2020] and CHEMKIN [Kee *et al.* 2000]. For aerothermochemical nonequilibrium effects in hypersonic flows, the Mutation++ library (MUlticomponent Thermodynamic And Transport properties for IONized gases in C++), developed and maintained at the VKI, has become the standard for high-fidelity simulations of high-speed and high-enthalpy flows [Scoggins *et al.* 2020]. This library can be coupled to existing flow solvers and is capable of modeling a range of partially ionized gas effects, together with nonequilibrium features, energy exchange processes and gas-surface interactions.

Part 1: Thus, the first objective of this work is to develop a high-fidelity computational tool coupled with Mutation++ to study chemical nonequilibrium effects in canonical hypersonic flows.

The flexibility and scope of the library comes at the expense of slowing down a typical large-scale simulation by a large factor, as shown in Figure 3, where typical simulation times for calorically and thermally perfect gases are juxtaposed with results for non-equilibrium chemical reactions. A wide margin can be observed. For this reason, non-equilibrium computations range among the most inefficient and laborious calculations in fundamental hypersonic research. To increase performance, most CFD codes use hard-coded chemistry [Di Renzo *et al.* 2020, Passiatore *et al.* 2021]. However, any change in the gas mixture or the thermochemical model comes at a human cost in terms of development, implementation and validation.

In the recent years, few studies have tried to make simulations in that flow regime faster using various machine learning techniques. For example, [Mao *et al.* 2021] and [Gkimisis *et al.* 2023] used respectively DeepONet and artificial neural network to predict the coupled flow in chemical nonequilibrium past a normal shock. [Zanardi *et al.* 2022] used physics-informed DeepONet to reduce the stiff master equations (equivalent to the thermochemical model in a state-to-state kinetic framework) into a ML-based surrogate. These authors reported up to 2 orders of magnitude faster prediction time but these studies were restricted to simple 0D or 1D configurations. Alternatively, other authors pursued the hybridization of the CFD code by replacing only the expensive thermochemical library with a ML-based surrogate model. [Novello *et al.* 2022] used deep neural networks to predict the output of Mutation++ in realistic 2D reenty configuration in chemical equilibrium. They observed speed up factors between 10 and 18.6.

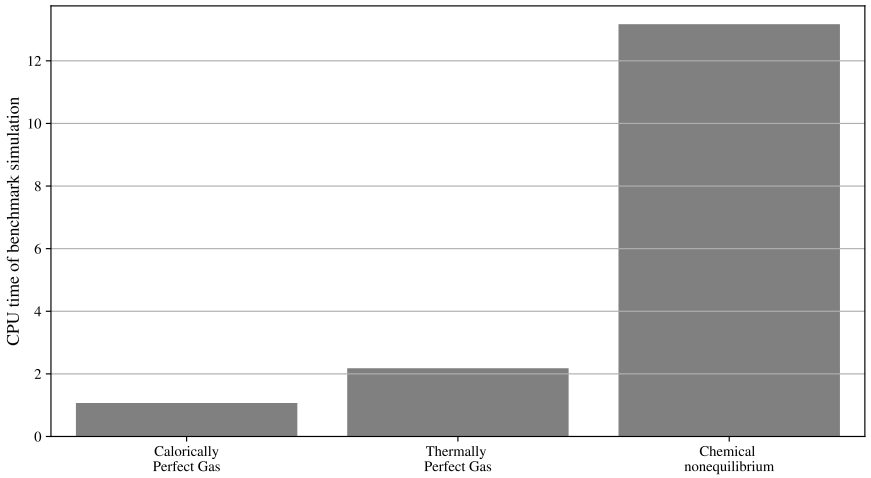


Figure 3: CPU time of a benchmark simulation, run with different aerothermochemical models. Including non-equilibrium effects in the simulation causes a significant increase in computational time.

More generally, many engineering applications need to evaluate an expensive function $\tilde{f}(\mathbf{x})$ many times. Therefore, it is of great interest to alleviate the CPU

burden of these applications by finding an efficient approximation of such functional forms, also known as a reduced-order model (ROM), instead of the full-order model (FOM). We here stress that we focus on static ROMs in the sense that they represent an input/output relation, as opposed to ROMs of dynamical systems that predict the time evolution of the system with a reduced number of degrees of freedom [Schmid 2010, Taira *et al.* 2017, Brunton *et al.* 2016, Cenedese *et al.* 2022].

One of the oldest and most common approach to approximate $\tilde{f}$ is to use structured tabulation. In a pre-processing step, values of $\tilde{f}$ are tabulated for a hypercube in the input space. Then, during the simulation, values of $\tilde{f}$ are linearly interpolated in the table. The Look-up Table method (LuT) was proven successful in many applications, such as tabulated chemistry for spray combustion [Franzelli *et al.* 2013], design of energy devices using organic Rankine Cycles [Pini *et al.* 2015] or simulations of hypersonic boundary layers in chemical equilibrium [Marxen *et al.* 2011b]. However, building and storing the table, together with the look-up procedure during the simulation, becomes computationally more intensive as the number of dimensions D of the input space increases. This demonstrates the well-known curse of dimensionality, where the volume of sample points needed to construct an accurate table increases exponentially with the number of dimensions of the input space. Similarly, linear interpolation in high dimensions is a tedious task. This latter point even prevents the application of this LuT-methodology in the case considered in this study where the input space dimension is $D = 6$. Pope developed the ISAT algorithm (In Situ Adaptive Tabulation) to overcome this deficiency in high dimensions with a storage/retrieval approach and demonstrate the concept on applications in the combustion field [Pope 1997].

Recently, more general data-driven methods that can tackle higher dimensional problems have also been proposed and saw considerable success in a variety of applications, particularly in the active research field known as surrogate modeling. Underlying this effort is the universal approximation theorem [Cybenko 1989] which proves that deep neural networks, with at least one hidden layer and non-linear activation functions, formally proposed by LeCun [LeCun 1985], can approximate any non-linear function of any dimension. In the fluid dynamics community, neural networks and other machine learning techniques have been used to improve the longstanding turbulence closure modeling problem, see the recent review by Duraisamy et al. [Duraisamy *et al.* 2019]. For instance, Lapeyre et al. [Lapeyre *et al.* 2019] used convolutionnal neural networks to estimate the turbulent sub-grid scale reaction rates in combustion applications. Liu et al. [Liu & Batill 2000] used neural-networks for surrogate-model-based optimization in aeronautics. However, the training cost of the network by back propagation becomes prohibitive as the number of neurons and layers increase – a necessity which might arise in complex high-dimensional problems. Radial basis functions networks (RBF), a special case of three layers neural networks [Broomhead & Lowe 1988, Buhmann 2000], can also be used for nonlinear function approximation in any dimension. Their training is easier and cheaper than classical neural networks as the optimal weights can be found by solving a linear system of equations.

RBFs have been widely used for surrogate modeling in many fields such as aerodynamic shape optimization [Jin *et al.* 2001, Peter *et al.* 2007] and meteorology [Chang *et al.* 2001], to name but two. Statistical surrogate modeling techniques have also found great success as they directly include an estimation of the error in the model. The method of kriging, originally developed for two-dimensional geostatistics problems [Krige 1951], has been extended to approximate input/ouput problems of any dimension by Sacks *et al.* [Sacks *et al.* 1989]; see the review by Kleijnen on the use of kriging for surrogate modeling [Kleijnen 2009]. Finally, Polynomial Chaos Expansion (PCE) is another technique that can generate surrogate models well suited for uncertainty quantification [Soize & Ghanem 2004].

Despite some success, the often brute-force nature of these algorithms may not always yield a satisfactory surrogate model in terms of accuracy and computational cost. Bouhlel et al. [Bouhlel *et al.* 2016] pointed out several performance issues when performing kriging in high dimensions ($D = 100$). This number of dimensions is common in reactive flow simulations where hundreds of species are tracked, even with reduced chemical mechanisms [Attili *et al.* 2014, Bansal *et al.* 2015, Bhagatwala *et al.* 2014]. Moreover, one common assumption in surrogate modeling relates to the smoothness of the approximated relation. This is not always true, especially in hypersonic applications where shocks and temperature discontinuities are amongst the typical features of such flows. Therefore, clever pre-processing steps can greatly improve the model's performance in these cases. For example, Bouhlel et al. [Bouhlel *et al.* 2016] coupled kriging with partial least-squares (PLS) methods to reduce the high-dimensional ($D = 100$) input space. In [Hawchar *et al.* 2017], principal component analysis (PCA) has been used as a pre-processing step before applying polynomial chaos expansions on the PCA basis [Hawchar *et al.* 2017]. When dealing with discontinuous functions, Bettebghor *et al.* [Bettebghor *et al.* 2011] proposed to cluster the input basis into different regions (to avoid a discontinuity within a cluster) and build a surrogate model on each of these regions. All models are then combined together and form a mixture of experts, as described in the literature [Hastie *et al.* 2009]. Yang [Yang 2003], however, pointed out that combining surrogate surfaces does not necessarily outperform a single model fitted over the entire input space. Hence, special care has to be taken in combining these steps.

Finally, even with thorough training, ROMs typically face limitations when it comes to extrapolation beyond their training range. This issue is widely recognized as a common challenge for data-driven techniques. Consequently, ROMs cannot be considered truly predictive or highly generalizable since their outputs may deviate significantly and yield incorrect results when confronted with inputs that fall outside the range of observed data. In [Mao *et al.* 2021, Zanardi *et al.* 2022, Gkimisis *et al.* 2023], the authors used inexpensive 0D and 1D simulations to extend the training range of their models. In the work of [Novello *et al.* 2022], the chemical equilibrium assumption led the authors to use less expensive perfect gas simulations to approximate the training range. In fact, the perfect gas and chemical equilibrium thermochemical models use the same input features. This

strategy is, however, not applicable to a nonequilibrium setting since the additional inputs (the local chemical composition) are impossible to predict beforehand. In this case, expanding the training range with additional realizations is too expensive, particularly when it involves a cost-intensive multi-dimensional nonequilbrium CFD simulation. This would also counteract the purpose of the ROM. To mitigate this issue, additional techniques and approaches have been proposed to enhance the ROMs' extrapolation capabilities. To that end, adaptive reduced-order models are a promising solution. Rather than being confined to a specific operating window or application, they have the capacity to learn dynamically and refine the model on-the-fly, enabling broader applicability and superior results in a wide range of applications. In the dynamical systems community, recent research focused on building adaptive ROMs that can learn new dynamics in situ [Peherstorfer 2020, Yano *et al.* 2021, Ramezanian *et al.* 2021, Huang & Duraisamy 2023]. For static ROMs, the ISAT algorithm [Pope 1997] is an adaptive LuT that can learn in situ new inputs/outputs relation if an error metric is satisfied. However, the algorithm relies on linear interpolation only. RBFs capable of on-the-fly learning have also seen some development [Platt 1991, Kadirkamanathan & Niranjan 1993, Karayiannis & Mi 1997, Huang *et al.* 2005, Bortman & Aladjem 2009].

Part 2: The second objective of this work is therefore to develop adaptive reduced-order thermochemical gas model capable of replacing the computationally expensive library and the memory-intensive look-up tables when modeling inter-species interactions and transport phenomena in the simulations of hypersonic flows in chemical nonequilibrium.

Original contributions in this work The first contribution of this work is the development of a high-fidelity computational tool to study chemical nonequilibrium effects in hypersonic flows. In Chapter 3, we present notably the first DNS of a jet in a hypersonic crossflow at high-enthalpy conditions. These findings reinforce the significance of accurate thermochemical modeling in order to achieve accurate predictions of hypersonic flows.

The major contributions of our work is the derivation of adaptive reduced-order model to improve the efficiency of simulation using such models. By employing a novel combination of pre-processing techniques from machine learning, efficient and accurate reduced-order models are achieved. These models are further enhanced through the integration of active learning procedures, ensuring their generalizability and predictive capabilities. We believe that the technique can be readily ported and benefit other fields relying on expensive high-dimensional functions evaluation.

Outline The first part of this work will be focused on physics-based simulation of hypersonic flows in chemical nonequilibrium. In Chapter 1, the equations governing the flow regimes discussed in this work are recalled. Then, the various terms and modeling choices to close the governing equations are discussed (i.e. thermochemical

modeling). Chapter 2 presents the numerical framework used to solve the governing equations. A study of the impact of each thermochemical model on the cost of a benchmark simulation is also carried out. In Chapter 3, the newly developed flow solver is first validated with a set of two $Ma = 10$ boundary layers with different wall boundary conditions and a $Ma = 2$ shock-wave boundary layer interaction (SBLI). Then, more complex configurations are studied, such as roughness, SBLI at higher Mach number and jet in crossflow. An analysis of the impact of the thermochemical model on such flows will also be carried out. Finally, Chapter 4 is pivotal. It presents the self-similar solutions used as reference solution in the sponge. These solutions will also be used as a low-cost way to generate training data in the second part of this work.

In the second part, focus will be set on data-driven techniques and particularly the derivation of reduced-order thermochemical gas models to speed up the simulation of hypersonic flows in chemical nonequilibrium. In Chapter 5, the strategy and training steps of the reduced-order models are presented. The data-driven models are then tested in the steady simulation of three benchmark cases as a proof of concept. Chapter 6 discusses the generalization of the technique to unsteady flows or when training data is not or only partially available. This introduces Chapter 7, where a procedure for 'on-the-fly' active learning is presented, making the reduced-order thermochemical models generalizable and predictive.

Part I

Numerical simulation of hypersonic flows in chemical nonequilibrium

Chapter 1

Governing equations

Contents

In the following, we describe the general equations governing the flow regimes discussed in this work. Our starting point is an overview of the conservation equations, followed by a detailed discussion of the various terms and modeling choices.

1.1 Governing equations

The nondimensional Navier-Stokes equations for fluids that consist of a mixture of n_s species, S, are presented in Eqs. (1.1-1.4). The derivation of these equations is present in many classical text, see for instance [Anderson 2019]. Eq. (1.1) is the continuity equation, which ensures global mass conservation in the system. Eqs. (1.2) represent the set of mass conservation equations for each species, where the right-hand side contains the net production rate terms, $\dot{\omega}_s$. For non-reacting gas mixtures, where the mixture composition is constant or a direct function of the thermodynamic state, only the global mass conservation equation, Eq. (1.1), is required, and the species mass conservation equations, Eqs. (1.2), can be omitted. However, for finite-rate reacting mixtures with varying composition, Eq. (1.1) must be solved together with Eqs. (1.2). In order to ensure mass conservation, the set of equations presented in Eqs. (1.2) is commonly solved for all but one species. The excluded species is typically chosen based on numerical considerations, where the species with the smallest concentrations is often avoided. Alternatively, all the species mass conservation equations can be solved while Eq. (1.1) is relaxed. In this work, the former procedure is adopted and $n_s - 1$ equations are solved for the species. Finally, Eqs. (1.3) and (1.4) represent the conservation of momentum and total energy in the system, respectively. In total, there are $n_s + 4$ conservation equations.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1.1}$$

$$\frac{\partial \rho_s}{\partial t} + \nabla \cdot (\rho_s(\mathbf{u} + \mathbf{V}_s)) = \dot{\omega}_s, \quad s \in S, \tag{1.2}$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \tau, \tag{1.3}$$

$$\frac{\partial \rho e_0}{\partial t} + \nabla \cdot (\rho (e_0 + P)\mathbf{u}) = \nabla \cdot (\tau \cdot \mathbf{u}) - \nabla \cdot \mathbf{q}, \tag{1.4}$$

The corresponding nondimensional quantities are the velocity components $\mathbf{u} = [u, v, w]^T$, t denotes time, ρ denotes the mixture density, and $\rho_s = \rho Y_s$ and Y_s are the partial density and mass fraction of species $s \in S$, respectively. Alternatively, the mole fraction X_s can be computed from the mass fraction as

$$X_s = Y_s \frac{\overline{W}}{W_s}, \tag{1.5}$$

where W_s and $\overline{W}^{-1} = \sum_{s \in S} Y_s / W_s$ are species-specific and mixture-averaged molar mass, respectively. In the momentum equation, Eq. (1.3), p stands for pressure, and τ is the viscous stress tensor, defined for a Newtonian fluid as

$$\tau = \frac{\mu}{Re_\infty} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T - \frac{2}{3} (\nabla \cdot \mathbf{u}) \mathbf{I} \right) + \left(\lambda + \frac{2}{3} \mu \right) (\nabla \cdot \mathbf{u}) \mathbf{I} \tag{1.6}$$

where μ is the dynamic viscosity, $\beta = \lambda + 2\mu/3$ is the bulk viscosity and $\mathbf{I}$ is the identity tensor. Stoke's hypothesis is used and $\beta = 0$. In the energy equation, Eq. (1.4), $e_0 = e + |\mathbf{u}|^2/2$ is the stagnation (or total) energy, with e denoting the specific internal energy. Additionally, the specific internal and total enthalpies are defined as $h = e + p/\rho$ and $h_0 = e_0 + p/\rho$, respectively. The heat flux vector takes the form

$$\mathbf{q} = -\frac{\kappa}{Re_\infty Pr_\infty Ec_\infty} \nabla T + \sum_{s \in S} \rho_s h_s \mathbf{V}_s, \tag{1.7}$$

where T denotes the temperature. The second term on the right-hand side of Eq. (1.7) is included only in the case of finite-rate chemistry (see §1.2.1.3). In this case, the term $\mathbf{V}_s$, appearing in Eqs. (1.2) and (1.7), denotes the diffusion velocity vector of species s, while h_s represents the species specific partial enthalpy.

The set of equations are non-dimensionalized using the reference parameters in Eq. (1.8), where $\tilde{T}_{ref} = (\gamma_\infty - 1)\tilde{T}_\infty$ and $\tilde{c}_\infty$ represents the speed of sound. Dimensional quantities are denoted with the superscript tilde.

$$\begin{gathered}
\rho = \frac{\tilde{\rho}}{\tilde{\rho_\infty}},\ p = \frac{\tilde{p}}{\tilde{\rho_\infty} \tilde{c}_\infty^2},\ T = \frac{\tilde{T}}{\tilde{T_{ref}}},\ \mathbf{u} = \frac{\tilde{\mathbf{u}}}{\tilde{c_\infty}},\ \rho_s = \frac{\tilde{\rho_s}}{\tilde{\rho_\infty}},\ \mathbf{V}_s = \frac{\tilde{\mathbf{V}}_s}{c_\infty} \\
\kappa = \frac{\tilde{\kappa}}{\tilde{\kappa_\infty}},\ \mu = \frac{\tilde{\mu}}{\tilde{\mu_\infty}},\ e_0 = \frac{\tilde{e}_0}{\tilde{c}_\infty^2},\ h_0 = \frac{\tilde{h}_0}{\tilde{c}_\infty^2},\ \dot{\omega_s} = \frac{\tilde{\omega_s}}{\tilde{\rho}_\infty \tilde{c}_\infty / \tilde{L}_{ref}}
\end{gathered} \tag{1.8}$$

The Reynolds number, Re_∞, and Prandtl number, Pr_∞, are defined in the free stream at the domain inlet. The Eckert number, Ec_∞, is also computed at the free

stream and is set to one. These numbers are defined using reference values as

$$Re_\infty = \frac{\tilde{\rho}_\infty \tilde{c}_\infty \tilde{L}_{ref}}{\tilde{\mu}_\infty} , \quad Pr_\infty = \frac{\tilde{\mu}_\infty \tilde{c}_{p,\infty}}{\tilde{\kappa}_\infty} , \quad Ec_\infty = \frac{\tilde{c}_\infty^2}{\tilde{c}_{p,\infty} \tilde{T}_{ref}}. \tag{1.9}$$

The thermodynamic properties and equation of state, as well as the transport properties and the chemical kinetics terms, depend on the modeling choices and are discussed in detail in the following §1.2.

1.2 Thermodynamic and chemical models

The solver employed in this study utilizes several thermochemical models that vary in complexity. In the subsequent sections, we elaborate on the specifics of each modeling approach and analyze the consequences of the underlying assumptions on the closure of the governing equations.

1.2.1 Multi-component gas mixtures with variable composition

In general, a reacting gas in a high-enthalpy flow is perceived as a multi-component mixture comprising a specific set of species, denoted as S, which interact through a predetermined network of reactions, as described in §1.2.1.3. Each component of the mixture is assumed to behave as a perfect gas where each particle has negligible interactions with the surrounding particles. This assumption is valid for weakly ionized and unmagnetized gases considered in this work.

1.2.1.1 Thermodynamics

Under this assumption, the classical equation of state for perfect gas for the mixture reads,

$$\tilde{p} = \tilde{\rho} R_u \tilde{T} / \overline{W}, \tag{1.10}$$

where R_u is the universal gas constant. The global thermodynamic properties of the mixture can be inferred from the properties of the individual species as,

$$\rho = \sum_{s \in S} \rho_s , \quad e = \sum_{s \in S} Y_s e_s , \quad h = \sum_{s \in S} Y_s h_s . \tag{1.11}$$

The species specific heats are defined as,

$$c_{p,s} = \left(\frac{\partial h_s}{\partial T} \right)_p , \quad c_{v,s} = \left(\frac{\partial e_s}{\partial T} \right)_V . \tag{1.12}$$

Therefore, the closure of the governing equations requires the modeling of individual species thermodynamic properties e_s, h_s, and $c_{p,s}$ for $s \in S$. Different approaches are available in the literature and are implemented in the Mutation++ library [Scoggins *et al.* 2020]. Mutation++ offers a flexible high-level application programming interface to model the physico-chemical properties of mixtures in

different levels of non-equilibrium. For this reason, it was coupled with our flow solver (see chapter 2) for the thermochemical modeling of hypersonic flows in chemical nonequilbrium.

In this work, the Rigid-Rotor Harmonic oscillator (RRHO) formulation of Mutation++ is used. This model is valid for the simple molecules considered and provide analytical expression for species individual thermodynamic properties. These expressions are derived using the partition function of statistical mechanics [Scoggins & Magin 2014, Scoggins 2017]. Mutation++ also provides the evaluation of the species specific heats, $c_{p,s}$ for $s \in S$ as a function of temperature T and a pressure of 1 bar through the NASA-9 or NASA-7 polynomial databases of McBride et al. [McBride *et al.* 2002]. Finally, temperature T can be recovered by solving the following nonlinear equation for temperature through a Newton-Raphson procedure,

$$h = \sum_{s \in S} h_s^0 + \int_{T^0}^{T} c_{p,s}(T')dT', \tag{1.13}$$

where, h_s^0 is a reference value of the specific enthalpy taken at the reference temperature $T^0 = 298.15$ K.

1.2.1.2 Transport coefficients

Based on the kinetic theory of dilute gases, the mixture transport properties, i.e dynamic viscosity μ, thermal conductivity κ, and the species diffusion velocities $\mathbf{V}_s$, are complex functions of the local thermodynamic state and composition through binary collision integrals [Hirschfelder *et al.* 1964].

A binary collision integral represents integrated collision cross-sections for binary species pairs which interact with one another over the whole range of thermal speed. Their appearence is a direct consequence of the solution of the Boltzmann equation. Indeed, the thorough derivation of the transport fluxes is based on the Chapman-Enskog perturbative solution of the Boltzmann equation [Mitchner & Kruger Jr 1973]. Through a spectral Galerkin projection using Laguerre-Sonine polynomials, transport system can be obtained. Solving the system then provides the transport coefficient. Using the first and second order approximation for dynamic viscosity and thermal conductivity respectively, the transport system reads,

$$\left\{ \sum_{i \in S} G_{s,i}^{\eta} \alpha_i^{\eta} = X_s \right\}, \quad \forall s \in S$$

$$\eta = \sum_{i \in S} \alpha_i^{\eta} X_i. \tag{1.14}$$

$\eta = \{\mu, \kappa\}$ and G^{η} is the respective transport matrix depending on the species mole fractions and binary collision integrals. The derivation of these transport matrices is presented in [Magin & Degrez 2004].

For efficiency reason, the binary collision integrals as function of temperature are stored in Mutation++ through curve-fitted fourth order exponential polynomial [Scoggins & Magin 2014]. Nonetheless, the inversion of the transport system remains computationally expensive, especially as the number of species in the mixture increases. Therefore, a more efficient method to solve this system, using the iterative Conjugate-Gradient (CG) is implemented in Mutation++. This formulation for the evaluation of dynamic viscosity and thermal conductivity is retained in this work.

Alternatively, simplified mixture rules are found in the literature and widely used in CFD of hypersonic flows in chemical nonequilibrium [Di Renzo & Urzay 2021, Passiatore *et al.* 2022], such as those of Gupta and Yos [Gupta *et al.* 1990] and Wilke [Wilke 1950]. These mixture rules combine the weighted individual species transport properties through algebraic equations, thus reducing the computational time of the evaluation of the mixture transport coefficient. However, these mixture rules where shown to have high error in the ionization range [Magin & Degrez 2004].

Hence, the mixture's thermodynamic and transport properties depend, in general, on any two thermodynamic properties and on its composition. The composition is generally considered an independent variable in chemical nonequilibrium, except where it then depends on other thermodynamic quantities for specific cases, as discussed in §1.2.1.4.

1.2.1.3 Finite-rate chemistry and chemical nonequilibrium – CNEQ

In cases where finite-rate chemistry cannot be neglected, the composition is treated as an independent variable. It becomes essential to model the mass production rate of species, denoted as $\dot{\omega}_s$ in Eq. (1.2), as well as the diffusion velocities of species, denoted as $\mathbf{V}_s$. This section provides a description of these terms.

In a general case, a set of reactions, R, is considered depending on the mixture in question. Each reaction, r, is characterized by a reaction rate, R_r, which is computed by the forward rate, k_f, and the backward rate, k_b. These, in turn, are obtained according to experimentally or theoretically calibrated Arrhenius formulas $k_f = C_r T^{n_r} \exp\left(T_{ar}/T\right)$, and $k_b = k_f / K_{eq,r}(T)$, where $K_{eq,r}$ is the reaction equilibrium constant at specific conditions. A generic chemical reaction can be represented by

$$\text{Reaction}(r): \qquad \sum_{s\in S} \nu'_{r,s} S_s \underset{k_b}{\overset{k_f}{\rightleftharpoons}} \sum_{s\in S} \nu''_{r,s} S_s, \tag{1.15}$$

where $\nu'_{r,s}$ and $\nu''_{r,s}$ are the stoichiometric coefficients for reactants and products in reaction r for species s. The net reaction rate is then given by

$$R_r = \left[k_f \Pi_s \left(\frac{\rho_s}{W_s}\right)^{\nu'_{r,s}} - k_b \Pi_s \left(\frac{\rho_s}{W_s}\right)^{\nu''_{r,s}} \right] \cdot \sum_{s\in S} \left(Z_{r,s} \frac{\rho_s}{W_s} \right), \tag{1.16}$$

where W_s is the molar mass of species s and its efficiency as a third-body M in reaction r is denoted by $Z_{r,s}$. The net mass production rates for species s from all

reactions are then computed as

$$\dot{\omega}_s = W_s \sum_{r \in R} \left(\nu''_{r,s} - \nu'_{r,s} \right) R_r. \tag{1.17}$$

Throughout this work, Park's five reactions chemical mechanism for dissociated air [Park 1989] was used. The species set is $S = [\mathrm{N_2}, \mathrm{O_2}, \mathrm{N}, \mathrm{O}, \mathrm{NO}]$, and the five reactions are, with the third-body M being any of the five species considered:

$$\begin{aligned} \mathrm{R1} &: \mathrm{N_2} + \mathrm{M} \rightleftharpoons 2\mathrm{N} + \mathrm{M} \\ \mathrm{R2} &: \mathrm{O_2} + \mathrm{M} \rightleftharpoons 2\mathrm{O} + \mathrm{M} \\ \mathrm{R3} &: \mathrm{NO} + \mathrm{M} \rightleftharpoons \mathrm{N} + \mathrm{O} + \mathrm{M} \\ \mathrm{R4} &: \mathrm{N_2} + \mathrm{O} \rightleftharpoons \mathrm{NO} + \mathrm{M} \\ \mathrm{R5} &: \mathrm{NO} + \mathrm{O} \rightleftharpoons \mathrm{N} + \mathrm{O_2} \end{aligned} \tag{1.18}$$

Local variations of species mass fraction trigger diffusion within the flow, represented in the governing equation by the diffusion flux, $\mathbf{J}_s = \rho_s \mathbf{V}_s$ appearing in Eq. (1.2). Following the description in [Marxen *et al.* 2013], and under the same assumptions, neglecting thermal and barodiffusion, the diffusion driving force for each species, $\mathbf{d}_s$, reduces to its molar fraction gradient,

$$\mathbf{d}_s = \nabla X_s. \tag{1.19}$$

Analogous to the transport coefficients (Eq. 1.14), the diffusion velocity for each species s, $\mathbf{V}_s$, is the solution of the Stefan-Maxwell linear system of equations for a multi-component mixture,

$$\left\{ \sum_{i \in S} G^V_{s,i} \mathbf{V}_i = -\mathbf{d}_s \right\}, \quad \forall\, s \in S \tag{1.20}$$

where G^V is the diffusion transport matrix of the system, which is itself a complex function of binary collision integrals and mole fractions [Magin & Degrez 2004].

While solving the system in Eq. (1.20) is the most accurate way to calculate the diffusion fluxes in a multi-component gas mixture, simpler formulations are found in the literature under the same assumptions [Hirschfelder *et al.* 1964, Ramshaw 1990, Ern & Giovangigli 1994]. A simpler expression based on Fick's diffusion model (applicable to binary mixtures) with a mass correction is given below in Eq. (1.21),

$$\mathbf{J}_s = -cW_s D_s \nabla Y_s + cY_s \sum_{i \in S} W_i D_i \nabla Y_i. \tag{1.21}$$

$c = \sum_{s \in S} (\rho_s / W_s)$ and D_s is the averaged diffusion coefficient for species s, defined

as

$$D_s = \frac{1 - X_s}{\sum_{r \neq s} X_r / D_{s,r}}, \tag{1.22}$$

where $D_{s,r}$ is the binary diffusion coefficient of species s within species r. The binary diffusion coefficient, $D_{s,r}$, is a function of the respective binary collision integral.

Instead of solving a linear system of equations of size N_s, using this model, the diffusion flux is computed from an algebraic equation, at a significantly reduced computational cost. Both implementations are available in the solver presented here, and the accuracy of the simplified approach is verified for practical cases in §3.1.

The constraints seen in Eq. (1.23) have to be respected for the kinetic and diffusive terms, hence

$$\sum_{s \in S} \dot{\omega}_s = 0 \ , \quad \sum_{s \in S} \rho_s \mathbf{V}_s = 0. \tag{1.23}$$

The heat flux takes the original form shown in Eq. (1.7), where the thermal conductivity is replaced by its frozen composition value. The diffusive heat flux is then taken into account explicitly through the second term in Eq. (1.7). The thermochemical library Mutation++ [Scoggins *et al.* 2020] also implement the computation of the chemical reaction and diffusion terms.

1.2.1.4 Special cases – frozen and equilibrium flows

There exist two extreme cases for a multi-component mixture, where its composition becomes irrelevant.

- **Frozen composition**, when the reaction time scale is much longer than the flow time scale (typically quantified by the Damköhler number, $Da \ll 1.0$), hence the composition of the flow is generally constant in time.
- **Local thermodynamic equilibrium - LTE**, when the reaction time scale is short enough for the reactions to bring the mixture to a state of local equilibrium ($Da \gg 1.0$), where the composition approaches its equilibrium composition that can be computed from thermodynamic relations by maximizing the system entropy or, equivalently, minimizing the Gibbs free energy.

In both cases of frozen composition and LTE, the mixture composition becomes a dependent variable, directly computed for a given thermodynamic state. Therefore, the species mass conservation equations, Eq. (1.2), and the corresponding reaction and diffusion terms are not needed. The heat flux takes the simplified form shown in Eq. (1.24),

$$\mathbf{q} = -\frac{\kappa}{Re_\infty Pr_\infty Ec_\infty} \nabla T, \tag{1.24}$$

where the thermal conductivity is computed at either frozen or LTE conditions. In the latter case, the equilibrium value of thermal conductivity is used $\kappa = \kappa_e q$.

In situations between these extremes, finite-rate chemistry needs to be tracked as described in §1.2.1.3 and the mixture composition changes in space and time.

1.2.2 Calorically and thermally perfect gases

When the gas is assumed to be calorically (CPG) or thermally perfect (TPG), the composition becomes irrelevant. The chemical diffusion and reaction terms in the governing equations vanish, and the thermodynamics and transport properties become simple functions of the thermodynamic state of the gas, resulting in a nondimensional equation of state given by

$$p = \rho \frac{\gamma - 1}{\gamma E c_\infty} T. \tag{1.25}$$

It should be noted that this formulation is valid in this form only for the CPG and TPG models.

In the case of a CPG, the thermodynamic properties, such as the specific heat, are considered constant. The internal energy and enthalpy thus become linear functions of temperature, in the nondimensional form $\gamma_\infty e = T$, which is derived from a dimensional expression $\tilde{h} = \tilde{c}_{p,\infty}\tilde{T}$.

In the case of a TPG, the thermodynamic properties are generally functions of temperature. Since the specific heat is now itself a function of temperature, the internal energy and enthalpy become nonlinear functions of temperature. In the solver, T is therefore computed at the end of an iteration from the conservative variable h through an iterative Newton-Raphson procedure. The relations used are based on the work presented in [Malik & Anderson 1991, Marxen *et al.* 2011b]. The specific heat is given as

$$\tilde{c}_p\left(\tilde{T}\right) = \tilde{c}_p^0 \left(1 + \frac{\gamma_\infty - 1}{\gamma_\infty} \left(\frac{\tilde{\Theta}}{\tilde{T}}\right)^2 \frac{e^{\tilde{\Theta}/\tilde{T}}}{e^{\tilde{\Theta}/\tilde{T}} - 1}\right), \tag{1.26}$$

where $\tilde{\Theta} = 3055\text{K}$ and $\tilde{c_p}^0$ such that $\tilde{c}_p\left(\tilde{T}_\infty\right) = \tilde{c}_{p,\infty}$.

The transport properties (viscosity and thermal conductivity) can be constant or follow analytical expressions as functions of temperature. In both the CPG and TPG assumptions, the viscosity is computed using Sutherland's law,

$$\tilde{\mu}\left(\tilde{T}\right) = \tilde{C}_1 \frac{\tilde{T}^{3/2}}{\tilde{T} + \tilde{T}_S}, \tag{1.27}$$

where $\tilde{C}_1 = 1.458 \times 10^{-6}$ kgm^{-1}s^{-1}K$^{-1/2}$ and $\tilde{T}_S = 110.4$ K.

In addition, in both cases, the heat flux takes the simplified form seen in Eq. (1.24) as the diffusive heat flux vanishes. In CPG, the equilibrium conductivity is assumed constant, $\tilde{\kappa} = \tilde{\kappa}_\infty$, while κ can be computed as in Eq. (1.8). Whereas,

in TPG, the equilibrium thermal conductivity is approximated by Keyes' law,

$$\tilde{\kappa}\left(\tilde{T}\right) = \tilde{C}_2 \frac{\tilde{T}^{1/2}}{1 + \left(\tilde{C}_3/\tilde{T}\right) 10^{-\tilde{C}_4/\tilde{T}}}, \tag{1.28}$$

with $\tilde{C}_2 = 2.646 \times 10^{-3}$ $\mathrm{Wm^{-1}K^{-3/2}}$, $\tilde{C}_3 = 245.4$ K, $\tilde{C}_4 = 12$ K.

Chapter 2

Numerical framework

Contents

The following section outlines the numerical framework of the Navier-Stokes solver and the implementation of the models described thus far. The governing equations (Eqs.(1.1-1.4)) are written in vector form as

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

where the state vector $\mathbf{Q}$ contains the conservative variable and $F(\mathbf{Q})$ is the flux vector:

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho_{s_1} \\ \vdots \\ \rho_{s_{n_s}} \\ \rho \mathbf{u} \\ \rho e_0 \end{pmatrix}, \mathbf{F}_C = \begin{pmatrix} \rho \mathbf{u} \\ \rho_{s_1} \mathbf{u} \\ \vdots \\ \rho_{s_{n_s}} \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + p \\ \rho h_0 \mathbf{u} \end{pmatrix}, \mathbf{F}_D = \begin{pmatrix} 0 \\ -\rho_{s_1} \mathbf{V}_{s_1} \\ \vdots \\ -\rho_{s_{n_s}} \mathbf{V}_{s_{n_s}} \\ \tau \\ \tau \cdot \mathbf{u} - \mathbf{q} \end{pmatrix}, \mathbf{S} = \begin{pmatrix} 0 \\ \dot{\omega}_{s_1} \\ \vdots \\ \dot{\omega}_{s_{n_s}} \\ 0 \\ 0 \end{pmatrix} \tag{2.2}$$

2.1 Computational domain and boundary conditions

The equations are integrated numerically (direct numerical simulation - DNS) in a three-dimensional Cartesian coordinate system $\{x, y, z\}$ with the origin placed at the leading edge of the flat plate, and x, y, and z point in the streamwise, wall-normal, and spanwise directions, respectively. A general sketch of the domain of interest is presented in Figure 2.1.

A reference solution is prescribed in the sponge regions, where typically a self-similar boundary layer is used. Please refer to chapter 4 for a complete description

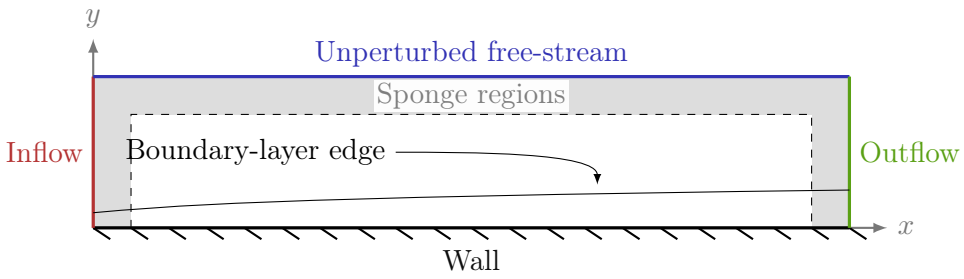


Figure 2.1: Schematic of a generic computational domain. Inflow and outflow can be enforced with reference solutions. The same approach is used at the free stream. The wall is non-catalytic, either isothermal or adiabatic. A variety of forced boundary conditions can be applied locally on the wall.

of the self-similar solutions, using the various gas models defined in §1.2. The solution is forced towards the reference by incorporating damping source terms in the right-hand side of the equations, as seen in Eq. (2.3).

$$\frac{\partial \mathbf{Q}}{\partial t} = F(\mathbf{Q}) - \sigma(\mathbf{x})(\mathbf{Q} - \mathbf{Q}_{ref}) \tag{2.3}$$

The sponge parameter $\sigma(\mathbf{x})$ is a smooth third-order polynomial function in space, vanishing inside the domain and reaching a high value near the boundaries which is selected empirically, such that it damps any reflected waves. The sponge can be completely omitted at the inflow when it is unnecessary, such as in supersonic configurations where upstream-traveling disturbances are minimal. In such cases, a Dirichlet type inflow boundary condition is used. The wall boundary conditions ($y = 0$) is either adiabatic or isothermal, while no slip and no catalysis are permitted for the reacting cases:

$$\mathbf{u} = 0, \quad \frac{\partial Y_s}{\partial y} = 0, \quad T = T_w \quad \text{or} \quad \frac{\partial T}{\partial y} = 0. \tag{2.4}$$

Other boundary conditions, such as blowing and suction at the wall, isolated roughness geometries, or jet injections, are also implemented and available. Finally, periodicity is assumed in the spanwise direction.

2.2 Discretization in space and time

The computational tools developed are based on the original Navier-Stokes solver developed by Nagarajan [Nagarajan *et al.* 2003, Nagarajan 2004]. The code has been applied to DNS and LES studies of flat-plate boundary layers in the subsonic [Sayadi *et al.* 2013] and the supersonic [Marxen *et al.* 2011b, Marxen *et al.* 2013, Marxen *et al.* 2014b, Marxen *et al.* 2014a] regime.

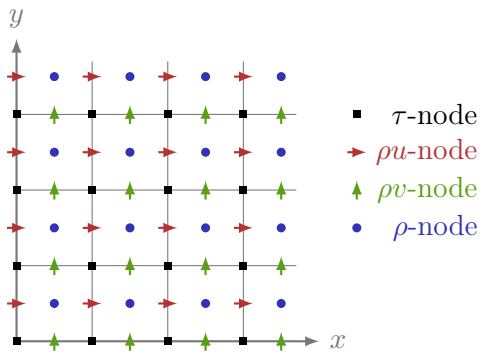


Figure 2.2: Schematic of a staggered grid in two dimensions. The cell-centered ρ nodes are where scalar variables are stored and computed, while the interface nodes ρu and ρv are where the streamwise and wall-normal velocities, fluxes, and gradients are stored and computed. The τ nodes are the grid nodes where the grid is generated, and where the solution is interpolated for post-processing and presentation. Extension to three dimensions is trivial, with an additional ρw node at the interface in the page-normal direction, for the spanwise velocities, fluxes, and gradients.

2.2.1 Spatial discretization

Spatial discretization is done using fourth-order central or sixth-order compact finite differences schemes [Lele 1992] described below. The governing equations are formulated in curvilinear coordinates and solved on a staggered grid, as seen in Figure 2.2. A detailed discussion of the curvilinear transformation is found in [Nagarajan 2004]. This transformation allows to use higher order schemes, derived for uniform cartesian grids, in more complex geometries and non-uniform grids, see §3.1.

2.2.1.1 Sixth order compact finite difference scheme

A family of high order schemes for approximating the first derivative on a uniform staggered grid where the functional value at half a mesh spacing Δx are used is given by [Lele 1992, Nagarajan 2004]

$$\alpha f'_{j-1} + f'_j + \alpha f'_{j+1} = b\frac{f_{j+3/2} - f_{j-3/2}}{4\Delta x} + a\frac{f_{j+1/2} - f_{j-1/2}}{2\Delta x}, \tag{2.5}$$

where f' are the approximate first derivative values. The three free parameters α, a and b are determined by matching the coefficients of the Taylor series to obtain highest order of accuracy, resulting in a one-parameter family of fourth order schemes with

$$\begin{aligned} a &= \frac{3}{8}(3 - 2\alpha), \\ b &= \frac{1}{8}(1 + 22\alpha). \end{aligned} \tag{2.6}$$

Finally, choosing $\alpha = 9/62$ leads to sixth-order accuracy.

The staggered grid arrangement implies the use of a midpoint interpolation. The fourth-order family of interpolation schemes is given by:

$$\alpha f^I_{j-1} + f^I_j + \alpha f^I_{j+1} = \frac{b}{2}(f_{j+3/2} - f_{j-3/2}) + \frac{a}{2}(f_{j+1/2} - f_{j-1/2}) \tag{2.7}$$

where f^I are the interpolated values. The coefficients are,

$$a = \frac{1}{8}(9 + 10\alpha),$$

$$b = \frac{1}{8}(6\alpha - 1), \tag{2.8}$$

and $\alpha = 3/10$ leads to a sixth-order interpolation scheme.

In practice, derivation (or interpolation) with a compact sixth-order schemes involves the resolution of a tridiagonal linear system, defined as follows for a periodic direction:

$$\begin{pmatrix} 1 & \alpha & & & \alpha \\ \alpha & \ddots & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \alpha \\ \alpha & & & \alpha & 1 \end{pmatrix} \begin{pmatrix} f'_1 \\ \vdots \\ f'_j \\ \vdots \\ f'_n \end{pmatrix} = \frac{b}{4\Delta x}\mathbf{f}^3_\Delta + \frac{a}{2\Delta x}\mathbf{f}^1_\Delta \tag{2.9}$$

where $\mathbf{f}^1_\Delta$ and $\mathbf{f}^3_\Delta$ are centered finite difference vectors of the form

$$\mathbf{f}^d_\Delta = \begin{pmatrix} f_{1+d/2} - f_{1-d/2} \\ \vdots \\ f_{j+d/2} - f_{j-d/2} \\ \vdots \\ f_{n+d/2} - f_{n-d/2} \end{pmatrix}. \tag{2.10}$$

2.2.1.2 Boundary schemes

Physical problems of interest commonly involve non-periodic boundaries. To handle this, the system of Eq. (2.9) need to be modified to account for the boundary and near boundary nodes while maintaining a high order of accuracy. Figure 2.3 shows a one-dimensional staggered grid with a boundary condition at the left end ($j = 0$). Due to the staggered arrangement, different boundary schemes have to be used for the boundary ρ or ρu nodes, for instance. These boundary schemes have been derived by [Nagarajan 2004] and involve one-sided finite difference. They are briefly summarized in that section.

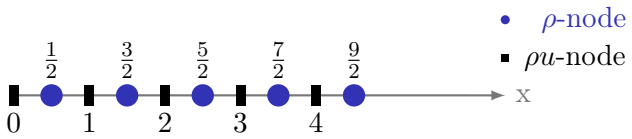


Figure 2.3: Schematic of a one-dimensional staggered grid with a physical boundary at the left end.

Boundary scheme for derivation: The scheme for the first derivative at the boundary ρu-node ($j = 0$) is written as

$$f_0^{'} + \alpha f_1^{'} = \frac{1}{\Delta x}(a f_{1/2} + b f_{3/2} + c f_{5/2} + d f_{7/2}). \tag{2.11}$$

Third order accuracy requires the following relationships to be satisfied:

$$\begin{aligned} a &= -\frac{1}{24}(23\alpha + 71), \\ b &= \frac{1}{8}(7\alpha + 47), \\ c &= \frac{1}{8}(\alpha - 31), \\ d &= \frac{1}{24}(-\alpha + 23), \end{aligned} \tag{2.12}$$

where $\alpha = 0$ is chosen for simplicity. The interior scheme is not applicable to the next ρu node ($j = 1$). The most compact fourth-order scheme is used instead by setting $\alpha = 1/22$, leading to $b = 0$ in Eq. (2.5).

The scheme for the first derivative at the boundary ρ-node ($j = 1/2$) is similarly written as

$$f_{1/2}^{'} + \alpha f_{3/2}^{'} = \frac{1}{\Delta x}(a f_0 + b f_1 + c f_2 + d f_3). \tag{2.13}$$

A third order family of schemes is obtained with the following coefficients

$$\begin{aligned} a &= \frac{1}{24}(\alpha - 23), \\ b &= \frac{1}{8}(-9\alpha + 7), \\ c &= \frac{1}{8}(9\alpha + 1), \\ d &= -\frac{1}{24}(\alpha + 1). \end{aligned} \tag{2.14}$$

Boundary scheme for interpolation: Interpolation at the boundary ρu-node uses the following scheme,

$$f_0^I + \alpha f_1^I = a f_{1/2} + b f_{3/2} + c f_{5/2}. \tag{2.15}$$

A one-parameter, third-order family of schemes is obtained with the following coefficients

$$\begin{aligned} a &= \frac{1}{8}(3\alpha + 15), \\ b &= \frac{1}{4}(3\alpha - 5), \\ c &= \frac{1}{8}(3 - \alpha). \end{aligned} \tag{2.16}$$

A stable scheme is obtained with $\alpha = 0$. For the ρu-node at $j = 1$, the most compact fourth order interpolation scheme is used by imposing $b = 0$ and $\alpha = 1/6$ in Eq. (2.7). The interpolation scheme for the boundary ρ-node ($j = 1/2$) reads

$$f_{1/2}^I + \alpha f_1^{3/2} = a f_0 + b f_1 + c f_2 + d f_3. \tag{2.17}$$

A fourth-order scheme is obtained by setting

$$\begin{aligned} a &= \frac{1}{16}(5 - \alpha), \\ b &= \frac{1}{16}(9\alpha + 15), \\ c &= \frac{1}{16}(9\alpha - 5), \\ d &= -\frac{1}{16}(1 - \alpha), \end{aligned} \tag{2.18}$$

with $\alpha = 0$ for simplicity. Finally, the interpolation scheme for the ρ-node at $j = 3/2$ is chosen to be a fourth order compact scheme with $\alpha = 1/4$ in Eq. (2.7).

2.2.1.3 Fourth order central finite difference scheme

As shown in 2.5.1, the parallel scalability of the sixth order scheme fails after 4096 cores. Indeed, derivation and interpolation of the sixth-order compact scheme requires solving the tridiagonal system in Eq. (2.9), which in turn involves communication to all the processors to build and solve the system. This results in an overhead communication cost, preventing ideal scalability after 4096 cores. Hence, in cases where ideal scalability is needed for greater number of processors (i.e. for 3D simulation), a fourth order central difference schemes can be used instead of the sixth-order compact by setting $\alpha = 0$ in Eqs. (2.5) and (2.6). The resulting scheme

for the first derivative is,

$$f_j' = b\frac{f_{j+3/2} - f_{j-3/2}}{4\Delta x} + a\frac{f_{j+1/2} - f_{j-1/2}}{2\Delta x}, \tag{2.19}$$

and for interpolation,

$$f_j^I = \frac{b}{2}(f_{j+3/2} - f_{j-3/2}) + \frac{a}{2}(f_{j+1/2} - f_{j-1/2}). \tag{2.20}$$

These relations are explicit functions of the functional value of f for the interior points. Accordingly, the boundary schemes are made explicit. Thus, no linear system is solved. This allows for an ideal scalability of the fourth order scheme up to a higher number of computation cores compared to the sixth-order compact scheme, as seen in 2.5.1, Figure 2.5.

2.2.2 Time integration

Time marching is done using explicit Runge-Kutta schemes. To integrate a general equation of the form

$$\frac{dy}{dt} = f(y,t) \tag{2.21}$$

from discrete time t^n to $t^{n+1} = t^n + \Delta t$, with time step Δt, a general Runge-Kutta scheme can be written as,

$$y(t^{n+1}) = y(t^n) + \Delta t \sum_{i=1}^{N_{rk}} b_i k_i, \tag{2.22}$$

where

$$\begin{aligned} k_1 &= f(y_n, t_n), \\ k_2 &= f(y_n + (a_{21}k_1)\Delta t, t_n + c_2\Delta t), \\ k_3 &= f(y_n + (a_{31}k_1 + a_{32}k_2)\Delta t, t_n + c_3\Delta t), \\ &\vdots \\ k_{N_{rk}} &= f(y_n + (a_{N_{rk}1}k_1 + \cdots + a_{N_{rk},N_{rk-1}}k_{N_{rk-1}})\Delta t, t_n + c_{N_{rk}}\Delta t). \end{aligned} \tag{2.23}$$

The number of Runge-Kutta substeps N_{rk}, the coefficients a_{ij} for $1 \leq j < i \leq N_{rk}$, the weights b_i for $i \in [1, 2, 3 \dots N_{rk}]$ and nodes c_i for $i \in [2, 3 \dots N_{rk}]$ define uniquely the scheme. They are usually stored in a Butcher tableau. In the solver, the low-memory two-register RK3 [Nagarajan 2004], the three-register total variation diminishing RK3-TVD [Gottlieb & Shu 1998], and the four-register RK4 schemes [Carpenter & Kennedy 1994, Kennedy *et al.* 2000] are implemented. Their butcher tableau is given in Table 2.1.

The choice of the timestep is limited by the convective stability limit of the

$$\begin{array}{c|ccc} 0 & & & \\ \frac{8}{15} & \frac{8}{15} & & \\ \frac{2}{3} & \frac{1}{4} & \frac{5}{12} & \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array} \qquad \begin{array}{c|ccc} 0 & & & \\ 1 & 1 & & \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & \\ \hline & \frac{1}{6} & \frac{1}{6} & \frac{2}{3} \end{array} \qquad \begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

Table 2.1: Butcher tableau of (left) the low-memory, two-register RK3 scheme, (middle) the RK3-TVD scheme and, (right) the classical RK4 scheme.

scheme, known as the Courant-Friedrichs–Lewy (CFL) condition:

$$\Delta t < \frac{C_O \Delta x}{|\mathbf{u}| + c_s}. \tag{2.24}$$

Where C_O is the Courant number. The CFL limit of each scheme is given in the following table:

Scheme	C_O
RK3	0.551
RK3-TVD	1
RK4	$2\sqrt{2}$

Table 2.2: CFL limit of each Runge-Kutta scheme implemented in the solver.

2.3 Shock-capturing scheme

In the presence of shocks and discontinuities, special treatment is necessary. Indeed, compact finite difference schemes are known to generate spurious numerical oscillations around discontinuities. In this work, we opted for a numerical shock-capturing scheme based on local artificial diffusivity (LAD) in curvilinear coordinates. The scheme, originally developed by Cook and Cabot [Cook & Cabot 2004], adds artificial diffusion near the discontinuities. The method was extended to curvilinear coordinates by Kawai and Lele [Kawai & Lele 2008] and verified for shock capturing in supersonic flow simulations [Kawai & Lele 2008, Kawai & Shankar 2010, Mani *et al.* 2009].

Grid-dependent artificial transport coefficients are added to the physical coefficients appearing in Eqs. (1.6) and 1.7. Namely,

$$\mu_t = \mu + \mu^* \tag{2.25}$$

$$\beta_t = \beta + \beta^* \tag{2.26}$$

$$\kappa_t = \kappa + \kappa^* \tag{2.27}$$

where $(.)^*$ denote the artificial transport coefficients and $(.)_t$ the resulting total coefficient. The artificial coefficients are computed as

$$\mu^* = C_\mu \rho \left| \overline{\sum_{l=1}^{3} \left[\sum_{m=1}^{3} \left(\frac{\partial \xi_l}{\partial x_m} \right)^2 \right]^{r/2} \frac{\partial^r f_\mu}{\partial \xi_l^r} \Delta_l^{r+2}} \right|, \tag{2.28}$$

$$\beta^* = C_\beta D_\beta f_{sw} \rho \left| \overline{\sum_{l=1}^{3} \left[\sum_{m=1}^{3} \left(\frac{\partial \xi_l}{\partial x_m} \right)^2 \right]^{r/2} \frac{\partial^r f_\beta}{\partial \xi_l^r} \Delta_l^{r+2}} \right|, \tag{2.29}$$

$$\kappa^* = C_\kappa \frac{\rho c_s}{T} \left| \overline{\sum_{l=1}^{3} \left[\sum_{m=1}^{3} \left(\frac{\partial \xi_l}{\partial x_m} \right)^2 \right]^{r/2} \frac{\partial^r e}{\partial \xi_l^r} \Delta_l^{r+1}} \right|, \tag{2.30}$$

where $C_\mu = 0.005$, $C_\beta = 3.5$ and $C_\kappa = 0.01$ are dimensionless, user-specified constants. $f_\mu = f_\beta = \nabla \cdot \mathbf{u}$ represents the dilatation term. ξ_l denotes the curvilinear coordinates while x_m refers to the cartesian coordinates, for $\{l,\ m\} \in [1, 2, 3]^2$. Δ_l is the local grid spacing in the physical space normal to the shock, proposed as an improvement to the original method by Mani et al. [Mani *et al.* 2009]. Following the recommendations in [Kawai & Lele 2008, Kawai & Shankar 2010, Mani *et al.* 2009], fourth order derivation, i.e. $r = 4$, is adopted. The fourth derivative are evaluated by sequential application of the first derivative, Eq. (2.5), and interpolation schemes, Eq. (2.7). The switching function f_{sw} in Eq. (2.29) is defined based on [Kawai & Shankar 2010] as

$$f_{sw} = H(-\nabla \cdot \mathbf{u}) \times \frac{(\nabla \cdot \mathbf{u})^2}{(\nabla \cdot \mathbf{u})^2 + |\nabla \times \mathbf{u}|^2 + \varepsilon}. \tag{2.31}$$

This expression is a combination of a heavisyde function H to turn off artificial bulk viscosity when the fluid is expanding (i.e. $\nabla \cdot \mathbf{u} > 0$) and a Ducros sensor [Ducros *et al.* 1999], which helps to localize artificial viscosity only in near-shock regions. Finally, $D_\beta = 1 - exp(\frac{-y^+}{A^+})$, with the Van Driest constant $A^+ = 26$, is a wall damping term to force the artificial viscosity to vanish near the wall.

In the case of finite-rate chemistry simulation, artificial diffusion is also added to all species average diffusion coefficient, defined in Eq. (1.22).

$$D_{s,t} = D_s + D_s^*. \tag{2.32}$$

We use Fiorina and Lele [Fiorina & Lele 2007] entropy-based sensor, extended to curvilinear coordinates. The artificial diffusion coefficient reads,

$$D_s^* = C_Y \frac{c_s}{c_p} \left| \overline{\sum_{l=1}^{3} \left[\sum_{m=1}^{3} \left(\frac{\partial \xi_l}{\partial x_m} \right)^2 \right]^{r/2} \frac{\partial^r s}{\partial \xi_l^r} \Delta_l^{r+1}} \right| \tag{2.33}$$

where s denotes entropy and $C_Y = 0.01$ is the model constant. The advantage of the entropy-based formulation is twofold. First, the artificial coefficient is computed only once for all species, reducing computation cost. Secondly, entropy enables detection of species discontinuities in contact discontinuities or reaction zones not necessarily related to shock waves.

In the definition of various artificial coefficients, the overbar denotes the truncated Gaussian filter defined by Cook and Cabot [Cook & Cabot 2004]. It is applied sequentially to all interior points and in each direction.

$$\overline{f_i} = a_0 f_i + a_1(f_{i-1} + f_{i+1}) + a_2(f_{i-2} + f_{i+2}) + a_3(f_{i-3} + f_{i+3}) + a_4(f_{i-4} + f_{i+4}) \quad (2.34)$$

$$\begin{aligned} a_0 &= \frac{3565}{10368}, \\ a_1 &= \frac{3091}{12960}, \\ a_2 &= \frac{1997}{25920}, \\ a_3 &= \frac{149}{12960}, \\ a_4 &= \frac{107}{103680}. \end{aligned} \quad (2.35)$$

Finally, a low-pass sixth-order compact filter is applied to each conservative variable at the end of each Runge-Kutta iteration.

$$\alpha_f \hat{f}_{i-1} + \hat{f}_i + \alpha_f \hat{f}_{i+1} = a f_i + \frac{b}{2}(f_{i+1} + f_{i-1}) + \frac{c}{2}(f_{i+2} + f_{i-2}) + \frac{d}{2}(f_{i+3} + f_{i-3}) \quad (2.36)$$

f and $\hat{f}$ denote the original and filtered quantities, respectively. The coefficients are set to

$$\begin{aligned} a &= \frac{1}{16}(10\alpha_f + 11), \\ b &= \frac{1}{32}(34\alpha_f + 15), \\ c &= \frac{1}{16}(6\alpha_f - 3), \\ d &= \frac{1}{32}(2\alpha_f - 1), \end{aligned} \quad (2.37)$$

and $\alpha_f \in [-1/2, 1/2]$ is a free parameter. When increased towards $1/2$, the cutoff frequency of the filter is increased. In this work, it is set to $\alpha_f = 0.490$ for all cases needing shock capturing.

2.4 Coupling with the Mutation++ library

In the case of a chemical nonequilibrium simulation, the thermodynamic and transport properties, and the source terms for chemical kinetics, are extracted from the Mutation++ library [Scoggins *et al.* 2020]. The library, written in `C++`, is coupled with the solver, written in `Fortran 95`, using a wrapper interface that facilitates the library function calls by implementing them as functions and subroutines in the solver. An on-the-fly communication between the solver and the library is necessary, with evaluation of the thermochemical and transport properties at each grid point and each iteration, given the local state vector. Quantities are therefore exchanged between the solver and the library for each grid point, at each time step. These function evaluation calls add a significant computational overhead to the solver since each state evaluation involves the iterative solution of nonlinear equations. This is demonstrated below in Figure 2.7. The library offers however valuable modularity, decoupling the thermochemical model from the core of the solver, and offering multiple capabilities to change gas mixtures, databases, and models. It is easily coupled to any computational fluid dynamics solver as an input/output problem $\mathbf{z} = f(\mathbf{x})$. More precisely, given the local state vector

$$\mathbf{x} = [\rho, \rho_s, \rho e] \in \mathbb{R}^D, \tag{2.38}$$

the library returns all physico-chemical properties needed to close the governing equations, as described in §1.2.1.

$$\mathbf{z} = f(\mathbf{x}) = [p, T, \mu, \kappa, D_s, h_s, \dot{\omega}_s] \in \mathbb{R}^{Dz}. \tag{2.39}$$

An overview of the Mutation++ library and its coupling to a CFD solver, taken from [Scoggins *et al.* 2020], is reproduced in Figure 2.4.

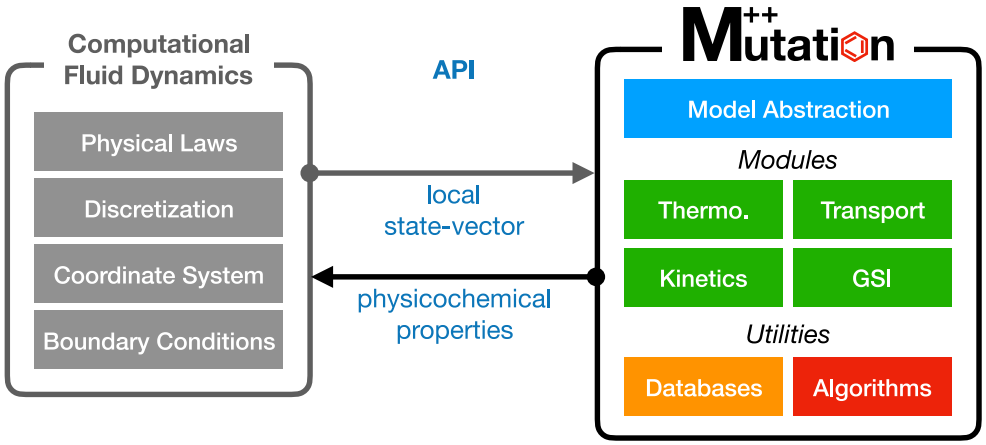


Figure 2.4: Schematic of the Mutation++ library and its coupling to a generic CFD solver. Reproduced from [Scoggins *et al.* 2020].

2.5 Code performance

2.5.1 Parallel scalability

Due to the stringent requirements of direct numerical simulation in terms of grid resolution ($\approx$ billion of degrees of freedom for a typical 3D calculation), the code needs a highly parallel computational framework. Moreover, it should scale well on massively parallel architectures. This is achieved through domain splitting in the streamwise and spanwise direction and an MPI communication framework.

The scalability performance of both the 6^{th} order compact scheme and the more memory efficient 4^{th} order central scheme was tested for a case of a flat plate boundary layer with a mesh of $\approx$ 11 Billion degrees of freedom on up to $\approx$65000 cores of the intrepid BlueGene/P system at the Leadership Computing Facility at Argonne National Laboratory (the details regarding the grid can be found in [Sayadi *et al.* 2013]).

The time required per time step in the code, using both the sixth-order and fourth-order scheme, are shown in Figure 2.5. An ideal scalability would translate to a linear curve, decreasing with the number of computational cores involved. The time required per time step in the code with the compact scheme increases beyond 4096 cores. However, the time per time step continue to decrease linearly in the code utilizing the fourth-order scheme even up to 65536 cores. These results prove the

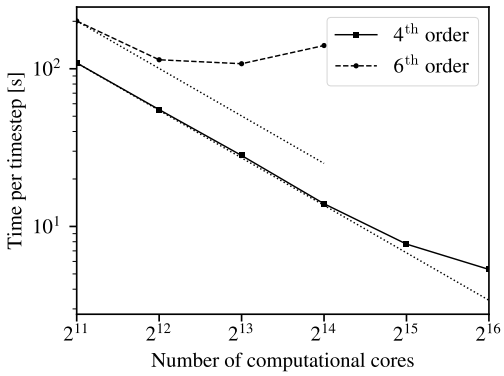


Figure 2.5: Strong scalability (time per iteration) using a grid of $\approx$ 11 Billion degrees of freedom. Fourth-order (solid line), and sixth-order (dashed line) finite difference schemes are compared with ideal scalability (dotted line).

excellent scalability of the code for up to 65536 computational cores using the fourth-order central finite difference scheme. A sixth-order accurate compact scheme scales inefficiently above 4096 cores, due to the overhead cost of solving the tridiagonal system of Eq. (2.9). Moreover, the excellent scalability of the code using the fourth-order scheme has been reassessed on TGCC's Irene-Rome supercomputer through a strong-scaling study using a mesh of $\approx$ 1.5 billion grid points (see §3.3.2 for details regarding this grid), on up to 16384 cores. As shown in Figure 2.6, the time per timestep of the solver decreases linearly with the number of processors used up to

16384 cores.

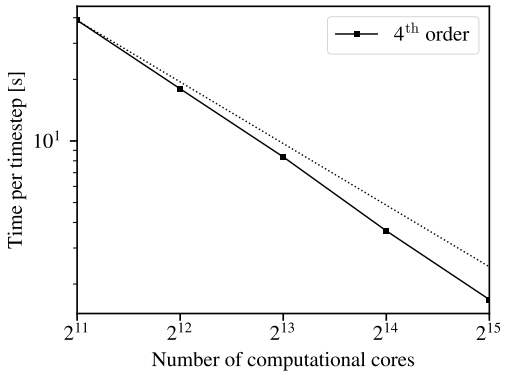


Figure 2.6: Strong scalability (time per iteration) using a grid of ≈ 1.5 Billion degrees of freedom. Fourth-order (solid line) finite difference scheme is compared with ideal scalability (dotted line).

2.5.2 Comparison between perfect-gas and real-gas models

Benchmark simulations have been run to compare the computational cost of the various thermochemical models implemented in the solver. The normalized time to solution, with respect to the simple CPG model, are presented in Figure 2.7. The flexibility and extended validity of using Mutation++ come with a large computational cost. The total simulation time is increased by more than one order of magnitude when using the CNEQ model compared to the CPG model, and by a factor of about six compared to the TPG model. A reduction of about 40% in computational cost can be achieved using Ramshaw's algebraic diffusion model, as discussed in §1.2.1.3, compared to the Stefan-Maxwell diffusion model. The impact of the simplified diffusion model on the accuracy of the solution in chemical nonequilibrium will be investigated in §3.1.

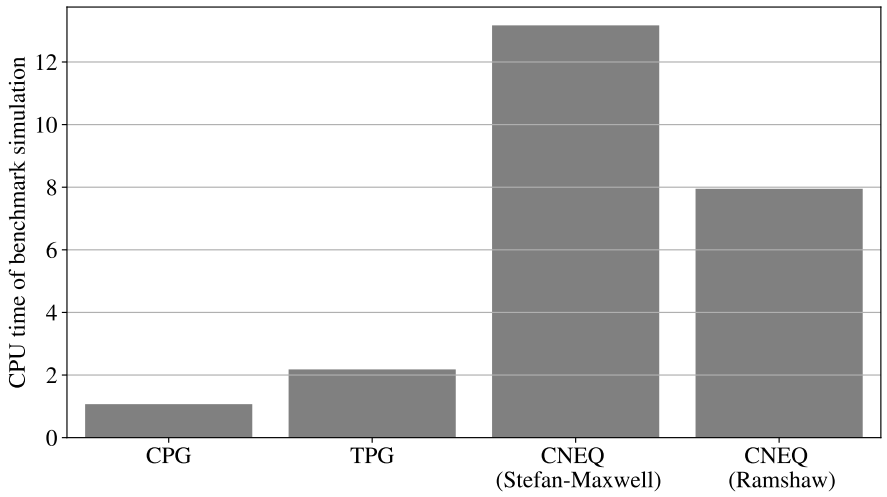


Figure 2.7: CPU time of a benchmark simulation, run with different aerothermochemical models. Including non-equilibrium effects in the simulation causes a significant increase in computational time.

Chapter 3

Numerical simulation of hypersonic flows

Contents

This chapter presents numerical simulations of several canonical hypersonic flows in earth atmosphere. The impact of the thermochemical model on such flows will also be studied. First, the numerical implementation is validated with a set of two $Ma = 10$ boundary layers with different wall boundary conditions and a $Ma = 2$ shock-wave boundary layer interaction (SBLI). The solver is then used to study more complex configurations such as roughness, SBLI at higher Mach number and jet in high-speed crossflow (JISC). In all cases, the CNEQ model uses a five component air mixture composed of N_2, O_2, N, O, and NO, interacting based on Park's chemical mechanism [Park 1989] for dissociated air (see §1.2.1.3).

3.1 Boundary layers with finite rate chemistry effects

A set of two hypersonic boundary layers in Earth's atmosphere at $Ma = 10$, based on Marxen et al. [Marxen *et al.* 2013, Marxen *et al.* 2014b], are considered here to verify the implementation of the numerical method presented in chapter 2. First, the steady-state solution and then the growth rate of forced secondary instabilities inside the two-dimensional boundary layer are compared. The main configurations

considered are: (i) case I (isothermal), and (ii) case A (adiabatic), referring to the boundary condition imposed at the wall. A third case, (iii) case R (roughness), is then designed by including a two dimensional roughness element at the wall using adiabatic conditions. The configurations and freestream conditions for all three cases are presented in Table 3.1, where ω and A are related to the forced perturbations and further explained in §3.1.1.

	Isothermal Case (I)	Adiabatic Cases (A, R)
Ma_∞	10.0	
Re_∞	10^5	
$T_\infty[K]$	278	350
$p_\infty[Pa]$	4135	3596
T_{wall}/T_∞	4.31	-
ω	45	34
$A/Ma\infty$	10^{-3}	

Table 3.1: Thermodynamic and freestream conditions for the $Ma = 10$ hypersonic boundary layers investigated in this study (adapted from [Marxen *et al.* 2013]).

The computational domain considered here has a finer resolution than in the previous studies. The nondimensional grid size in the streamwise direction is $\Delta x = 0.075$, simulating a domain from $x_0 = 14.0$ to $x_1 = 86.0$ using $N_x = 960$ grid points. The sponge regions extend for 5 and 15 nondimensional units at the inflow and outflow, respectively. In the wall-normal direction, $N_y = 211$ grid points are used, clustered near the wall using Eq. (3.1), with the last 26 points in the free stream used in the sponge region. In Eq. (3.1), y_0 and y_1 represent the minimum and maximum wall-normal coordinates, respectively, and κ_y the stretching factor. Table 3.2 summarizes the details of the computational domain as well as the resulting resolution. This resolution is demonstrated sufficient to properly resolve the dynamics through a grid convergence analysis (performed for case I) where finer and coarser grid are also investigated (see Table 3.2). The results are shown below in Figure 3.3.

$$y(m) = y_0 + (y_1 - y_0)\left((1-\kappa_y)\left(\frac{m-1}{N_y-1}\right)^3 + \kappa_y\left(\frac{m-1}{N_y-1}\right)\right),\ m \in [1, N_y] \quad (3.1)$$

All three thermochemical models, namely TPG, LTE and CNEQ, are systematically tested in both cases.

3.1.1 Disturbance forcing

A single frequency, two-dimensional disturbance is introduced in the baseflow at the wall through a blowing-suction strip extending from $x_s = 19.3$ to $x_e = 20.7$, centered at $x_c = 20$. This disturbance can be classified as a second-mode instability according to [Malik & Anderson 1991], which is most amplified in compressible

Case	x_0	x_1	y_0	y_1	κ_x	κ_y	N_x	N_y
Case A	14.0	85.0	0	1.6	1.0	0.15	960	211
Case I - baseline	14.0	85.0	0	1.6	1.0	0.15	960	211
Case I - fine	14.0	65.0	0	1.7	1.0	0.15	1920	301
Case I - coarse	14.0	70.0	0	1.5	1.0	0.15	768	151

Table 3.2: Mesh configuration for the hypersonic boundary layer simulations

boundary layer flows [Mack 1975, Mack 1984]. This case was initially studied in [Malik & Anderson 1991] using linear stability theory. The author predicted a most amplified non-dimensional frequency of $\omega = 3.4 \times 10^{-5}$. This case was later revisited by Marxen *et al.* [Marxen *et al.* 2011b, Marxen *et al.* 2013] using direct numerical simulations and different thermochemical model (perfect gas, chemical equilibrium and nonequilibrium), where good agreement for the growth-rate and amplitude functions were found at $R_x = \sqrt{Re_x} = 2000$ compared to the linear stability theory. The analysis was then further extended to weakly nonlinear stages [Marxen *et al.* 2014a].

The disturbance has a nondimensional forcing frequency ω given as,

$$\omega = 2\pi \tilde{f} \frac{\mu_\infty}{\rho_\infty u_\infty^2}. \tag{3.2}$$

Here, $\tilde{f}$ is the dimensional frequency of the disturbance. The amplitude of the velocity perturbation A is defined as a fraction of the freestream velocity u_∞. The freestream conditions, reference scales, forcing frequency, and amplitude are summarized in Table 3.1. The velocity boundary conditions are defined as in [Marxen *et al.* 2011b, Marxen *et al.* 2013]

$$\begin{aligned} v(x,0,t) &= A sin(\omega t) s(\xi), \\ u(x,0,t) &= 0. \end{aligned} \tag{3.3}$$

The shape function s is defined within the strip as

$$s(\xi) = 18.1875\xi^5 - 35.4375\xi^4 + 20.25\xi^3, \tag{3.4}$$

where the auxiliary coordinate ξ is defined as

$$\xi = \begin{cases} (x - x_s)/(x_c - x_s) & \text{for } x_s < x < x_c, \\ (x - x_e)/(x_e - x_c) & \text{for } x_c < x < x_e, \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

After the introduction of the disturbance, the wall temperature distribution is held constant (isothermal boundary condition) for case A. The unsteady simulation is then advanced until transient effects are advected out of the domain and a time-periodic state is achieved.

3.1.2 Post-processing

In order to compare the dynamics of the unsteady simulations, N_T flow snapshots are collected over one forcing period, and the root-mean-squared (RMS) decomposition of the flow field is performed for a given primitive quantity $\phi \in [\rho, u, v, w, P, T, \ldots]$ as

$$\overline{\phi} = \frac{1}{N_T} \sum_{i=1}^{N_T} \phi_i, \tag{3.6}$$

$$\phi_{RMS} = \sqrt{\frac{1}{N_T} \sum_{i=1}^{N_T} (\phi_i - \overline{\phi})^2}. \tag{3.7}$$

In the following, the RMS wall-pressure signals will be compared in addition to the streamwise disturbance amplification. The latter is computed in two ways: i) using the wall-normal maximum of the RMS wall-normal velocity,

$$E^1_{RMS}(x) = \max_y (v_{RMS}(x, y)). \tag{3.8}$$

ii) or using the wall-normal integral of RMS wall-normal velocity squared,

$$E^2_{RMS}(x) = \int_0^{y_{max}} v_{RMS}(x, y)^2 dy. \tag{3.9}$$

Finally, the normalized growth rate α_i is computed using

$$\alpha_i(x) = \frac{1}{E^j_{RMS}(x)} \frac{\partial E^j_{RMS}(x)}{\partial x}, \tag{3.10}$$

for $j \in [1, 2]$. The comparison of the growth rates calculated from the current simulations and the previously published results serves as a validation of the accuracy of the solver. Indeed, the spatial derivative in Eq. (3.10) makes the growth rate highly sensitive to any changes in the flow dynamics.

3.1.3 results

3.1.3.1 Isothermal case – case I

The mean streamwise velocity and temperature profiles at the streamwise location where $R_x = 2000$ for case I are presented in Figure 3.1(a,b), compared with the results of [Marxen *et al.* 2013], where only the TPG model was investigated. This figure shows perfect agreement between previous and current results for the TPG profiles. By additionally investigating the LTE and CNEQ models in the present study, we discover minor differences compared to the TPG solution. A straightforward interpretation of this finding is that the isothermal condition leads to a maximum temperature of $T_{max} \approx 1740K$ in the boundary layer. Below 2000K air chemistry is known to be generally inactive [Anderson 2019] with almost negligible

dissociation. This is supported in Figure 3.1(c) where the species mass fraction profiles are plotted and no radical species (O, NO, N) is found in significant quantities throughout the boundary layer. Hence, the thermally perfect gas assumption holds reasonably well, and more complex models reproduce a similar behavior.

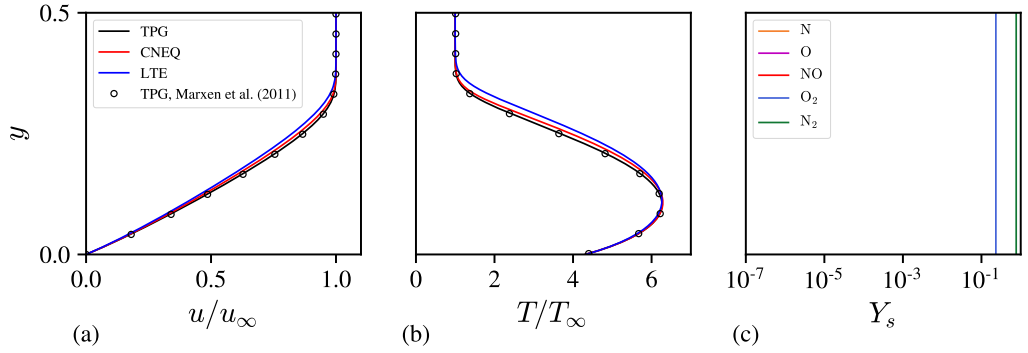


Figure 3.1: (a) Streamwise velocity, u, (b) temperature, T, and (c) species mass fraction, Y_s, profiles in the boundary layer for case I at $R_x = 2000$. Current results (solid lines) are compared to Marxen et al. [Marxen *et al.* 2013] (symbols) for different gas models: TPG (black), LTE (blue), CNEQ (red). (c) From left to right : O_2, N_2.

Figure 3.2(a) shows the base-flow solution for the streamwise velocity and a snapshot of the evolution of the forced perturbation is presented in Figure 3.2(b). Figure 3.2(c,d) shows the corresponding wall-pressure RMS signal and the growth rate, α_i, of the second-mode instability in the boundary layer, respectively. Growth rates are computed using Eq.(3.8) and are not plotted for $x < 25$ due to their erratic behavior in that region. Excellent agreement is found downstream in the investigated range (from $x = 25$ to $x = 50$) with the results of [Marxen *et al.* 2013] for the TPG model. The grid convergence analysis results are shown in Figure 3.3. Increasing the resolution in both directions does not lead to a significant difference in the instability growth rate compared to the baseline grid. Moreover, the current simulations are better resolved than those of [Marxen *et al.* 2013], and the improved resolution has been found to explain the more accurate prediction and the disappearance of small oscillations upstream. Interestingly, despite the CNEQ base flow being almost identical to that of the TPG model, the instability grows earlier in the boundary layer and to a higher amplitude compared to the case modeled using a TPG assumption. This is also conveyed in Figure 3.2(c) where the RMS pressure peaks upstream and to a slightly higher value in the CNEQ case. Thus, we find that finite-rate chemistry alters the growth and decay of the perturbations directly, even though it has a negligible effect on the base-flow solution, potentially altering the stability and transition behavior without an obvious effect on the base flow. These results are in agreement with previous studies.

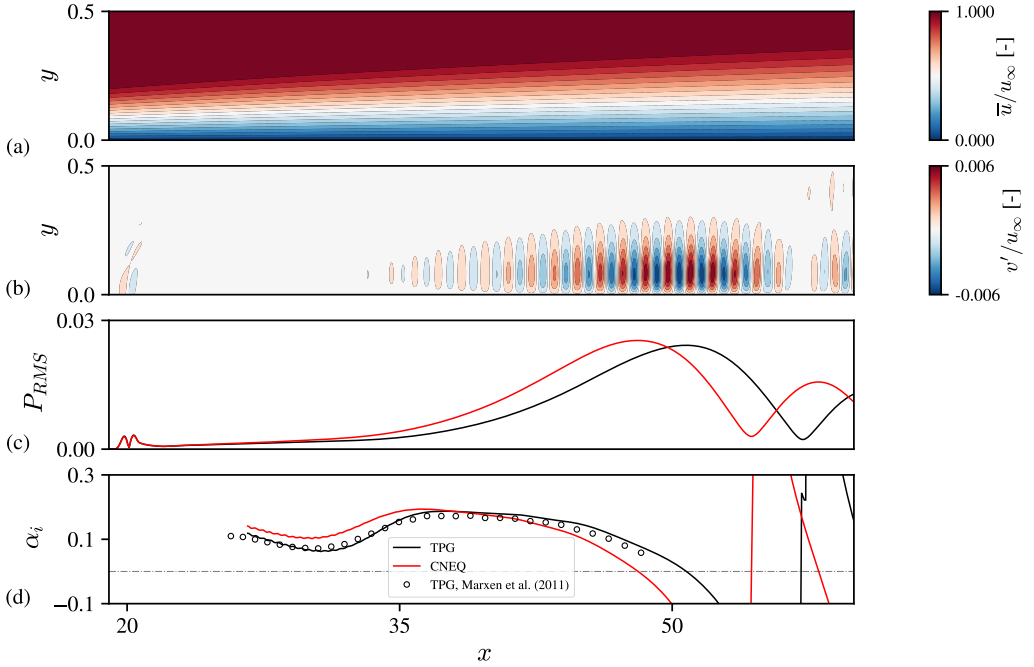


Figure 3.2: Contours of normalized (a) mean streamwise velocity $\overline{u}$, and (b) wall-normal perturbation velocity v' in the laminar isothermal Mach-10 boundary layer in chemical nonequilibrium (Case I, TPG). Evolution of (c) RMS wall pressure, and (d) growth rate α_i (computed using Eq.(3.8)) as a function of streamwise position x for different gas models: TPG (black), CNEQ (red). (d) Current results (solid lines) are compared to Marxen et al. [Marxen *et al.* 2013] (symbols).

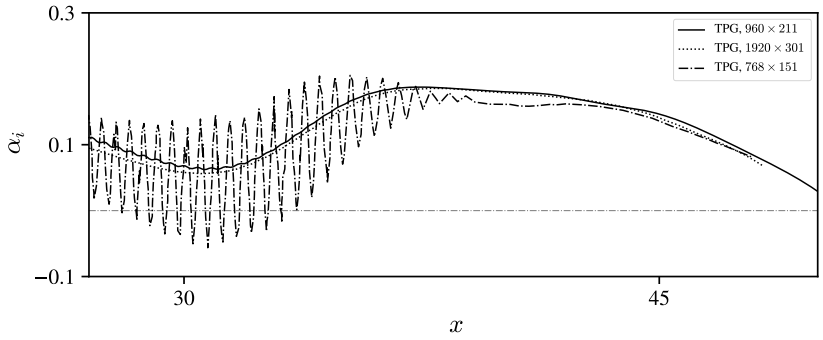


Figure 3.3: Evolution of growth rate α_i (computed using Eq. (3.8)) as a function of streamwise position x for different grid resolution: coarse (dash-dotted), baseline (solid), fine (dotted).

3.1.3.2 Adiabatic case – case A

In the adiabatic case, a higher free-stream temperature is imposed by design to promote CNEQ effects.

The mean flow and temperature profiles at a streamwise Reynolds number location $R_x = 2000$ are presented in Figure 3.4(a,b) and compared to the results of Marxen [Marxen *et al.* 2013]. Similar to the isothermal case, practically perfect agreement is found for all models compared to the previous results. However, in this case, the base flows differ significantly depending on the model used for the gas. Wall temperature decreases significantly from a frozen-chemistry (fixed-composition) assumption to a CNEQ model to an LTE model. These results indicate that the CNEQ effects in the flow are significant in this case and need to be accounted for to achieve accurate base-flow predictions.

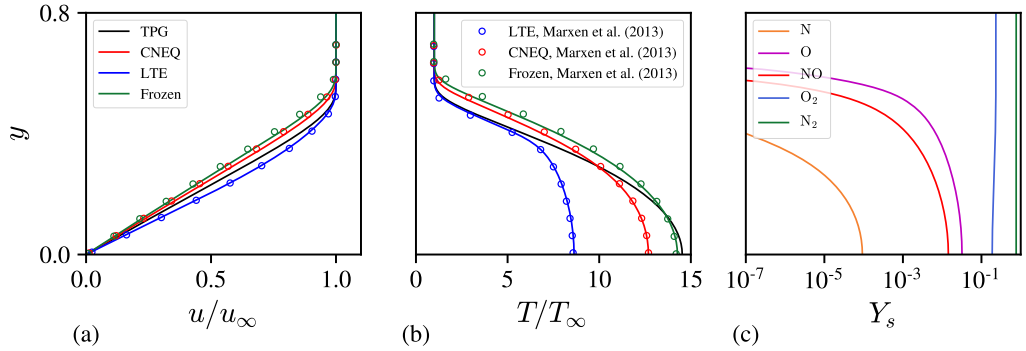


Figure 3.4: (a) Streamwise velocity, u, (b) temperature, T, and (c) species mass fraction, Y_s, profiles in the boundary layer for case A at $R_x = 2000$. Current results (solid lines) are compared to Marxen et al. [Marxen *et al.* 2013] (symbols) for different gas models: TPG (black), LTE (blue), CNEQ (red), frozen chemistry (green). (c) From left to right : $\mathrm{N}, \mathrm{NO}, \mathrm{O}, \mathrm{O}_2, \mathrm{N}_2$

Due to the near-wall temperature approaching $T_{wall} \approx 4900K$ at the inflow, N_2 and O_2 rapidly start to dissociate to $\mathrm{N}, \mathrm{O}, \mathrm{NO}$ through endothermic chemical reactions. This is illustrated in Figure 3.4(c), presenting all the mass fraction profiles at the streamwise location where $R_x = 2000$. Close to the wall, O_2 mass fraction decreases while O and NO are produced. To a smaller extent, N is also created through N_2 dissociation. Moreover, the dissociated species concentrations $(\mathrm{O}, \mathrm{NO}, \mathrm{N})$ build up as the species are convected downstream while continuously being produced. Consequently, the wall temperature decreases progressively along the streamwise direction due to cooling by endothermic dissociation.

In Figure 3.5(a), we see the base-flow solution for the streamwise velocity, and a snapshot of the evolution of the forced perturbation in Figure 3.5(b). Figure 3.5(c,d) shows the corresponding RMS wall pressure signals and growth rates in the adiabatic case, respectively. Good agreement is found for the CNEQ model up to $x = 40$ ($R_x = 2000$). Downstream, the computed growth rate differs slightly from the previous results but the overall trend is similar. This difference can be explained by the older thermochemical model implemented in Mutation++ at the time of the original study of Marxel et al. [Marxen *et al.* 2013]. The growth rate in the TPG model is however noticeably different compared to the cases using the CNEQ model,

highlighting the inadequacy of perfect gas models in high-enthalpy cases.

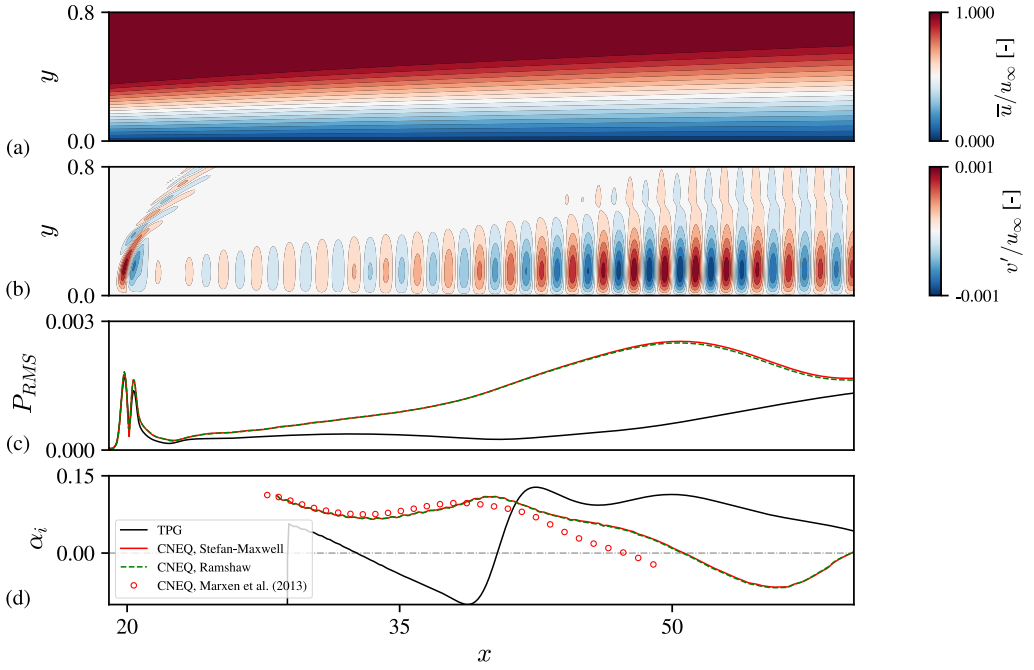


Figure 3.5: Contours of normalized (a) mean streamwise velocity $\overline{u}$, and (b) wall-normal perturbation velocity v' in the laminar adiabatic Mach-10 boundary layer in chemical nonequilibrium (Case A, CNEQ). Evolution of (c) RMS wall pressure, and (d) growth rate α_i (computed using Eq.(3.8)) as a function of streamwise position x for different gas models: TPG (black), CNEQ with Stefan-Maxwell multicomponent diffusion model (red), CNEQ with Ramshaw simplified multicomponent diffusion model (green). (d) Current results (solid lines) are compared to Marxen et al. [Marxen *et al.* 2013] (symbols).

Ramshaw's simplified algebraic diffusion model (see §1.2.1.3) has a negligible impact on the accuracy of the results. In fact, the base flow and growth rates of second mode instability for case A were computed again using Eq. (1.21). Base flows are practically identical in terms of all relevant variables. As shown in Figure 3.5(d), the growth rate is in perfect agreement with that computed using the Stefan-Maxwell model. A slight discrepancy in the RMS pressure signals after the peak at $x = 50$ is observed, which can be attributed to small changes in the diffusion fluxes. Therefore, Eq. (1.21) is a good compromise between performance and physical accuracy as it reduces the cost of a CNEQ simulation by 40% compared to the Stefan-Maxwell multicomponent diffusion model, see Figure 2.7. The same model was also used in various numerical studies of hypersonic boundary layers with finite-rate chemistry [Passiatore *et al.* 2021, Di Renzo *et al.* 2020, Di Renzo & Urzay 2021]. In the following, the CNEQ model will systematically be used with Ramshaw's simplified algebraic diffusion model if not otherwise specified.

3.1.3.3 Roughness case – case R

Early experiments [Van Driest & Blumer 1968] have shown that the inclusion of roughness elements in supersonic/hypersonic boundary layers can influence transition location. Marxen et al. [Marxen *et al.* 2010] studied the influence of a two dimensional roughness element on a $Ma = 4.8$ supersonic boundary layer via direct numerical simulations. The roughness element was shown to boost the linear growth mechanism in the boundary layer in a narrow band of frequency while waves outside that range were damped. More recent investigations [Duan *et al.* 2013, Heitmann & Radespiel 2013] qualitatively confirmed these previous results. Few studies, however, took into account the impact of a more detailed thermochemical model on the flowfield, a necessity in the hypersonic regime. Stemmer et al. [Stemmer *et al.* 2017] studied the steady flow around a three-dimensional roughness element in a $Ma = 6.3$ boundary layer and considered TPG, LTE and CNEQ models. The nonequilibrium case was shown to decrease the wall temperature in the wake of the roughness (in agreement with the results of case A) but increased the heat transfer at the wall. Hence, these results underline further the need for nonequilibrium simulations. In light of these findings, studying linar instability mechanism in a roughness case, including chemical nonequilibrium effects, would be of interest to the community.

To that end, a two-dimensional roughness element is added in the previous flat-plate geometry (case A) with the following analytical expression (taken from [Marxen *et al.* 2010]):

$$y_{wall}(x) = \frac{h_R}{2} \sum_{k=-1}^{1} k \tan(s_R(x - x_{0,k})), \text{ with } x_{0,k} = x_{c,R} - k l_R/2, \tag{3.11}$$

where h_R, l_R and $x_{c,R}$ denote height, length and centered location of the roughness, respectively. The parameter s_R controls the smoothness of the roughness element. The roughness geometry is compiled in Table 3.3 and the modified mesh is presented in Figure 3.6.

h_R	0.2
l_R	1
$x_{c,R}$	30
s_R	5

Table 3.3: Roughness geometry parameters (case R).

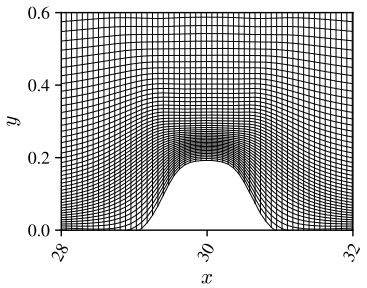


Figure 3.6: Modified grid in the vicinity of the roughness element (case R).

The freestream and thermodynamic quantities are kept the same as in case A to facilitate the comparison. The flow deviation caused by the roughness induces a recirculation bubble, a separation shock, an expansion fan and a compression wave, as seen in Figure 3.7(a) where the mean density field is presented.

The flow is forced with the same two dimensional disturbance described in Table 3.1. However, the amplitude of the disturbance is increased from $A = 1 \times 10^{-3}$ to $A = 2 \times 10^{-2}$. Case A was simulated again with the higher amplitude and both TPG and CNEQ thermochemical models. No significant changes in the instability growth pattern are observed for the simulation in CNEQ, as shown in Figure 3.8. However, at higher disturbance amplitude, the second location of positive amplification is slightly shifted downstream for TPG simulations. The overall trend remains nonetheless similar.

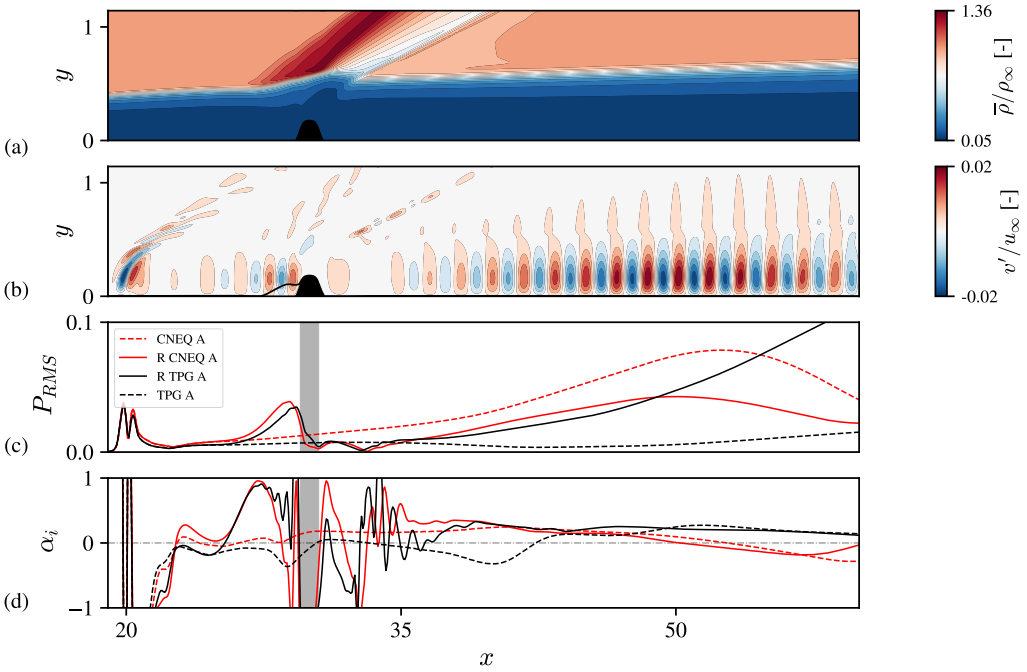


Figure 3.7: Contours of normalized (a) mean density $\overline{\rho}$, and (b) wall-normal perturbation velocity v' in the laminar adiabatic Mach-10 boundary layer in chemical nonequilibrium with roughness (Case R, CNEQ). Evolution of (c) RMS wall pressure, and (d) growth rate α_i (computed using Eq. (3.9)) as a function of streamwise position x for different gas models: TPG (black), CNEQ (red).

The data acquisition and post-processing steps are done as described in 3.1.2. The growth-rate is computed using Eq. (3.9). Contours of instantaneous wall-normal velocity perturbations are presented in Figure 3.7(b). The presence of the roughness strongly affects the development of the instability in its vicinity. Further downstream, the perturbation evolution reverts back to that of a flat-plate behavior. These observations are supported in Figure 3.7(c) where the RMS wall-pressure signals of the perturbed boundary layer with and without roughness

are presented. The wall-pressure signal is strongly modified near the roughness element (grey vertical band). Further downstream, after the end of the recirculation bubble on the leeward side of the roughness, both curves exhibit the same trend. However, the flow with a roughness-induced shock and expansion shows lower amplitude of the pressure perturbations (only 50% of the flat plate value at the peak near $x = 55$). This can be explained in light of the results of Marxen *et al.* [Marxen *et al.* 2010, Marxen *et al.* 2014b], where a similar analysis was performed in a compressible Mach 4.8 boundary layer. From a linear stability theory point of view, two-dimensional roughness elements were shown to act as amplifiers to two-dimensional disturbances in a narrow band of frequency while shifting the most amplified frequency to lower values. Here, the forcing frequency was selected based on the flat-plate case. Thus, if the roughness induces a shift of the most amplified frequency to lower values, the expected growth of the two-dimensional disturbance will be weaker, which is what is observed in Figure 3.7(c). The same observation can be made by looking at the growth-rate of the wall-normal velocity perturbation in Figure 3.7(d). The growth rate of the flat-plate geometry (red dashed line) is juxtaposed with the one of the roughness geometry (red solid line). After $x = 40$, both growth rates follow the same trajectory, with lower values for the roughness case. The roughness acts as a band-pass filter on the most amplified frequency of the flat-plate case due to its shift to lower frequency. Additionally, the same case has been run with a perfect gas assumption ((R TPG A), black lines in Figure 3.7(c,d)). The pressure fluctuations are of the same order of magnitude but the instability grows over a longer distance downstream of the roughness and ultimately reaches higher amplitudes. The growth rate becomes negative at $x = 65$, compared to $x = 50$ for the chemical nonequilibrium case. Hence, in a perfect gas framework, the roughness has an inverse effect and amplifies the instability growth directly downstream. This stresses again the importance of the thermochemical model for the accurate simulation of hypersonic flows.

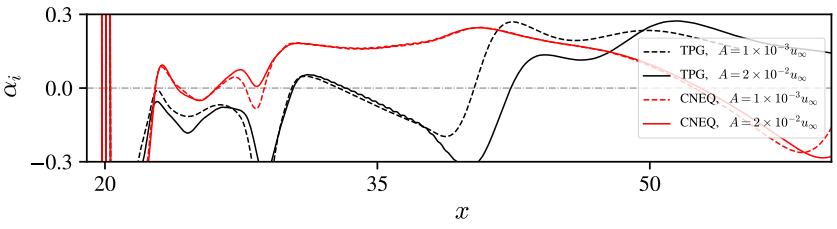


Figure 3.8: Evolution of growth rate α_i (computed using Eq. (3.9)) as a function of streamwise position x for different amplitude of perturbation: $A = 1 \times 10^{-3}$ (dashed), $A = 2 \times 10^{-2}$ (solid) and different gas model: TPG (black), CNEQ (red).

3.2 Shock wave boundary layer interaction

Supersonic and hypersonic flows over complex geometries usually involve shock wave boundary layer interaction (SBLI). The large pressure gradient induced by the impinging shock may cause the separation of the boundary layer with the occurrence of a recirculation bubble. This bubble can in turn change the stability characteristics of the flow on the vehicle surface. In the following section, we first validate the capability of the code to accurately simulate SBLI against a benchmark $Ma = 2$ shockwave laminar boundary layer experiment by Hakkinen [Hakkinen *et al.* 1959]. Then, a higher Mach number case is designed, based on [Hildebrand *et al.* 2018], with a high free-stream temperature, to directly assess the effect of finite-rate chemistry compared to a perfect gas assumption. The reference length is based on the displacement thickness δ_0^* of the unperturbed boundary layer at the inviscid shock location x_0.

SBLI Case	$x_0 - x_1$	$y_0 - y_1$	κ_x	κ_y	N_x	N_y
$Ma_\infty = 2.0$	$40 - 240$	$0.0 - 80$	1.0	0.15	960	501
$Ma_\infty = 5.92$	$19 - 254$	$0.0 - 36$	1.0	0.15	960	450

Table 3.4: Mesh configuration for the two SBLI cases investigated.

3.2.1 A validation case – $Ma = 2$ SBLI

The capability of the solver to correctly simulate compressible flows including shocks is verified using a benchmark SBLI case, first investigated experimentally by [Hakkinen *et al.* 1959] and later numerically in several studies [Katzer 1989, Morgan *et al.* 2010]. In this case, a laminar $Ma = 2$ boundary layer over an adiabatic plate is separated by an impinging shock with a shock angle of $\theta \approx 32°$. The Reynolds number based on the impinging location of the shock is $Re_{x_0} = 3 \cdot 10^5$. All other flow conditions match the simulation by [Katzer 1989]. In the top-sponge opposite of the wall, Rankine-Hugoniot oblique shock relations are used to propagate the oblique shock downward, progressively introducing the discontinuity into the domain. The grid used in that case is finer than in the previous studies, using respectively 960 and 501 points in the streamwise and wall-normal directions. All grid parameters are specified in table 3.4. Due to the relatively low Mach number, only a TPG simulation is performed for this configuration.

Figure 3.9(a) shows the density field after convergence of the residuals to machine precision. All relevant flow features of the SBLI are present: the recirculation bubble, the separation shock, the expansion fan, and the reflected shock. The results compare qualitatively well to those of [Morgan *et al.* 2010]. Figure 3.9(b,c) presents the wall pressure and skin friction coefficient along the wall. The quantitative agreement with the results of Morgan et al. [Morgan *et al.* 2010] is almost perfect.

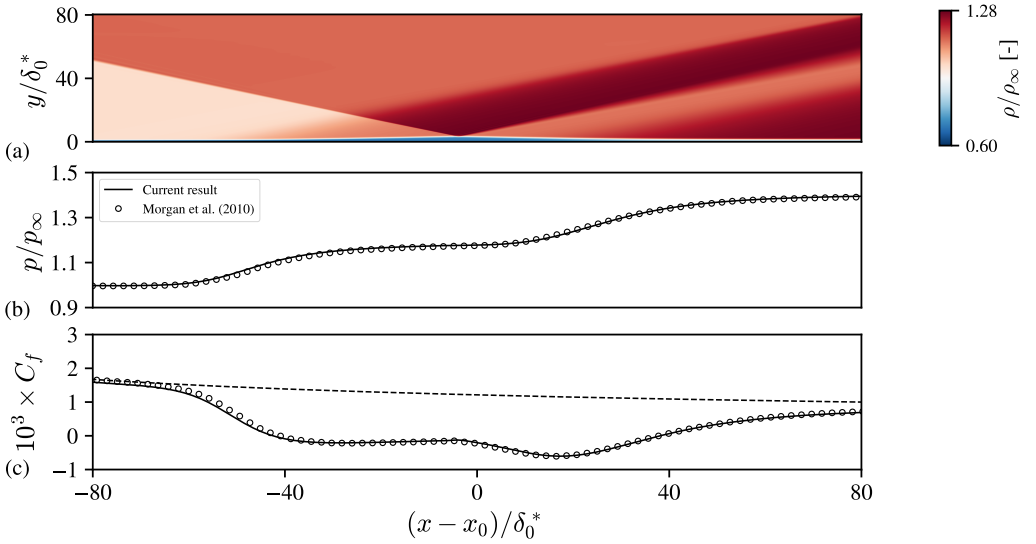


Figure 3.9: (a) Density contour for the $Ma = 2$ laminar SBLI case. The impinging and reflected shockwaves are visible, as well as the recirculation bubble and other relevant flow features. (b) Wall pressure and, (c) skin friction coefficient C_f streamwise distribution. The inviscid shock impingement location is marked with x_0, and coincides with the location of zero on the x-axis. Current results (solid black line) are compared to Morgan et al. [Morgan *et al.* 2010] (symbols).

3.2.2 Chemical nonequilibrium effects in laminar SBLI – $Ma = 5.92$ SBLI

Furumoto et al. [Furumoto *et al.* 1997] studied the real-gas effects on a steady oblique SBLI at $Ma = 7$. The high enthalpy at the free stream leads to Oxygen and Nitrogen dissociation and a reduction of the size of the recirculation region as well as peak heating on the surface due to the endothermic real-gas effects. However, the authors note that the thermochemical model used was rather simplistic. More recently, Volpiani [Volpiani 2021] studied an oblique SBLI at $Ma = 6$ in chemical non-equilibrium with both laminar and turbulent inflow boundary layers. These results highlight the same trend, with a smaller recirculation bubble and higher skin friction at the wall when thermochemistry is considered. However, in this case, the thermochemical model was also simplistic compared to the one included in Mutation++.

In this section, we propose to study the effects of finite-rate chemistry on a steady SBLI at $Ma = 5.92$. A similar case was studied numerically by Hildebrand et al. [Hildebrand *et al.* 2018] where the authors used freestream cryogenic conditions of the ACE Hypersonic Wind Tunnel facility at Texas A&M University [Semper *et al.* 2012], and hence a perfect gas assumption. Here, the freestream Mach number and Reynolds number at impinging location, Re_{x_0}, are kept the same as in [Hildebrand *et al.* 2018], while the free-stream pressure and temperature are increased to match the post-shock conditions of a 15° wedge flying at $Ma_\infty = 14$

at an altitude of 25 kilometers to promote real-gas effects. The computational domain is a rectangle of size 256×36 reference length units. The reference length is computed from the inflow Reynolds number $Re_{\delta^*} = 9660$ of the original study [Hildebrand *et al.* 2018], with the updated free-stream conditions, $p_\infty = 60967$ Pa and $T_\infty = 1110.5$K, resulting in a similar impinging location and corresponding Reynolds number, $Re_{x_0} = 1.15 \cdot 10^6$. A total of 960 grid points are used in the streamwise direction and 450 in the wall-normal direction, clustered near the wall using eq. (3.1). The mesh configuration is specified in Table 3.4. For this case, both the TPG and CNEQ models are investigated.

Figure 3.10(a) shows the density field of the CNEQ simulation. Again, all the relevant flow features of the SBLI are present: the recirculation bubble, the separation shock, the expansion fan, and the reflected shock. Figure 3.10(b) shows the concentration of atomic oxygen Y_O. While inexistant in the freestream, O is present in large quantities within the boundary layer and the recirculation bubble. The wall pressure and skin friction coefficient for both the TPG and CNEQ models are presented in Figure 3.10(c,d). The length of the separation bubble is smaller when considering CNEQ effects, in agreement with the literature. This is expected with the high concentration of dissociated species in the bubble as seen in Figure 3.10(a,d). O_2 almost completely dissociates within the recirculation bubble to form its atomic counterpart O. This endothermic process extracts energy out of the recirculation bubble, making the interaction weaker. Just upstream of the reattachment location, the flow exhibits higher skin friction and wall pressure in the CNEQ case before converging to the same value as the TPG case after reattachment. These trends are in agreement with the results in [Volpiani 2021].

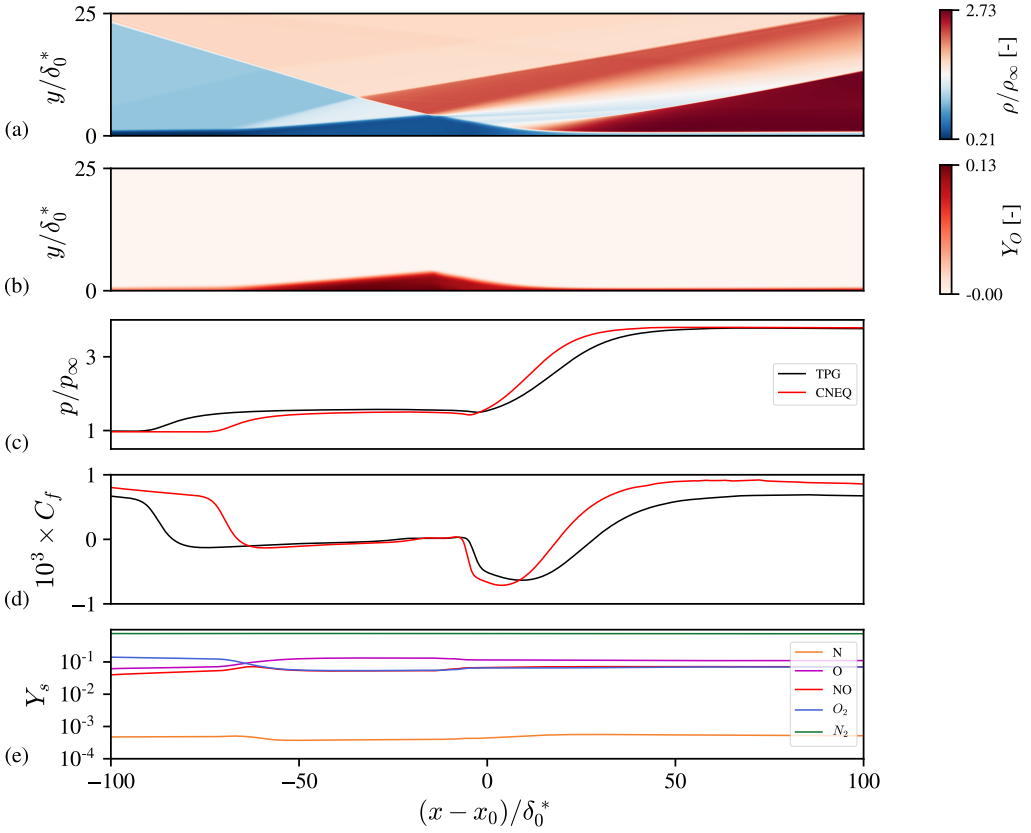


Figure 3.10: Contours of (a) normalized density and, (b) atomic oxygen mass fraction Y_O for the $Ma = 5.92$ laminar SBLI case in CNEQ. The impinging and reflected shockwaves are visible, as well as the recirculation bubble and other relevant flow features. The recirculation bubble contains high proportion of dissociated oxygen. (c) Wall pressure and, (d) skin friction coefficient C_f streamwise distribution for different gas models: TPG (black), CNEQ (red). The inviscid shock impingement location is marked with x_0, and coincides with the location of zero on the x-axis.

3.3 Jet in hypersonic crossflow

Normal jet injection into a high-speed crossflow (JISC) is another canonical flow configuration relevant to a wide range of applications in hypersonic flight. For example, proposed designs of scramjet engines use sonic under-expanded jet injection into a supersonic crossflow to enhance fuel and oxidizer mixing and sustain supersonic combustion. One can also use jet injection into a supersonic/hypersonic crossflow as a reaction control thruster for aerodynamic maneuvering during atmospheric flight [Grandhi & Roy 2017]. The jet in supersonic crossflow has also been the subject of a wide range of experimental [Erdem 2011, Santiago & Dutton 1997, Ben-Yakar *et al.* 2006] and numerical studies [Chai *et al.* 2015, Kawai & Lele 2010, Peterson *et al.* 2006, Miller *et al.* 2018].

The nondimensional parameter that governs the development of flow features is the jet to crossflow momentum ratio J [Karagozian 2014], given as

$$J = \frac{\rho_j U_j^2}{\rho_\infty U_\infty^2},$$

and the density ratio S when compressibility effects are important [Nair *et al.* 2019],

$$S = \frac{\rho_j}{\rho_\infty}.$$

3.3.1 A preliminary two dimensional case – $Ma = 5$ 2D JISC

The analysis of finite-rate chemistry effects on the JISC set-up is performed for a two-dimensional flow. In this case, the jet is injected through a rectangular slot, where the spanwise extent is assumed far larger than its streamwise extent. Two-dimensional JISC have been extensively studied by Spaid and Zukoski [Spaid & Zukoski 1968] for a range of inflow Mach number and momentum ratio.

The two-dimensional case is designed by considering a 15° wedge flying at $Ma = 9$ at an altitude of 25 kilometers. This configuration results in a $Ma = 5$ boundary layer past the nose shock. A self-similar solution with free-stream values assigned to post-shock conditions and an adiabatic wall results in $T_{wall} \approx 4000$ Kelvin (K). This temperature is sufficiently hot to observe intensified dissociation of O_2 into atomic Oxygen and thus can result in a flow in CNEQ. For this case, the CNEQ and TPG models are considered. The free-stream and jet thermodynamic conditions are summarized in table 3.5, where subscript j denotes the jet quantities. The computational domain is a rectangle of size $32.5D \times 10D$ where $\tilde{D} = 2$mm is the jet slot width. In the streamwise direction, the inflow starts at $x_0 = 38.5$ and the injection zone is centered at $x = 52.5$. The inflow and outflow sponges extend for 1 and 2 jet widths, respectively. In the wall-normal direction, grid points are clustered near the wall using Eq. (3.1), and the last 20 points are used for the sponge. The discretization for this case is specified in Table 3.6. The simulation is first initialized with an adiabatic condition and the resulting wall temperature is held constant after the jet injection is enforced.

Freestream				Jet			
Ma_∞	p_∞[Pa]	T_∞[K]	Re_D	Ma_j	p_j/p_∞	T_j/T_∞	J
5	49800	947	25291	1.0	10.0	1.0	0.4

Table 3.5: Thermodynamic conditions for the hot 2D JISC.

Figure 3.11(a) shows a numerical Schlieren image (contours of the norm of the density gradient) for the jet in crossflow simulation in CNEQ. Several characteristic flow features are observed. The jet has its own shock structure with a barrel shock and a Mach disk. It mainly acts as an obstruction in the flow. As a result, a large

Case	$x_0 - x_1$	$y_0 - y_1$	κ_x	κ_y	N_x	N_y
JISC	$38.5 - 71.0$	$0.0 - 10.0$	1.0	0.15	1625	500

Table 3.6: Mesh configuration for the hot 2D JISC.

bow-shock forms in the freestream that divert the incoming flow. The induced large pressure gradient causes the upstream boundary layer to separate upstream of the injection zone. The thickening of the boundary layer in turn induces a separation shock that interacts with the bow shock. Finally, a shear layer emanates from the high-speed jet and the low-speed flow past the bow shock, promoting vortex shedding through the Kelvin-Helmoltz instability. This results in a strong coupling between the shock structures, the recirculation bubble, and the shear layer downstream of the injection zone.

Looking at the pressure distribution at the wall, Figure 3.11(c), a steep pressure rise due to separation is first encountered. The pressure then plateaus with 3 spikes, corresponding to primary, secondary and tertiary upstream vortices, respectively. These vortical structures can be associated to the three 'pockets' seen in Figure 3.11(a,b). Downstream of the jet, the pressure drop corresponds to the downstream vortex structures, followed by a hump due to reattachment of the boundary layer and recompression shock. Therefore, the structure observed are qualitatively similar as the one described in [Spaid & Zukoski 1968]. Similar flow patterns were observed for a $Ma = 16$ axisymmetric shock-dominated hypersonic laminar separated flow over a double cone, studied by Tumuklu et al. [Tumuklu *et al.* 2018]. In their study, a supersonic underexpanded jet is generated through an Edney type IV pattern in the shock-laminar separation bubble. The jet is inherently unsteady and becomes the root of the SBLI instability.

Concerning finite-rate chemistry effects, areas of high concentration of O are observed inside the recirculation bubble in Figure 3.11b. In this high-temperature region, atomic Oxygen is produced through the endothermic O_2 dissociation. This reaction absorbs energy from the flow and reduces the size of the recirculation bubble by 10% compared to the TPG simulation in Figure 3.11(c). This observation holds for the duration of the instantaneous snapshots considered and is similar to the trend observed in the steady $Ma = 5.92$ SBLI comparison between the TPG and CNEQ models. When CNEQ effects are considered, the adiabatic wall temperature decreases due to endothermic reactions (by about 250K at the wall) and the boundary layer becomes slightly thicker. The induced cooling effect near the wall leads to a smaller recirculation bubble and a weaker bow shock.

3.3.2 Three-dimensional jet in cold hypersonic crossflow – $Ma = 5$ 3D JISC

In most engineering applications of interest, the jet is injected through a circular-shaped nozzle. Flow features are fully three-dimensional, with the formation of a

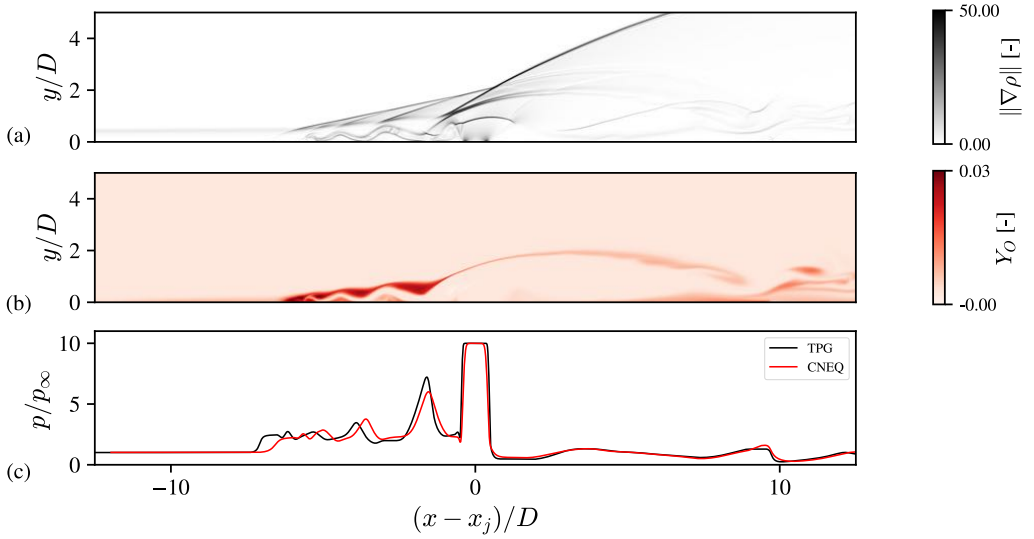


Figure 3.11: Instantaneous (a) numerical Schlieren and (b) atomic Oxygen mass fraction fields for the 2D $Ma_\infty = 5$ JISC. Jet location is at $x/D = 52.5$. The shock structure, shear layer and recirculation bubble dynamics are highlighted in the numerical Schlieren (a). (c) Wall pressure streamwise distribution for different gas models: TPG (black), CNEQ (red).

bow-shock which bends in the spanwise direction. In the upstream recirculation bubble, the vortical structures wrap around the injection zone and form horseshoe vortices that are convected downstream, until eventually breaking down. This induces significant flow in the spanwise direction and is known as the three-dimensional relieving effect. As a consequence, the size of the recirculation bubble is reduced compared to the two-dimensional set-up. The formation of vortices in the jet shear layer above the Mach disk can be attributed to the Kelvin-Helmholtz instability, arising from the velocity difference between the slower fluid behind the bow shock and the high speed jet. From a time-averaged point of view, two counter-rotating vortices are observed above the Mach disk and are responsible for downstream mixing of freestream with the injected gas.

High-fidelity numerical and experimental data of JISC in the hypersonic regime are extremely scarce. In fact, most studies are done in the supersonic regime [Chai *et al.* 2015, Kawai & Lele 2010, Peterson *et al.* 2006]. Erdem perfomed experimental studies of $Ma = 5$ JISC with different injectant gas and momentum ratio [Erdem 2011]. Recently, Miller et al. [Miller *et al.* 2018] performed a large-eddy simulation (LES) of one of the cases studied by Erdem to shed light on the unsteady and transient mechanism developing during the injection. It is therefore of interest to the community to have access to a high-fidelity database of a hypersonic jet in crossflow case, backed by experimental data. These intricate and complex features also serve as a perfect playground to identify coherent flow structures in the flow using data-driven techniques.

To this end, a three-dimensional case is designed based on the experiment of

Erdem [Erdem 2011]. The freestream temperature is set at $T_\infty = 62.5$K. The low freestream temperature permits the use of the Thermally Perfect Gas (TPG) model. The freestream and jet conditions are summarized in Table 3.7.

Freestream				Jet			
Ma_∞	p_∞[Pa]	T_∞[K]	Re[m^{-1}]	Ma_{jet}	p_j/p_∞	T_j/T_∞	J
5	1210	62.5	13.1×10^6	1	29	4	1.16

Table 3.7: Freestream and thermodynamic conditions for the cold 3D JISC (From [Erdem 2011]).

3.3.2.1 Computational set-up

The computational domain corresponds to a box of size $54D \times 9D \times 20D$, where $D = 2.2$ mm, is the circular injector diameter. The box starts at 25 jet diameter downstream of the leading edge of the flat plate and the jet inlet is placed at $x/D = 52.5,\ y/D = 0,\ z/D = 10$. At the inflow and outflow, the thickness of the sponge layer is set to 2 jet diameters.

In order to set the resolution needed to perform the DNS simulations, viscous units are used. Using classical turbulent boundary layer theory, the minimum grid spacing at the wall is $\Delta^+ = 2.847[\mu\text{m}]$. Following the DNS calculations of a turbulent boundary layer performed by [Sayadi *et al.* 2013], a uniform grid spacing of 12 viscous units is set for the spanwise direction. In the streamwise direction, however, knowing that the upcoming boundary layer is laminar, the grid spacing is initially set to 195 viscous units at the inlet and is then gradually refined up to $x/D = -3.75$ where $\Delta x^+ = 15$ with a 5^{th} order version of Eq. (3.1). Based on the experimental results, $x/D = -3.75$ is inside the recirculation zone of the separation bubble ahead of the jet inlet, where laminar flow assumptions still hold. Therefore, the larger grid size will not influence the development of the jet downstream and does not violate the DNS assumptions. The streamwise grid spacing then remains uniform at $\Delta x^+ = 15$. In the wall-normal direction, the stretching of Eq. (3.1) is applied and the last 30 points are used for the sponge layer. The resolution requirements described leads to a total number of points of ≈ 1.5 billion.

The computational domain and the resulting 1.5 billion points grid used to capture the small-scale structures of the flow appearing in the leeward side of the jet are summarized in Table 3.8.

x_0	x_1	y_0	y_1	z_0	z_1	κ_x	κ_y	Δ_x^+	Δ_y^+	Δ_z^+	N_x	N_y	N_z
25	79	0	9	0	20	0.40	0.12	15	1	12	2048	697	1024

Table 3.8: Mesh configuration for the cold 3D JISC.

The simulation was performed on TGCC's Irene Rome supercomputer under

allocation No. 2021-A0102B12426 and No. 2022-A0122B13432 made by GENCI, using 8192 computational cores. After advection of the transient, time resolved slices and subdomain were saved every 500 iterations for a time period of $D/c_\infty = 2$ flow-trough time, corresponding to a sampling Strouhal number of $S_t = f_s D/u_\infty = 15$. The simulation used roughly 8 million core hours.

3.3.2.2 Validation

Current time-averaged results of the simulation are compared to the experimental results reported by Erdem [Erdem 2011], which are the only available validation data. In Figure 3.12(a-c), streamwise pressure distribution at three different spanwise locations are compared. In terms of both magnitude and location of the peaks, excellent agreement is found at $z/D = 10$ (midplane) and $z/D = 4.77$. At $z/D = 9.10$, the pressure peak is slighly shifted upstream in our simulation. This can be an effect of the spanwise extent of the domain and the periodic boundary condition. Additionally, Erdem reported the measured Mach disk height and separation length. The Mach disk height is within 10% of the experimental value, and actually closer to the theoretical prediction based on a formula by Cassel [Cassel 2003]. Good agreement is also observed for the separation length, as seen in Table 3.9?

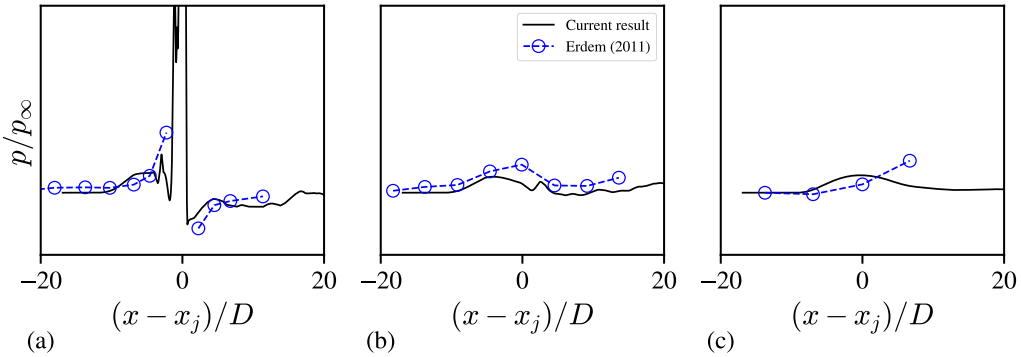


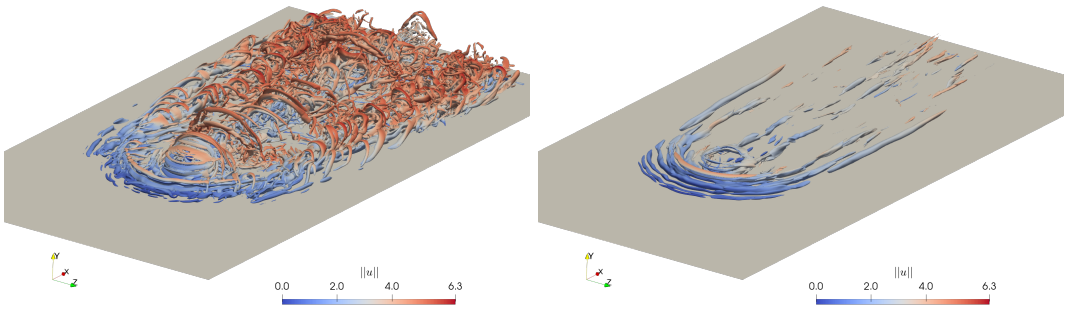
Figure 3.12: Comparison of the time averaged streamwise pressure distribution for the 3D JISC case with the experiment of Erdem [Erdem 2011] at different spanwise location: (a) $z/D = 10$ (centerline), (b) $z/D = 4.77$, and (c) $z/D = 9.10$.

	h_j/D	x_{sep}
Erdem (2011)	1.9 ± 0.075	7.95 ± 0.15
Current results	1.72	8.4
Theory	1.78	

Table 3.9: Comparison of Mach disk height and separation length of the time-averaged flow and of the experiments [Erdem 2011].

3.3.2.3 Flow structures

Figure 3.13(a,b) shows iso-surfaces of the Q-criterion [Jeong & Hussain 1995] for an instantaneous snapshot and the time-averaged flowfield, indicating the vortical structures in the flow. Horseshoe vortices are highlighted. They form in the recirculation bubble upstream of the injection zone and bend aroung the jet. In Figure 3.13(b), the mean vortex breaks down downstream, corresponding to the apparition of smaller structures in the instantaneous flowfield. Close to the jet centerline, vortices are periodically shed due to the shear-layer instability between the high-speed jet and the low-speed flow past the bow shock. Closer to the wall, on the leeward side of the jet, wake vortices are also noticeable on both instantaneous and time-averaged visualizations.



(a) **(b)**

Figure 3.13: Isosurfaces of the Q-criterion as an indicator of vortical structures in (a) the instantaneous flow field, and (b) the average flow, colored by the magnitude of the velocity vector.

Figure 3.14(a-f) shows a numerical Schlieren in the midplane $z/D = 10$ for the time-averaged flowfield and instantaneous snapshots. On the time-averaged flow (Figure 3.14(a)), the same shock structures as those in the two-dimensional case are easily recognizable, Figure 3.11(a). The jet expands, forming a barrel shock with a Mach disk on top. The bow shock now has a 3D structure as it bends in the spanwise direction. A separation bubble and separation shock are still present in front of the jet injection. However, they are smaller than in the 2D case, even at a higher jet-to-crossflow momentum effect, due to the 3D relieving effect. Looking at the series of instantaneous Schlieren, Figure 3.14(b-f), the vortex shedding induces a whip-like movement on the bow-shock as well as the jet shock structure. This coupling of the shock structures with the shear layer likely alter the upstream pressure gradient and in turn, the recirculation bubble.

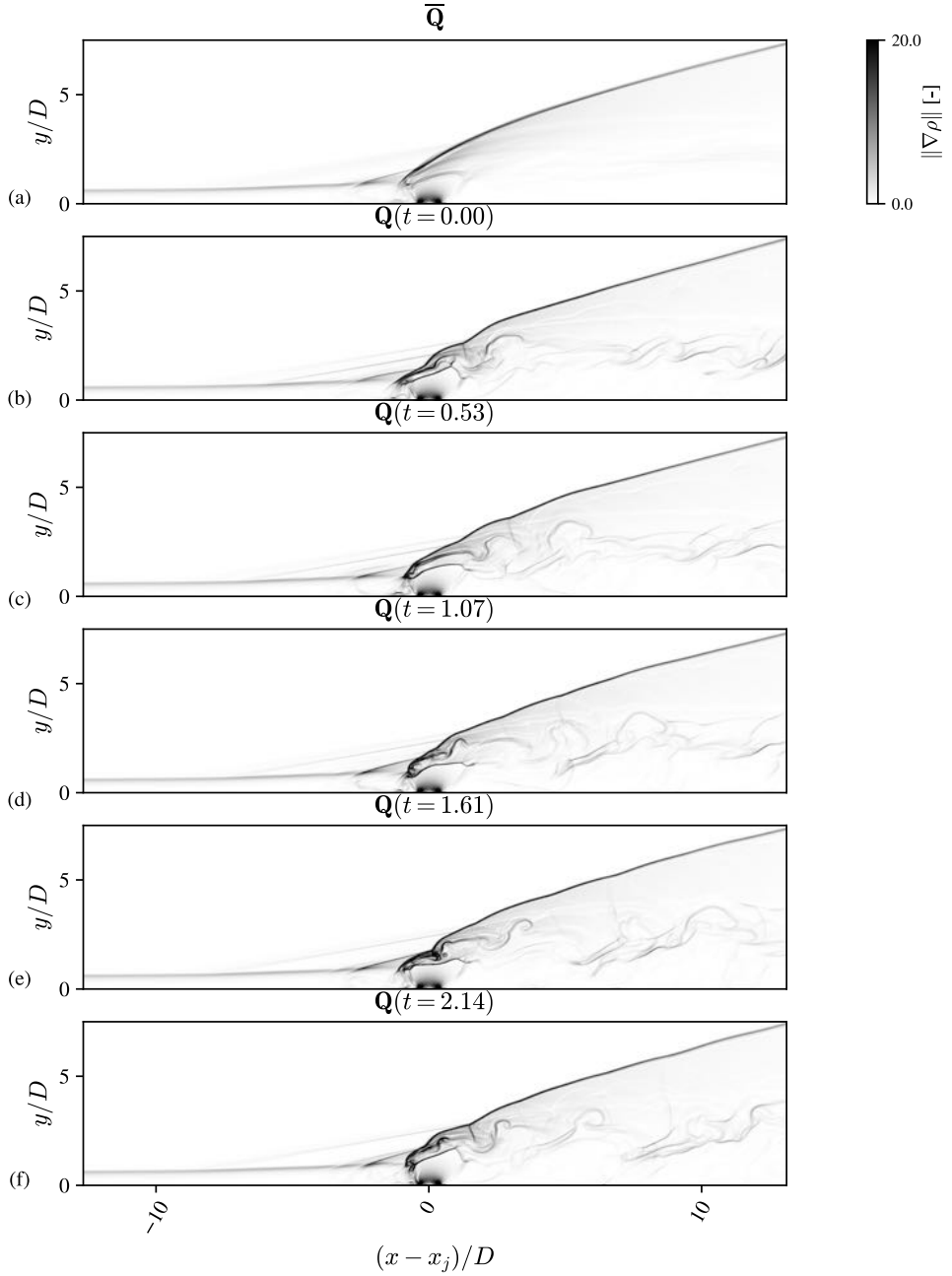


Figure 3.14: Numerical Schlieren of (a) time-averaged flow, and (b-f) instantaneous snapshot. The jet injection coincides with the location of zero on the x-axis. The shock structure, shear layer and recirculation bubble dynamics are highlighted in the numerical Schlieren.

3.3.3 Modal decomposition of coherent flow structures in the $Ma = 5$ 3D JISC

For design purposes, it is important to understand the unsteady characteristics of the JISC configuration [Miller *et al.* 2018]. In fact, the way these instabilities interact with the meanflow and with each other alter both mean flow characteristics, such as drag and thrust, and unsteady components such as mixing, heating and local pressure. In our case, Figures 3.13 and 3.14 highlight how complex the JISC flowfield is, exhibiting different temporal and spatial features.

3.3.3.1 Proper Orthogonal Decomposition

To decipher the most important unsteady flow features and their associated physical mechanism, a common strategy is to extract dominant modes from the dataset [Taira *et al.* 2017]. In fluid dynamics, Proper Orthogonal Decomposition (POD), introduced by Lumley [Lumley 1967], extracts modes by optimizing the mean square of the field variables considered. We assume that the flowfield $\mathbf{Q}(\mathbf{x}, t)$ can be decomposed in the following manner

$$\mathbf{Q}(\mathbf{x}, t) - \overline{\mathbf{Q}}(\mathbf{x}) = \sum_i a_i(t)\mathbf{\Phi}_I(\mathbf{x}). \tag{3.12}$$

Where $\overline{\mathbf{Q}}$ represent the temporal mean, $a_i(t)$ and $\mathbf{\Phi}_j(\mathbf{x})$ are the time coefficients and spatial modes, respectively. The goal of POD is to find the optimal basis of modes $\mathbf{\Phi}_i,\ i \in [1, r]$, in the least-square sense, that best represent the flow data. Let us consider the data matrix $\mathbf{X}$, defined as,

$$\mathbf{X} = [\mathbf{Q}(t_1)\ \mathbf{Q}(t_2)\ \ldots\ \mathbf{Q}(t_m)] \in \mathbb{R}^{n\times m}, \tag{3.13}$$

where m flow snapshots $\mathbf{Q}(t_j)$, at corresponding time $t_j,\ j \in [1, m]$, are horizontally stacked. n is usually equal to the grid size times the number of variables considered. A robust technique to find the relevant modes is to perform the singular value decomposition (SVD) of the data matrix:

$$\mathbf{X} = \mathbf{\Phi\Sigma\Psi}^T, \tag{3.14}$$

where $\mathbf{\Phi} \in \mathbb{R}^{n\times n}$ and $\mathbf{\Psi} \in \mathbb{R}^{n\times m}$ represent the left and right singular vector, respectively. Due to the relation between eigenvalue decomposition of the correlation matrix $\mathbf{XX}^T$ and SVD [Taira *et al.* 2017], the singular value contained in $\mathbf{\Sigma} \in \mathbb{R}^{n\times n}$ follows $\sigma_j^2 = \lambda_j$, and $\mathbf{\Phi}$ contains the POD modes.

$$\mathbf{\Phi} = [\mathbf{\Phi}_1\ \mathbf{\Phi}_2\ \ldots\ \mathbf{\Phi}_n] \tag{3.15}$$

The time coefficient of mode i at time t_j can then be recovered by projecting the original data onto the corresponding mode as

$$a_i^j = \mathbf{X}_j\mathbf{\Phi}_i^T \tag{3.16}$$

For incompressible flows, if the velocity field **u** is used in the snapshot matrix, the POD modes are optimal, in the L_2 sense, to best represent the kinetic energy of the flow with the minimal number of modes. In fact, in incompressible flows, only the kinematic variables are dynamically important. In our case, due to compressibility, both kinematic and thermodynamic variables contribute to the energy of the flow. Hence, to obtain POD modes that best represent the total energy of compressible flow, the Chu norm is commonly used instead [Chu 1965, Hanifi *et al.* 1996].

3.3.3.2 Application to the shear layer instability

The analysis on the shear layer instability is based on a collection of $m = 160$ flow snapshots of the midplane slice, as shown in Figure 3.14. Figure 3.15 shows the first 14 singular values, related to the energy of each mode. As expected, the energy of each mode quickly drops, proving that only a few modes can capture the dominant flow structures.

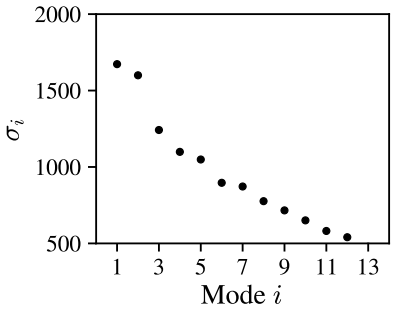


Figure 3.15: Singular value associated to the shear-layer modes.

Figure 3.16 displays the first five odd-numbered mode shapes. The even numbered mode shapes present the same patterns, with a slight shift in the streamwise direction. Figure 3.17 shows the power spectrum density (PSD) of each mode, computed based on its time coefficient $a_i(t)$. Modes come in pairs with a similar well-defined beating frequency. Hence, each pair of modes combine to form traveling structures [Taira *et al.* 2017] in the streamwise direction, namely the vortex shedding. To gain confidence in the physical meaning of these modes, we computed the inflection point of the mean streamwise velocity profiles at each streamwise location. From a hydrodynamic stability point of view, an inflection point is a necessary condition to support shear instabilities [Schmid *et al.* 2002]. In the interaction region, the velocity profiles exhibit several inflection points. The selection is based on an iterative procedure. At the left hand side of the domain, the first inflection point in the laminar boundary layer profile is chosen. Then the next inflection point is chosen as the closest to the previous one. The resulting inflection line is added on top of all mode shapes in Figure 3.16. The line clearly crosses the streamwise momentum structures corresponding to the vortex shedding. This supports the Kelvin-Helmoltz instability mechanism as the root of shear layer

vortex shedding. Secondly, the POD modes clearly highlight the coupling between the vortex shedding and the bow shock movement described above. In fact, the perturbations in the shear layer correspond to opposite sign perturbations along the bow shock in all modes. This indicates the spatial correlation between these two features. Physically, pressure decreases during a shedding event. The suction effect then locally accelerates the flow and the bow shock changes its angle and position locally.

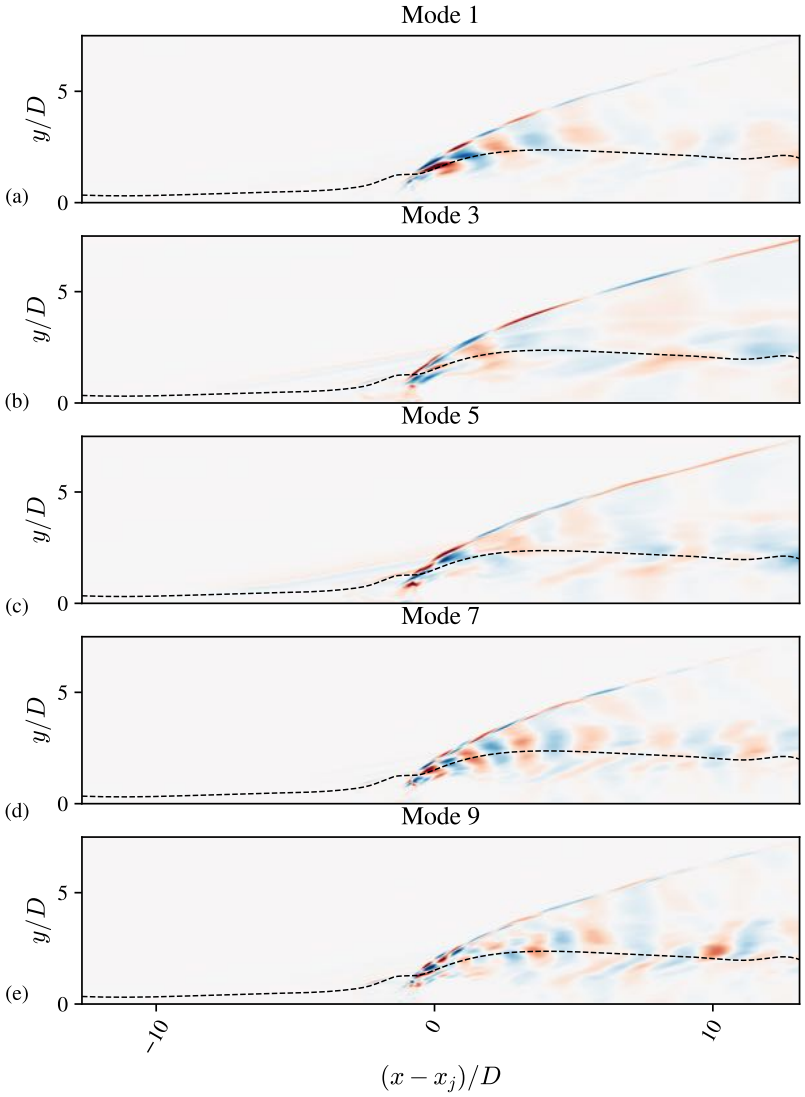


Figure 3.16: First 5 odd POD modes of the mid-plane slice for streamwise momentum ρu. The dashed line represent the mean-flow inflection line.

In terms of temporal scales, the fundamental frequency (modes 3 and 4) corresponds to a Strouhal number of $St = 0.5$. This value is in the range of the values reported in other numerical and experimental studies of jet in

supersonic crossflow [Chai *et al.* 2015, Kawai & Lele 2010, Peterson *et al.* 2006, Miller *et al.* 2018, Ben-Yakar *et al.* 2006].

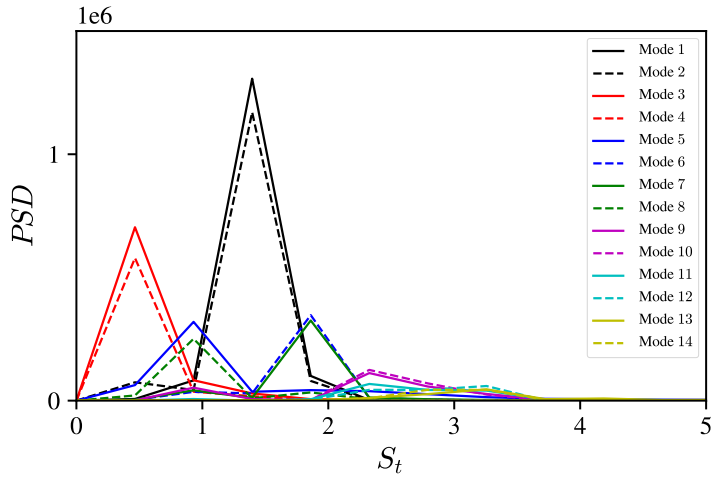


Figure 3.17: Power spectrum density of shear-layer modes for the cold $Ma_\infty = 5$ 3D JISC case.

The streamwise energy distribution of each pair of modes is plotted in Figure 3.18. As expected, the first pair is the most energetic overall. However, the second pair becomes more energetic 5 jet diameter after the jet injection. If we compare the first and third mode shapes in Figure 3.16 (a) and (b) respectively, the third mode has a larger wavelength in the streamwise direction. Physically, this energy transfer between modes corresponds to the elongation and rotation of shear-layer vortices, as seen in Figure 3.14(b-f).

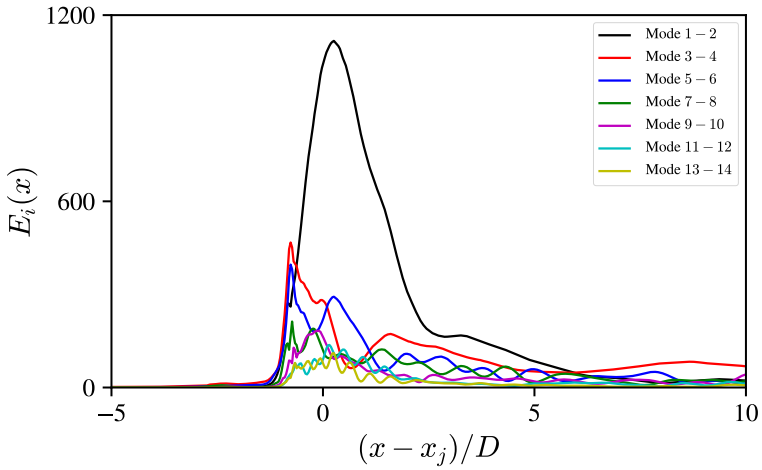


Figure 3.18: Streamwise energy distribution of shear-layer modes pair for the cold $Ma_\infty = 5$ 3D JISC case.

In summary, the Proper Orthogonal Decomposition modes prove to be highly

effective in extracting coherent vortex shedding structures and capturing their evolution in the context of the jet in crossflow interaction. By applying POD on two-dimensional jet data, we are able to identify and isolate the dominant flow patterns associated with the vortex shedding in the shear layer. These POD modes provide valuable insights into the dynamic behavior and development of these vortical structures, allowing for a deeper understanding of the complex flow physics involved in jet in crossflow phenomena.

Chapter 4

Self-similar solutions for hypersonic boundary layers

Contents

The reference solution enforced in the numerical sponge at the outer portion of the computational domain (Figure 2.1) is usually chosen as a (locally) self-similar solution. This section outlines the mathematical formulation of these (locally) self-similar solutions for hypersonic boundary layers. The focus is set on the lesser known locally self-similar solution in chemical nonequilibrium [Lees 1956, Di Renzo & Urzay 2021, Williams *et al.* 2021]. In addition to their use as reference solutions for simulation in chemical nonequilibrium, they also provide a fairly accurate estimation of the thermodynamic states and composition encountered in a typical scenario of a hypersonic boundary layer.

4.1 Formulation

4.1.1 Conservation equations for two-dimensional boundary layers

Let us consider conservation equations for mass, species, momentum and stagnation enthalpy. for a steady two-dimensional boundary layer. These equations are derived from the Navier-Stokes equation, Eqs. (1.1-1.4), using the typical order of magnitude analysis, i.e. $\delta \ll L$, where δ is the boundary layer thickness in the wall normal direction and L is a reference length in the streamwise direction. The equations simplify to the following form

$$\frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0, \tag{4.1}$$

$$\rho u\frac{\partial Y_s}{\partial x}+\rho v\frac{\partial Y_s}{\partial y}=-\frac{\partial}{\partial y}\left(\rho Y_s\mathbf{V}_{s,y}\right)+\dot{\omega}_s\ ,\quad \forall s\in S, \tag{4.2}$$

$$\rho u\frac{\partial u}{\partial x}+\rho v\frac{\partial u}{\partial y}=-\frac{dp_e}{dx}+\frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}\right), \tag{4.3}$$

$$\rho u\frac{\partial h_0}{\partial x}+\rho v\frac{\partial h_0}{\partial y}=-\frac{\partial}{\partial y}\left(\mu\frac{\partial u}{\partial y}+\kappa\frac{\partial T}{\partial y}-\sum_{s\in S}\rho Y_s h_s\mathbf{V}_{s,y}\right). \tag{4.4}$$

$\mathbf{V}_{s,y}$ is the wall normal component of species $'s'$ diffusion velocity vector, defined in Eq. (1.21). The equations are subject to the following boundary conditions away from the wall as $y\to\infty$

$$\rho=\rho_e,\quad u=u_e,\quad Y_s=Y_{s,e},\quad T=T_e, \tag{4.5}$$

where $(.)_e$ denotes an edge quantity. At the wall ($y=0$), similarly as in §2.1, we have the classical no-slip and non catalytic boundary conditions

$$u=v=0,\quad \frac{\partial Y_s}{\partial y}=0. \tag{4.6}$$

The wall can be either adiabatic $\partial T/\partial y=0$ or isothermal $T=T_w$. We assume no streamwise pressure gradient, hence $p=p_\infty$ and all the edge quantities are equal to the freestream quantities denoted by $(.)_\infty$.

4.1.2 Locally self-similar transformation

Following Lees [Lees 1956], as well as the more recent work of Di Renzo et al. [Di Renzo & Urzay 2021] and Williams et al. [Williams *et al.* 2021], we introduce the similarity variables

$$\xi(x)=\rho_e\mu_e u_e x\quad\text{and}\quad \eta(x,y)=\frac{u_e}{\sqrt{2\xi}}\int_0^y\rho dy. \tag{4.7}$$

By introducing a streamfunction $\psi=f(\eta)/\sqrt{2\xi}$, such that

$$u=u_e f^{'},\quad\text{and}\quad \rho v=-\rho_e\mu_e u_e\frac{f}{\sqrt{2\xi}}-\eta_x f^{'}\sqrt{2\xi}, \tag{4.8}$$

the continuity equation Eq. (4.1) is automatically satisfied. $(.)^{'}$ indicates differentiation with respect to η and $\eta_x=\partial\eta/\partial x$. With this set of variables, the momentum equation reduces to

$$\left(Cf^{''}\right)^{'}+ff^{''}=0, \tag{4.9}$$

where $C=\rho\mu/(\rho_e\mu_e)$ is the Chapman-Rubesin parameter. Using the species diffusion velocity formulation of Di Renzo et al. [Di Renzo & Urzay 2021], which is equivalent to Eq. (1.21), each species conservation equation can be cast into the

following form,

$$\left[\frac{C_s Y_s}{Sc_{e,s}}\left(\frac{X_s^{'}}{X_s}-\sum_{j\in S}\frac{D_j Y_j}{D_s X_s}X_j^{'}\right)\right]+fY_s^{'}+Da_s\dot{\Omega}_s=0,\quad \forall s\in S. \tag{4.10}$$

Similarly as the Chapman-Rubesin parameter for momentum, $C_s=\rho^2 D_s/(\rho_e^2 D_{s,e})$ is a nondimensional group of variable for mass diffusion. $Sc_{s,e}=\mu_e/(\rho_e D_{s,e})$ is the Schmidt number based on edge conditions, representing the ratio of momentum to mass diffusion. $Da_s=x/(u_e t_{ch,s})$ is a Damköhler number defined as a ratio between streamwise residence time and the characteristic chemical time scale $t_{ch,s}$ of species s, based on the maximum temperature in the boundary layer. Finally, $\dot{\Omega}_s=t_{ch,s}\dot{\omega}_s/\rho$ is the normalized production rate of species s.

The equation for stagnation enthalpy takes the following form,

$$\frac{c_{p,e}T_e}{h_{0,e}}\left(\frac{C_T\theta^{'}}{Pr_e}\right)^{'}+fm^{\prime}+\frac{u_e}{h_{0,e}}\left(Cf^{'}f^{''}\right)^{'}+\sum_{s\in S}\frac{h_{s,e}}{h_{0,e}}\left[\frac{C_s g_s Y_s}{Sc_{e,s}}\left(\frac{X_s^{'}}{X_s}-\sum_{j\in S}\frac{D_j Y_j}{D_s X_s}X_j^{'}\right)\right]^{'}=0, \tag{4.11}$$

where $\theta=T/T_e$, $C_T=\rho\kappa/(\rho_e\kappa_e)$ and $Pr_e=\mu_e c_{p,e}/\kappa_e$ is the Prandtl number based on edge condition. Additionally, $g_s=h_s/h_{s,e}$ is the species-specific enthalpy, normalized by its edge value and equivalently for the stagnation enthalpy, $m=h_0/h_{0,e}$.

4.1.3 Thermochemical model

The self-similar equations (4.9-4.11) are closed with the perfect gas equation of state, rewritten in terms self-similar variables as

$$\theta=\frac{\rho_e\overline{\overline{W}}}{\rho\overline{\overline{W}}}. \tag{4.12}$$

Similar to the Navier-Stokes equations (1.1-1.4), the solution of Eqs. (4.9-4.11) requires closure by modeling different nondimensional terms representing transport of momentum C, mass C_s, and energy C_T, as functions of the thermodynamic state and composition.

For the five thermochemical models devised in §1.2, only the CNEQ model necessites the resolution of Eqs. (4.10). In all other cases, these equations can be omitted and the third term on the left-hand side of Eq. (4.11) vanishes. In any of these cases, the solution obtained is only a function of η through $f(\eta)$ and becomes independent of the streamwise location. Therefore, it can be coined a self-similar solution. In the physical space, a one-dimensional self-similar profile can be scaled with $\sqrt{x}$ to recover the full flowfield.

However, in the case of CNEQ, the edge Damköhler number $Da_s=x/(u_e t_{ch,s})$ introduces a dependency with respect to the streamwise location x, in Eqs. (4.10). Hence, the solution is a function of η and x and can only be coined a locally self-

similar solution. In other words, in the physical space, at each streamwise location of interest, a new solution has to be recalculated.

4.1.4 Numerical resolution

The locally self-similar solutions in chemical nonequilibrium presented herein were generously computed by C.T. Williams and M. Di Renzo during Stanford CTR summer program 2022. Their method uses a second-order finite-difference discretization and a Newton-Raphson solver to numerically solve Eqs. (4.9-4.11), supplemented with Eq. (4.12) [Williams *et al.* 2021, Di Renzo & Urzay 2021].

4.2 Results

This section investigates the accuracy of locally self-similar solutions in chemical nonequilibrium by comparing to direct numerical simulation.

For each case, locally self-similar solutions are generated at each discrete streamwise location and interpolated onto the final grid. The resulting field of locally self-similar solutions is used as the initial and reference solution for the numerical sponge. The solution is then advanced with the Navier-Stokes solver until convergence of the residuals to machine precision.

4.2.1 An isothermal boundary layer at suborbital enthalpy

The comparison is first carried out for a two-dimensional, Mach-10, cooled-wall hypersonic boundary layer, based on the conditions studied by [Di Renzo & Urzay 2021]. Lengths are non-dimensionalized using the displacement thickness δ_0^* at the inflow, with a corresponding Reynolds number $Re_{\delta_0^*} = 6,000$. The freestream temperature is $T_\infty = 1039$ K, and the wall temperature is set to $T_w = 1700$ K. Freestream pressure is set to $p_\infty = 57.1$ kPa. The computational domain extends from $x = 65\delta_0^*$ to $265\delta_0^*$ in the streamwise direction, with 960 grid points uniformly distributed. In the wall-normal direction, the domain extends from $y = 0$ to $5\delta_0^*$, and 211 grid points are used, clustered near the wall using Eq. (3.1) to properly resolve the large gradients in that region.

The self-similar solutions were generated using 0.5 core hours on the Yellowstone cluster at Stanford, compared with 1,500 core hours needed to obtain the converged laminar DNS.

Figure 4.1 displays a comparison of the DNS and self-similar profiles at $R_x = 1000$. Despite the strongly cooled-wall, aerodynamic heating produces temperatures of the order of $T_{max} \approx 4T_\infty$ at a wall-normal distance of $y = 0.8$, as shown in Figure 4.1(b). This temperature is sufficiently high to promote dissociation of O_2 into its atomic counterpart, as well as to produce NO. However, it is too low to produce significant amounts of atomic N, as shown in Figure 4.1(c). The temperature then decreases towards the wall after the peak aerodynamic heating,

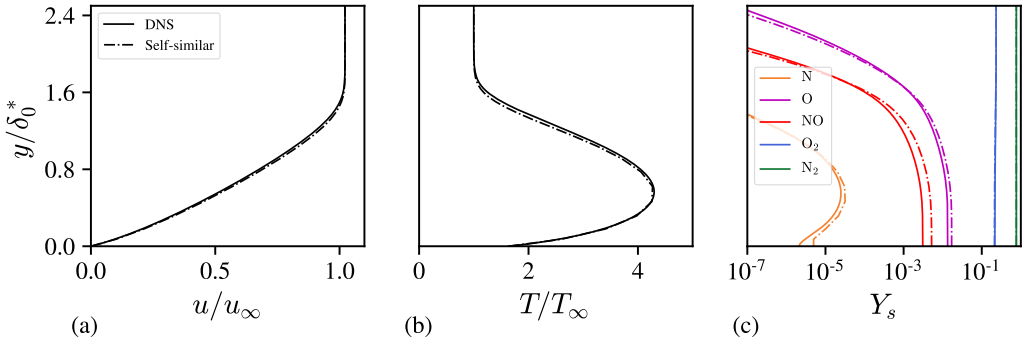


Figure 4.1: (a) Streamwise velocity, u, (b) temperature, T, and (c) species mass fraction, Y_s, profiles in the boundary layer for Di Renzo's [Di Renzo & Urzay 2021] isothermal boundary layer at $R_x = 1000$. DNS results (solid lines) are compared to the locally self-similar solution (dash-dotted). (c) From left to right : $\mathrm{N}, \mathrm{NO}, \mathrm{O}, \mathrm{O}_2, \mathrm{N}_2$

due to wall cooling. Mass fractions of O and NO, on the other hand, stay relatively high, owing to the diffusion of radicals away from the wall.

Almost perfect agreement is observed for the velocity and temperature profiles. The most notable difference is in the species mass fractions, where higher dissociation is observed in the self-similar solution than in the DNS. This can be explained by the streamwise species diffusion flux that is neglected in the locally self-similar solution.

4.2.2 Case A

The second comparison uses the adiabatic Mach-10 boundary layer, case A, for which a complete description is done in §3.1.3.2. The profiles of Figure 3.4 are juxtaposed to the initial profiles, computed with a locally self-similar solution in chemical nonequilibrium, at $R_x = 2000$ in Figure 4.2.

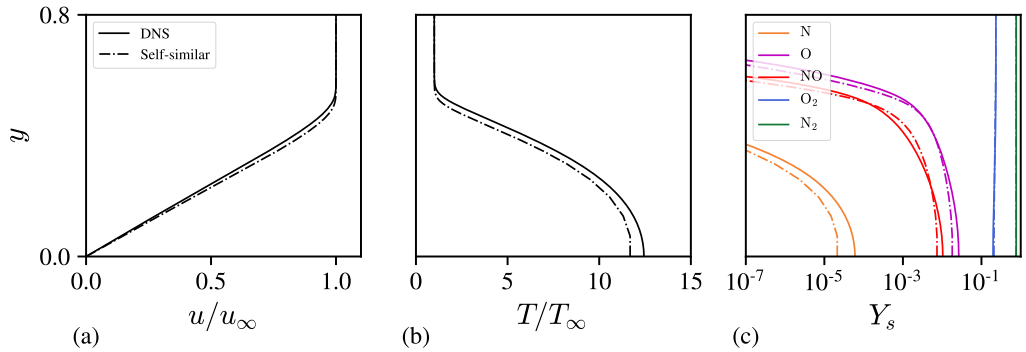


Figure 4.2: (a) Streamwise velocity, u, (b) temperature, T, and (c) species mass fraction, Y_s, profiles for case A at $R_x = 2,000$. DNS results (solid lines) are compared to the locally self-similar solution (dash-dotted). (c) From left to right : $\mathrm{N}, \mathrm{NO}, \mathrm{O}, \mathrm{O}_2, \mathrm{N}_2$

In this adiabatic set-up, a higher discrepancy is found between the different profiles. The adiabatic temperature at the wall is significantly higher at $T_{\text{wall}}/T_\infty = 12.44$, compared to $T_{\text{wall}}/T_\infty = 11.68$ for the self-similar solution. This difference of 266 K induces more dissociation within the boundary layer, as seen on Figure 4.2(c) where the radical species N, O and NO are all present in higher quantities near the wall. Finally the DNS boundary layer is thicker than the self-similar counterpart, Figure 4.2(a). This larger difference between the self-similar solution and DNS in this caser can be explained by higher maximum temperature, where $T_{\text{wall}} \approx 4900$ K. This theoretically leads to a greater degree of dissociation, resulting in increased streamwise diffusion that is not considered in the locally self-similar solution.

Part II

Data-driven reduced order thermochemical model

Chapter 5

Algorithm for self-learning of reduced-order thermochemical model

Contents

In this chapter, we first detail the strategy leading to the different training steps of a reduced-order thermochemical model for hypersonic flow simulation in chemical nonequilibrium. Secondly, we focus on the coupling with the flow solver for closed-loop simulations. The case chosen to showcase the different steps is the steady adiabatic $Ma = 10$ boundary layer, described in details in 3.1.3.2. Finally, the last section presents the application of the reduced-order thermochemical model for the efficient simulation of steady hypersonic flow in chemical nonequilibrium. Three cases are chosen, case A, §3.1.3.2, an isothermal boundary layer at suborbital enthalpy, §4.2.1, and the shock-wave boundary layer interaction scenario presented in §3.2.2.

This section is based on our recent publication on the topic [Scherding *et al.* 2023].

5.1 Strategy

As described in §2.4, high-fidelity simulations of hypersonic flows in chemical nonequilibrium rely on detailed thermochemical models of the mixture properties. In the context of this work, the mixture properties are obtained by coupling the flow solver with the Mutation++ library [Scoggins *et al.* 2020]. It can be seen as an input/output problem $\mathbf{z} = f(\mathbf{x})$. More precisely, given the local state vector

$$\mathbf{x} = [\rho, \rho_s, \rho e] \in \mathbb{R}^D, \tag{5.1}$$

the library returns all physico-chemical properties needed to close the governing equations

$$\mathbf{z} = f(\mathbf{x}) = [p, T, \mu, \kappa, D_s, h_s, \dot{\omega}_s] \in \mathbb{R}^{Dz}. \tag{5.2}$$

As shown in Figure 2.7, these functions calls increase by more than one order of magnitude the cost of simulation in CNEQ compared to an equivalent CPG simulation. While these function calls cannot be entirely avoided, existing features of the flow inspire strategies to seek a less expensive method to evaluate the required properties. (i) Flows have history. In other words, several calls to the library may be redundant since some thermodynamic states are seen multiple times throughout the simulation. (ii) Any flow of interest contains only a subset of all possible thermodynamic states, given its nature and freestream conditions. Hence, only a small subset of the input space of function $\tilde{f}$ needs to be accessed. (iii) While data-driven method requires a lot of data for training, some (rare) flows of interest, such as hypersonic boundary layers in chemical non-equilibrium, exhibit elegant, locally self-similar solutions [Lees 1956] that can be used for training instead of an expensive direct numerical simulation (DNS). This final point will be explored in §6.2.2.

The proposed algorithm leverages these features by creating a surrogate model of the function $\tilde{f}$ only on a subset of input states relevant to the simulation, which is commonly represented as a low-dimensional manifold in $\mathbb{R}^D$. This allows us to first perform dimensionality reduction of the input space (see [Bouhlel *et al.* 2016, Hawchar *et al.* 2017]). Next, following a similar approach as in [Bettebghor *et al.* 2011], regions with different dynamics and/or discontinuities between them are clustered into a low-dimensional representation. Finally, surrogate models are constructed on each cluster of this low-dimensional space. Hence, the training of the algorithm is performed in three steps: (i) dimensionality reduction, (ii) community clustering, and (iii) surrogate model construction. Once trained, the model replaces the look-up library already in place to predict the thermochemical properties of the mixture within the flow solver. We stress that the lighter version of the library will perform correctly only on the range of conditions seen during the simulation. A general schematic of the training process and the coupling with the flow solver is presented in Figure 5.1.

It should be noted that this strategy is also applicable to flows in thermal non-equilibrium, where the internal energy modes are out of equilibrium with the

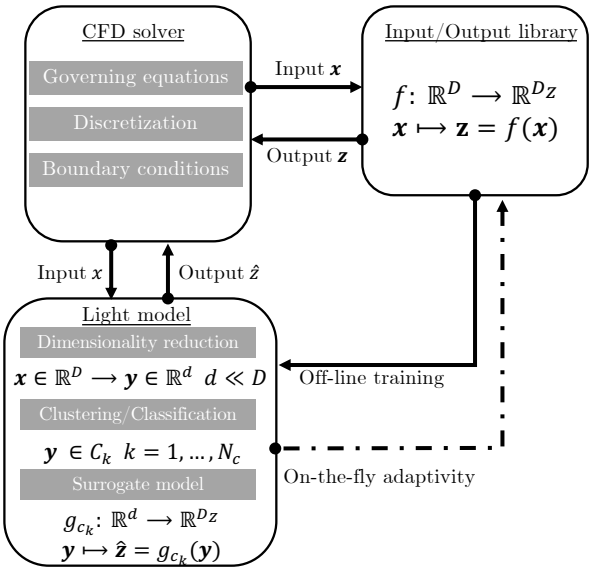


Figure 5.1: General schematic of the model training and coupling to replace any expensive Input/Output library.

translational energy of the flow [Munafò *et al.* 2015]. Since the additional source terms modeling the energy exchange in the internal energy equation(s) are modeled as a function of the local thermodynamic state vector as well, they can be added to the outputs and a surrogate can be constructed accordingly. This endeavor however lies outside of the scope of the current study and will be pursued as a subject of a future work.

5.2 Training

5.2.1 Data acquisition

To train the model, N thermodynamic state vectors $\tilde{\mathbf{q}}_{\mathrm{th}}$ are randomly sampled on the grid of a previously converged simulation in chemical non-equilibrium and concatenated into the input vector $\tilde{\mathbf{X}} \in \mathbb{R}^{N \times D}$ (here, $D = 6$). The corresponding outputs from the library are collected and concatenated into the output vector $\tilde{\mathbf{Z}} \in \mathbb{R}^{N \times D_Z}$, with $D_Z = 18$. Figure 5.2 shows the numerical range of selected output variables along each input, normalized between 0 and 1 with a minimum-maximum scaling

$$\mathbf{X} = \frac{\tilde{\mathbf{X}} - \tilde{\mathbf{X}}_{min}}{\tilde{\mathbf{X}}_{max} - \tilde{\mathbf{X}}_{min}}, \qquad \mathbf{Z} = \frac{\tilde{\mathbf{Z}} - \tilde{\mathbf{Z}}_{min}}{\tilde{\mathbf{Z}}_{max} - \tilde{\mathbf{Z}}_{min}}. \tag{5.3}$$

Taking dynamic viscosity μ, for example, it shows a strong dependency on density ρ and internal energy ρe but a low variation with respect to the radicals'

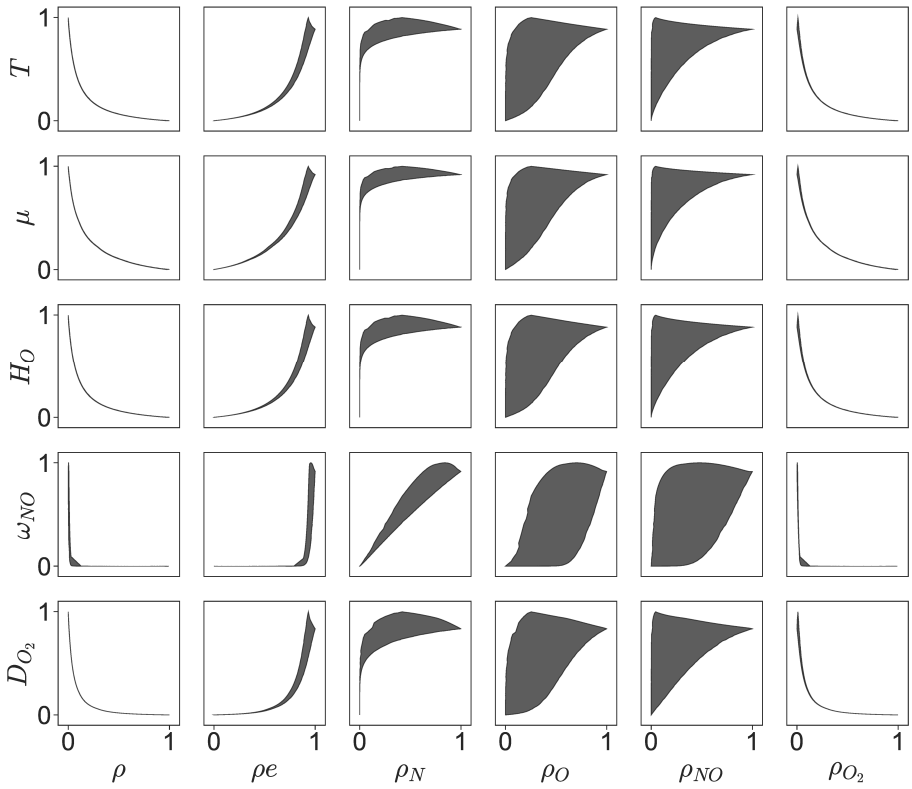


Figure 5.2: Numerical range of selected Mutation++ outputs $\mathbf{Z}$ (vertical) with respect to the inputs $\mathbf{X}$ (horizontal).

partial densities ρ_O, ρ_N and ρ_{NO}. The same observations can be made for all other outputs. Hence, the variation of the function $\tilde{f}$ with respect to the inputs can be accurately represented on a low-dimensional subspace of the inputs. This motivates the first step of the algorithm, namely, dimensionality reduction.

5.2.2 Dimensionality Reduction

The goal of this section is to find an effective algorithm for dimensionality reduction of the input space in order to construct a mapping between its reduced-order representation and the output of the library

$$\hat{\mathbf{Z}} = g(\mathbf{Y}), \tag{5.4}$$

where g is the approximation of the scaled library f, in the low-dimensional subspace of the inputs, $\mathbf{Y}$ is the reduced-order representation of an input $\mathbf{X}$ and $\hat{\mathbf{Z}}$ the prediction of the model. The benefit of this first pre-processing step is to maintain high accuracy of the surrogate model, while decreasing the overall cost of construction and evaluation. In fact, constructing a response surface faces the well-known curse of dimensionality; as the number of input dimensions increases,

the cost of constructing an accurate surface increases exponentially. This approach was proven successful in [Bouhlel *et al.* 2016] where PLS was used in tandem with kriging to reduce the dimension of the input space.

5.2.2.1 Principal component analysis

The most common technique for dimensionality reduction of a dataset $\mathbf{X} \in \mathbb{R}^{N\times D}$ in high dimensions is principal component analysis (PCA) (see eg. [Shlens 2014]). The principal components of $\mathbf{X}$ are found through the eigenvalue decomposition of the correlation matrix $\mathbf{X}\mathbf{X}^T$ of the data. The dataset $\mathbf{X}$ is then projected onto $d < D$ leading eigenvectors (or principal components) of the covariance matrix, resulting in a low-dimensional representation $\mathbf{Y} \in \mathbb{R}^{N\times d}$ of the original dataset. Alternatively, the principal components can be found directly by applying the singular value decomposition to $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. They correspond to the left singular vector in $\mathbf{U}$. In the fluid community, this technique is known as proper orthogonal decomposition (POD) and has been applied to the jet in hypersonic crossflow in §3.3.3.

However, depending on the shape of the manifold, the variations of the output variables with respect to the low-dimensional sub-space may not be properly preserved, which is the case presented in Figure 5.3(a) with points of high and low temperature projected onto similar locations. This example illustrates the limitations of PCA for dimensionality reduction of a dataset constrained to a nonlinear manifold.

5.2.2.2 Auto-encoders

Nonlinear dimensionality reduction via auto-encoders (AE) typically have a higher compression rate than linear techniques. An auto-encoder is a parametric model (i.e. a deep neural network with an activation function σ) that embeds the input dataset $\mathbf{X} \in \mathbb{R}^{N\times D}$ into a low-dimensional representation $\mathbf{Y} \in \mathbb{R}^{N\times d}$ through an encoder function $\mathbf{E}$. The low-dimensional representation is then decoded back to the input space with the decoder function $\mathbf{D}$, producing a reconstruction of the input $\widehat{\mathbf{X}} \in \mathbb{R}^{N\times D}$.

$$\begin{aligned} \mathbf{Y} &= \mathbf{E}(\mathbf{X}) \\ \hat{\mathbf{X}} &= \mathbf{D}(\mathbf{Y}) \end{aligned} \tag{5.5}$$

The weights of the two networks $\mathbf{E}$ and $\mathbf{D}$ can be trained using back-propagation of the L_2 error $\|\mathbf{X} - \widehat{\mathbf{X}}\|_2$ through the network. If the activation function is selected as the identity (i.e $\sigma(\mathbf{x}) = \mathbf{x}$), the auto-encoder is linear and unbiased

$$\begin{aligned} \mathbf{E} &= W_E \\ \mathbf{D} &= W_D \end{aligned} \tag{5.6}$$

where W_E and W_D are the weights matrices of the encoder and decoder, respectively. These optimal weights can be found through PCA. In fact, the linear latent space of dimension d of the encoder will span the same sub-space as the top d PCA singular vectors. The equivalence between the two techniques was first shown by

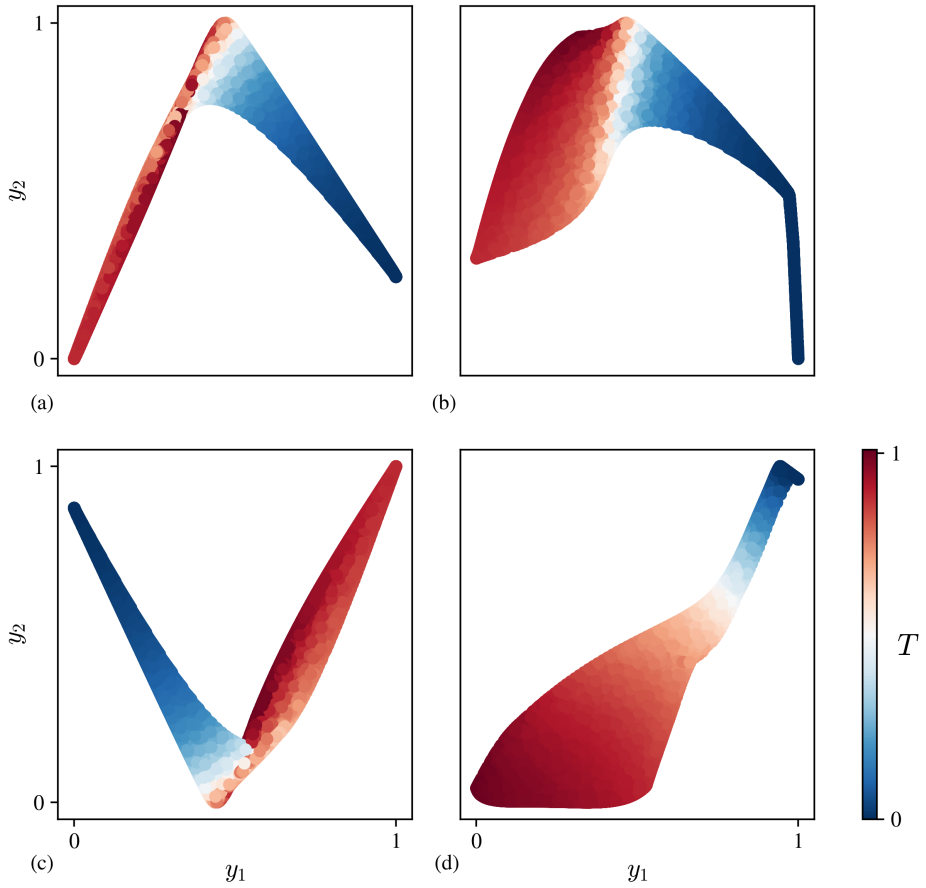


Figure 5.3: Low-dimensional representation $Y \in \mathbb{R}^{N\times d}$ $(d = 2)$ of $X \in \mathbb{R}^{N\times D}$, colored by temperature T. (y_1, y_2) denote the coordinates of the latent space. Obtained with (a) PCA (b) AE (c) PLS (d) IO-E.

Baldi & Hornik [Baldi & Hornik 1989]. Correspondingly, a two-layered nonlinear auto-encoder can be mathematically described as follows

$$\begin{aligned} \mathbf{Y} &= W_{E,2}\sigma(W_{E,1}\mathbf{X} + \mathbf{b_{E,1}}) + \mathbf{b_{E,2}} \\ \hat{\mathbf{X}} &= W_{D,2}\sigma(W_{D,1}\mathbf{Y} + \mathbf{b_{D,1}}) + \mathbf{b_{D,2}} \end{aligned} \tag{5.7}$$

where σ is a nonlinear activation function, $W_{E,1} \in \mathbb{R}^{H\times D}, W_{E,2} \in \mathbb{R}^{d\times H}$ are the weights matrices of the first and second layer of the encoder with respective biases $\mathbf{b_{E,1}} \in \mathbb{R}^H$ and $\mathbf{b_{E,2}} \in \mathbb{R}^d$. H denotes the dimension of the hidden layer. The matrices and bias vectors of the decoder have transposed dimensions. This corresponds to the minimal architecture (i.e. with one hidden nonlinear layer and an output linear layer) requested by the universal approximation theorem [Cybenko 1989]. However, more hidden layers can be considered. Figure 5.3(b) shows the manifold unrolled in two dimensions with an auto-encoder, colored by the magnitude of the temperature, an output of the library. The AE outperforms PCA

by preserving the local structure and preventing points at different thermodynamic states (i.e different temperatures) to be projected onto the same location. However, the highest temperature zone is concentrated in a thin layer adjacent to the lower temperature area. This will result in strong and unphysical gradients of the surrogate model within this region.

5.2.2.3 Partial least-squares

Since our interest lies in reducing the dimensionality of the input to construct a reduced-order surrogate model of the input/output relations, it is useful to entangle the input into a low-dimensional space that best reconstructs the outputs. In analogy to PCA finding dependencies between the inputs, partial least-squares (PLS) finds a basis of the input space that optimally accounts for features in the output space. It has been used to construct surrogate models aimed at reducing the dimensions of the input space (see [Bouhlel *et al.* 2016]). Different variants of PLS now exist, using either a singular value decomposition (PLS-SVD) or iterative algorithms (such as PLS-W2A in [Wegelin 2000]). While it has been shown that in cases where the dimension of the latent space is strictly greater than one, PLS-SVD differs from PLS-W2A and its variant PLS2, no major differences in the resulting latent space were observed. The results of PLS-SVD are presented here to highlight the similarity to PCA. Given the input $\mathbf{X}$ and output vectors $\mathbf{Z}$, the PLS-SVD algorithm [Wegelin 2000] determines

$$\mathbf{X}^T\mathbf{Z} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T. \tag{5.8}$$

Similar to PCA, the projection of the input is then obtained by projecting onto the $d < D$ top left singular vectors

$$\mathbf{Y} = \mathbf{X}\overline{\mathbf{U}} \tag{5.9}$$

where $\overline{\mathbf{U}} \in \mathbb{R}^{D\times d}$ is the truncated left-singular matrix. Figure 5.3(c) shows the training set projected onto the two-dimensional basis generated with PLS. As expected, adding the output information in the computation improves the output's representation in the reduced-order input basis compared to PCA. However, an artificial discontinuity is created near the high-gradient region that was not present in Figure 5.3(b). This highlights the lower compression rate of linear techniques compared to nonlinear ones.

5.2.2.4 Input/output-encoders

The strategy adopted here is therefore a nonlinear dimensionality reduction of the input data using input/ouput-encoders (IO-E), a modified version of auto-encoders (AE) adapted to input/ouput relations. To that end, the decoder architecture is

modified to best reconstruct the output of the library $\mathbf{Z} \in \mathbb{R}^{N \times D_Z}$ as

$$\widehat{\mathbf{Z}} = W_{D,2}\sigma(W_{D,1}\mathbf{Y} + \mathbf{b_{D,1}}) + \mathbf{b_{D,2}} \tag{5.10}$$

where the weight matrices are now $W_{D,1} \in \mathbb{R}^{H_D \times d}, W_{D,2} \in \mathbb{R}^{D_Z \times H_D}$ with respective biases $\mathbf{b_{D,1}} \in \mathbb{R}^{H_D}$ and $\mathbf{b_{D,2}} \in \mathbb{R}^{D_Z}$. H_D now represents the hidden dimension of the decoder. The resulting network is then trained via back-propagation of the reconstruction error based on the output $\|\mathbf{Z} - \widehat{\mathbf{Z}}\|_2$. The input/output-encoder (IO-E) architecture is illustrated in fig. 5.4.

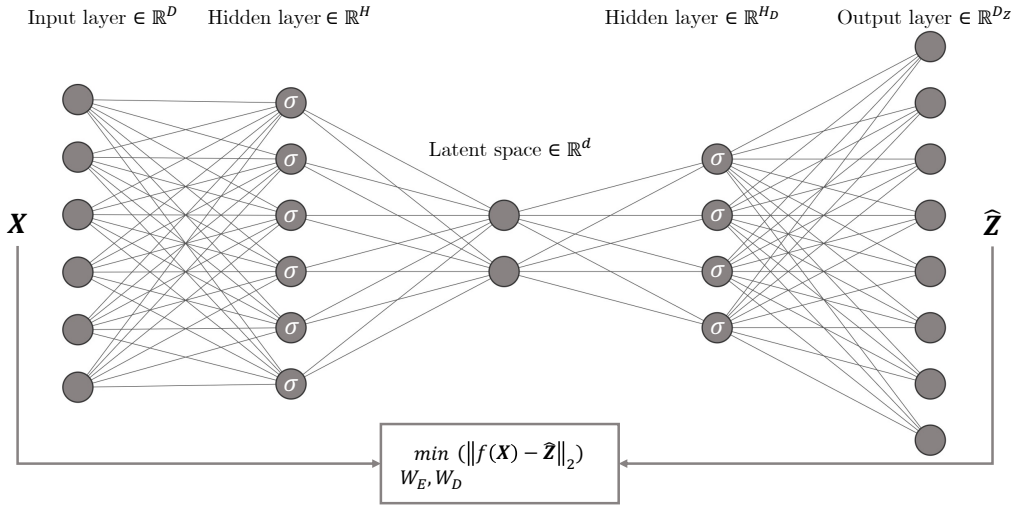


Figure 5.4: Schematic of the input/output-encoder architecture proposed. An input sample $\mathbf{X}$ is passed through the encoder network to generate a low-dimensional representation $\mathbf{Y}$. $\mathbf{Y}$ is decoded back to predict the library output $\widehat{\mathbf{Z}}$. At each training step, the L_2 loss $\|\mathbf{Z} - \widehat{\mathbf{Z}}\|_2$ between the library and the network prediction is calculcated and backpropagated until convergence of the encoder and decoder weights, W_E and W_D, is reached.

Figure 5.3(d) shows the equivalent manifold using the IO-E architecture, with the same number of hidden dimensions. The IO-E architecture outperforms all other techniques as the high-temperature region is now projected onto its own properly defined zone with smooth variations. This feature will have a marked impact on the performance of the library when linked to the solver. It is worth noting that the decoder part of the IO-E could be used directly to predict the output of the library. However, we will see in §5.4.1.4 that the accuracy of the full IO-E is not optimal, motivating the application of clustering algorithms to separate regions with varying behavior in the reduced dimension to improve the performance of the regressor.

5.2.3 Community clustering

In the second step of the algorithm, we seek to discover clusters within our data. In our present context, a cluster represents a subset of data that shares

similar thermodynamic features. These feature classification will then be useful in constructing a dedicated surrogate surface of the low-dimensional input manifold. To this end, Newman's spectral algorithm for community detection in a network [Newman 2006] is used. A clear advantage of Newman's algorithm is that the number of clusters is not defined *a priori*, in contrast to more common clustering techniques (e.g. k-means). The number of clusters is instead the result of a maximization procedure performed on the modularity Q of the network. In other words, the number of thermodynamic clusters in the flow are determined only from the data and is not based on *a priori* knowledge of the user. This knowledge might be even impossible to come by in complex, unsteady hypersonic flows subjected to shocks.

Following this approach, the Euclidean distance matrix Δ of the dataset in low-dimensional space is computed

$$\Delta_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|^2 \quad \text{for} \quad (\mathbf{Y}_i, \mathbf{Y}_j) \in \mathbf{Y}^2. \tag{5.11}$$

The dataset is subsequently recast into an undirected network with a binary adjacency matrix A, constructed as

$$A_{ij} = \begin{cases} 1 & \text{if } \Delta_{ij} < \varepsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{5.12}$$

Two points are connected with an edge if their Euclidean distance is below a certain threshold ε. This threshold is usually chosen as a fraction of the mean of the distance matrix Δ. The influence of the threshold ε on the number of clusters will be investigated in §5.4.1.2.

Finally, the dataset is progressively split into two communities until the modularity, Q, is maximized. The modularity is defined as the proportion of edges contained within a community over the same proportion for a random reference network. Let k_i be the number of edges pointing towards the data point numbered i, and m the total number of connections within the network. Then, the probability of having an edge between i and j in the random reference network is $k_i k_j / m$. Hence, the modularity Q is defined as,

$$Q = \frac{1}{m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{m} \right) \delta_{c_i, c_j}, \tag{5.13}$$

where δ_{c_i,c_j} is the Kronecker delta for the communities of i and j. For instance, $\delta_{c_i,c_j} = 1$ only when i and j belong to the same community. By defining a vector $\mathbf{s}$ where $s_i = 1$ if vertex i belongs to the first group, and $s_i = -1$ otherwise, we can reformulate

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{m} \right) (s_i s_j + 1) = \frac{1}{2m} \mathbf{s}^{\mathbf{T}} B \mathbf{s}, \tag{5.14}$$

where the modularity matrix B is given as,

$$B_{ij} = A_{ij} - \frac{k_i k_j}{m}. \tag{5.15}$$

Since the graph is undirected, the modularity matrix B is symmetric and the modularity Q represents a Rayleigh quotient of matrix B. In order to maximize Q, we need to choose a vector s that is parallel to the principal eigenvector (corresponding to the largest eigenvalue) of B, $\mathbf{v}$, which can be achieved by setting $s_i = 1$ if $v_i > 0$ and $s_i = -1$ if $v_i < 0$. In order to partition the graph into more than two communities, this algorithm is repeated until the modularity of each subgraph can no longer be increased. A thorough descripion of the full algorithm can be found in [Newman 2006].

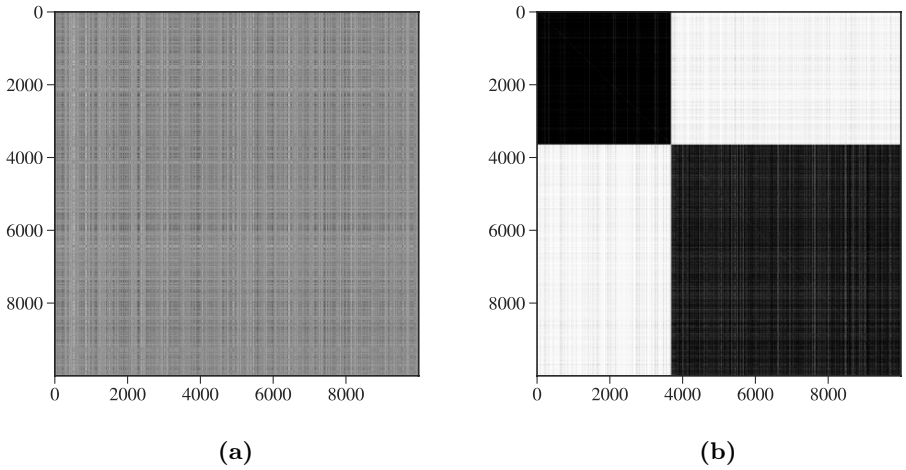


Figure 5.5: Demonstration of the Newman algorithm. (a) Original distance matrix Δ of dataset $\mathbf{Y}$. (b) Restored communities from running Newman's algorithm on A, showing two distinct clusters

Figure 5.5 shows the application of the clustering algorithm to the boundary layer data. The distance matrix Δ is constructed on $\mathbf{Y}$. After running the algorithm on the subsequent adjacency matrix A, two distinct clusters are identified, highlighted by the low distance between the points within a cluster in Figure 5.5(b).

5.2.4 Surrogate model construction

Finally, a surrogate surface is computed on the scattered low-dimensional points of each cluster. Various algorithms can be used to that end, such as kriging [Kleijnen 2009, Bouhlel *et al.* 2016], artificial neural networks [Sun & Wang 2019], or radial basis functions neural networks [Broomhead & Lowe 1988, Powell 1992, Buhmann 2000]. In the application of interest to this work, radial basis functions networks (RBF) provided the best trade-off between performance and accuracy, as

well as an easy training step. However, it should be noted that our choice is not definitive and can be easily changed.

RBF can be seen as a special type of three layer deep neural network with radial basis functions as activation functions in the hidden layer, and an associated center $\mathbf{x}_i$, as shown in Figure 5.6. The parameters of the models are the N_{R} input training

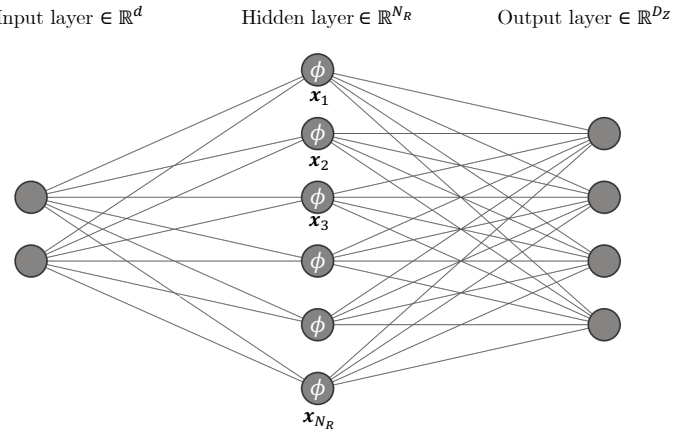


Figure 5.6: Schematic of the radial basis function network architecture.

points $\mathbf{X}^c \in \mathbb{R}^{N_{\mathrm{R}} \times d}$ and the associated weights $\mathbf{\Lambda} \in \mathbb{R}^{N_{\mathrm{R}}}$. Given the set of input points $\mathbf{x}_1, \ldots, \mathbf{x}_{N_{\mathrm{R}}} \in \mathbb{R}^d$ and the function value at these points $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{N_{\mathrm{R}}})$, the radial basis function (RBF) interpolant g is given by

$$g(\phi, \mathbf{x}) = \sum_{i=1}^{N_{\mathrm{R}}} \lambda_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) \tag{5.16}$$

where ϕ is the kernel function whose value depends on the distance $r = \|\mathbf{x} - \mathbf{x}_i\|$ between the evaluation point $\mathbf{x}$ and the center $\mathbf{x}_i$ of the RBF. The weights $\mathbf{\Lambda} = [\lambda_1, ..., \lambda_{N_{\mathrm{R}}}]^T$ that minimize the mean-square error of the RBF over the training input points can be obtained through the solution of the linear system

$$\mathbf{\Phi\Lambda} = \mathbf{f}, \tag{5.17}$$

where $\mathbf{f} = [f(\mathbf{x}_1), ..., f(\mathbf{x}_{N_R})]^T$ denotes the vector containing the function values at the RBF center. The kernel matrix $\mathbf{\Phi}$ is defined as

$$\mathbf{\Phi}_{i,j} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|). \tag{5.18}$$

In this study, the thin-plate spline kernel [Wood 2003] was used, i.e.,

$$\phi(r) = r^2 log(r). \tag{5.19}$$

With this particular kernel, the kernel matrix is only conditionally positive definite. To ensure a unique solution for the interpolation weights, the system is augmented

by a polynomial $p \in \Pi_m^d$ (space of polynomials of d variables and degree up to m) to the right hand side of eq. (5.16) [Buhmann 2000], resulting in

$$g(\mathbf{x}) = \sum_{i=1}^{N_\mathrm{R}} \lambda_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + p(\mathbf{x}). \tag{5.20}$$

The extra degrees of freedom are accounted for by enforcing orthogonality of the coefficients with respect to the polynomial space as

$$\sum_{i=1}^{N_\mathrm{R}} \lambda_i r(\mathbf{x}) = 0, r \in \Pi_m^d. \tag{5.21}$$

Finally, the polynomial coefficients $\mathbf{c} = [c_1, ..., c_{d_p}]^T$ and the RBF coefficients $\mathbf{\Lambda}$ are found through the solution of the following linear system

$$\begin{pmatrix} \mathbf{\Phi} & \mathbf{P} \\ \mathbf{P}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{\Lambda} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ 0^{d_p} \end{pmatrix}, \tag{5.22}$$

where $\mathbf{P}_i = [1, \mathbf{x}_i, ..., \mathbf{x}_i^{d_p}]$ for $i \in [1, N_\mathrm{R}]$ is the polynomial matrix.

Due to the large size of the training set, using one cluster center per training point (i.e. $N_\mathrm{R} = N = O(10^5)$) will likely result in overfitting and prohibitive computational cost [Schwenker *et al.* 2001]. Following the recommendations in [Schwenker *et al.* 2001], the interpolant is constructed in two steps. First, the *k-means* algorithm with $N_\mathrm{R} \ll N$ clusters is applied on the concatenation of the input and output vector $(\mathbf{X}, \mathbf{Z}) \in \mathbb{R}^{N \times (D + D_Z)}$. The addition of the output vector results in a low within-cluster variance of the outputs and will ultimately improve the surrogate model. The N_R centroids obtained with *k-means*, $\mathbf{x}^c$, are sent to the library to compute the function value vector $\mathbf{f}$. Simultaneously, they are encoded in the low-dimensional space to obtain the N_R cluster centers, $\mathbf{y}_1^c, ..., \mathbf{y}_{N_R}^c \in \mathbb{R}^d$, that will be used to train the RBF. Figure 5.7 shows the resulting tesselation in the embedded space after applying the *k-means* algorithm with $N_\mathrm{R} = 250$ for each cluster. The influence of the number of RBF centers on the quality of the surrogate model will be assessed in §5.4.1.3.

Following this approach, a single interpolant, g_{c_k}, is constructed for each cluster, c_k; in other words, g_{c_k} is the approximation of the scaled library function f on the low-dimensional subspace corresponding to cluster c_k. The advantage of having one interpolant per cluster is twofold. First, it allows the surrogate model to best fit a region with a given dynamics of the high-dimensional function, especially in the presence of discontinuities, similar to the approach of Bettebghor et al. [Bettebghor *et al.* 2011]. Secondly, as the surrogate model spans a smaller range of input parameters, less centers are required to capture the given dynamics accurately, resulting in a lighter model with faster evaluation time. To enforce continuity of the surrogate surface near the cluster boundaries, the nearest centroids that do not

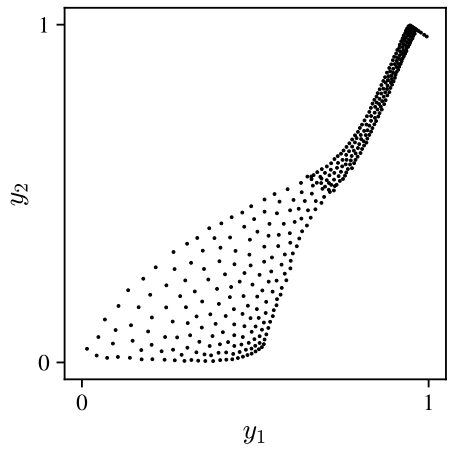


Figure 5.7: Tesselation of the low-dimensional space after applying the *k-means* algorithm with $N_{\mathrm{R}} = 250$ on the concatenation of the input $\mathbf{X} \in \mathbb{R}^{N \times D}$ and output $\mathbf{Z} \in \mathbb{R}^{N \times D_Z}$ vectors. Black dots represent the cluster centroids' low-dimensional representation $\mathbf{y}_1^c, ..., \mathbf{y}_{N_{\mathrm{R}}}^c$.

belong to the considered cluster are added to its training set.

5.3 Coupling to the flow solver

Once the model is trained, we can replace the calls of the solver to the look-up library with the new lighter model. New points have to go through three steps: (i) Out-of-sample dimensionality reduction, (ii) classification, and (iii) interpolation. These three steps will be described in the following. Let $\mathbf{X}^t \in \mathbb{R}^{N_t \times D}$ denote the stack of all new points.

5.3.1 Out-of-sample encoding

The low-dimensional representation of $\mathbf{X}^t$ (after proper scaling of $\tilde{\mathbf{X}}^t$) is straightforward. Indeed, the point is simply fed to the encoder portion of the input/output-encoder

$$\mathbf{Y}^t = W_{E,2}\sigma(W_{E,1}\mathbf{X}^t + \mathbf{b_{E,1}}) + \mathbf{b_{E,2}}. \tag{5.23}$$

This results in a fast and inexpensive encoding of the new out-of-sample points. In fact, the time complexity of the encoding step is $O(H \times C_{ac} \times L \times N_t)$, where H is the maximum number of neurons in a layer, C_{ac} is the complexity of the activation function and L is the number of layers in the encoder step of the input/output-encoder.

5.3.2 Classification

The next step is to determine to which community the new state $\mathbf{Y}^t$ belongs. To this end, after applying Newman's algorithm, a random forest classifier is trained on the resulting clusters. A random forest classifier, formally proposed by Breiman [Breiman 2001], is a collection of n_{tree} tree-based classifiers, $h(\mathbf{x}, \Phi_k), k = 1, ..., n_{tree}$, where Φ_k are identically distributed random vectors. Each tree votes for the most likely class of input vector $\mathbf{x}$, and the majority wins. fig. 5.8 shows the confusion matrix C of the classifier trained on the two clusters obtained above with $n_{tree} = 20$. The clusters of the embedded new points $\mathbf{Y}^t$, not seen during the training-phase of the classifier, are predicted and compared to the true clusters (given by Newman's algorithm). The off-diagonal values count the number of points that are assigned to the wrong cluster. All off-diagonal values are zero, demonstrating the ability of the classifier to correctly predict the community of an out-of-sample point.

The time complexity of the induction time of the classifier is $O(n_{tree} \times N_t log(N_t))$ (see the book by Witten, Frank and Hall [Frank *et al.* 2004] for a demonstration), where n_{tree} is the number of decision trees in the random forest. The logarithmic term accounts for the worst case scenario for the maximum depth of each tree. However, the maximum depth is usually set to a smaller value, resulting in a complexity of $O(n_{tree} \times depth \times N_t)$, which results in relatively fast classification times.

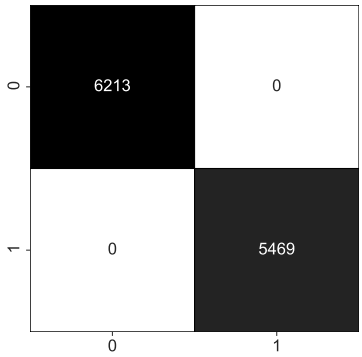


Figure 5.8: Confusion matrix C of the random forest classifier ($n_{tree} = 20$), tested on unseen data points during training. C_{ij} is the number of points belonging to cluster c_j predicted to be in cluster c_i.

5.3.3 Evaluation

Finally, once $\mathbf{Y}^t$ has been found to belong to cluster c_k, the corresponding RBF g_{c_k} is called to evaluate the thermochemical properties $\widehat{\mathbf{Z}}^t$ of the mixture at that state $\mathbf{X}^t$,

$$\widehat{\mathbf{Z}}^t = g_{c_k}(\mathbf{Y}^t) = \sum_{i=1}^{N_\mathrm{R}} \lambda_i \phi(\|\mathbf{Y}^t - \mathbf{Y}_i^c\|). \tag{5.24}$$

$\widehat{(.)}$ denotes the predicted value by the reduced library, as opposed to the true value that would have been given by the target library, $\mathbf{Z}^t$. Finally, these properties are re-scaled according to

$$\tilde{\widehat{\mathbf{Z}}}^t = \widehat{\mathbf{Z}}^t * (\tilde{\mathbf{Z}}_{max} - \tilde{\mathbf{Z}}_{min}) + \tilde{\mathbf{Z}}_{min} \tag{5.25}$$

and passed back to the flow solver.

The time complexity of each surrogate model is given by $O(C_{RBF} \times N_t \times N_\mathrm{R} \times d)$, where C_{RBF} is the complexity of the kernel function. The time limiting part of the RBF interpolation is the calculation of the distance matrix that scales with $O(N_t \times N_\mathrm{R} \times d)$. Hence, using a relatively small number N_R of RBF centers, compared to N_t, will greatly improve the performance of the surrogate model. It should be noted that the dimensionality reduction directly contributes to the added performance of the surrogate model, as $d < D$.

5.3.4 Global time performance

The time complexity of the whole algorithm can be recovered by adding the time complexity of each of the three steps. Hence, the total time complexity of the algorithm can be written as $O(C_{ML}N)$, where $C_{ML} = O(HC_{ac}L + n_{tree}depth + dN_\mathrm{R}C_{RBF})$.

5.3.5 Implementation details

The reduced-order thermochemical model is implemented in `python` in a custom object class, `thermoROM`. Common machine learning libraries are used during the different training steps:

- Input-Output encoder: TensorFlow
- Newman clustering: custom implementation using Numpy and Scipy
- k-means: Scikit-Learn
- RBF: custom implementation using Numpy and Scipy

At the start of the simulation, the reduced-order thermodynamic model class, `thermoROM`, is initialized. This step loads in memory the different objects needed: the encoder, the classifier and the RBFs. At every iteration, the variables are evaluated following the steps described above by using the `evaluate` method of the class.

Calling the class within the flow solver, written in `fortran`, required the use of the `callpyfort` library (see the GitHub repository). An additional wrapper of the class was implemented to accomodate the input and output data format required by the library.

5.4 Closed-loop testing on steady hypersonic flows

This section presents the application of the reduced-order thermochemical model for the efficient simulation of steady hypersonic flow in chemical nonequilibrium. First, the laminar Mach 10 adiabatic boundary layer (case A - §3.1.3.2) is chosen as a benchmark case. Each step of the off-line training is assessed, and then the performance of the model in predicting the output quantities when coupled to the Navier-Stokes solver is analyzed. The same strategy is then applied to the isothermal boundary layer of Di Renzo and Urzay [Di Renzo & Urzay 2021], described in §4.2.1, and the shock-wave boundary layer interaction scenario presented in §3.2.2. These additional cases are chosen to showcase the robustness of the technique to more complex thermodynamic manifold.

In all three cases, the model is trained on the converged steady solution of the Navier-Stokes equations. Starting from this steady solution, the simulation is restarted with the data-driven model in a closed-loop simulation. These tests will allow to assess the stability, accuracy and performance of the model.

5.4.1 Case A

$N = 200,000$ thermodynamic state vectors $\mathbf{x} = [\rho \ \rho e \ \rho_s]$ are sampled from the converged solution and concatenated into the input dataset $\mathbf{X}$. The corresponding Mutation++ output dataset $\mathbf{Z}$ is also obtained.

5.4.1.1 Input/output encoding

The architecture and hyperparameters of the input/output-encoder used for this test case are provided in Table 5.1.

Architecture			
Encoder		**Decoder**	
Layer	Size	Layer	Size
Input	6	Latent space	2
Fully connected	12	Fully connected	6
Fully connected	6	Fully connected	12
Latent space	2	Output	18

Hyperparameters			
Parameter	Value	Parameter	Value
Learning rate	1×10^{-3}	Epochs	2000
Loss	Mean-squared error	Batch size	256
Activation function	tanh	Optimizer	Adam (keras default)

Table 5.1: Architecture and hyperparameters of the input/output-encoder used to train the IO-E network on the $Ma = 10$ adiabatic flat plat boundary layer test case.

An important parameter is the number of dimensions required to properly unfold the input manifold. To this end, the architecture and hyperparameters, as well as the random seed to initialize the network weights, are held constant, and only the dimension of the latent space is varied. We follow the reconstruction loss $\|\widehat{\mathbf{Z}} - \mathbf{Z}\|_2$ of the testing set as a function of the latent space dimension d. As seen in Figure 5.9, the loss saturates after $d = 2$, suggesting that two dimensions are sufficient to represent the input manifold, originally in $\mathbb{R}^6$.

5.4.1.2 Newman's community clustering

The second step is to cluster the data on the low-dimensional manifold using Newman's algorithm. The only hyper-parameter needed for Newman's clustering algorithm is the threshold ε to determine the adjacency matrix A from the distance matrix Δ, as described in §5.2.3. To showcase the robustness of the number of clusters with respect to the threshold ε, the algorithm is applied over a range of thresholds, chosen as multiples of the mean of the distance matrix $\overline{\Delta}$. As shown in Figure 5.10, over the range tested, the algorithm returns $N_c = 2$ clusters before over-fitting with extreme values.

Figure 5.11(a) shows the two clusters obtained in the embedded space. As expected, the two clusters define different regions in the reduced space. To gain

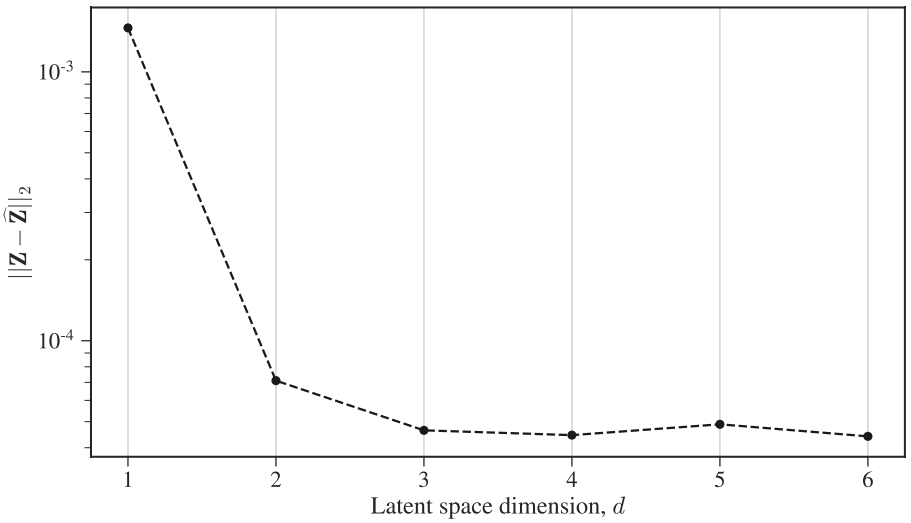


Figure 5.9: Reconstruction loss $\|\widehat{\mathbf{Z}} - \mathbf{Z}\|_2$ of the embedding done by IO-E with respect to the number of latent space dimensions d.

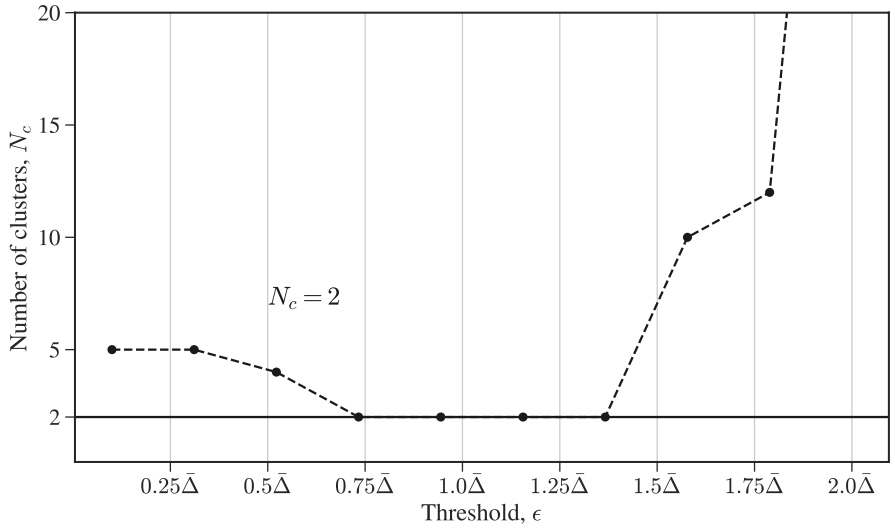


Figure 5.10: Number of clusters N_c obtained with Newman's algorithm as a function of the threshold ε, given as multiples of the distance matrix mean $\overline{\Delta}$.

more physical insight, Figure 5.11(b) shows randomly selected points of the training set, mapped back to their original location in the Cartesian space and colored by their cluster number. Contours of temperature T in Kelvin are added. Each cluster fills a region of the flow with different levels of chemical non-equilibrium. The blue cluster represents the free stream, where temperature is low and chemistry is frozen. Alternatively, the red cluster corresponds to the near-wall region with dissociated species and high temperatures.

The random forest classifier has been trained with $n_{\text{tree}} = 20$. This number proved sufficient to obtain an acceptable prediction accuracy of new points clusters, as demonstrated in Figure 5.8.

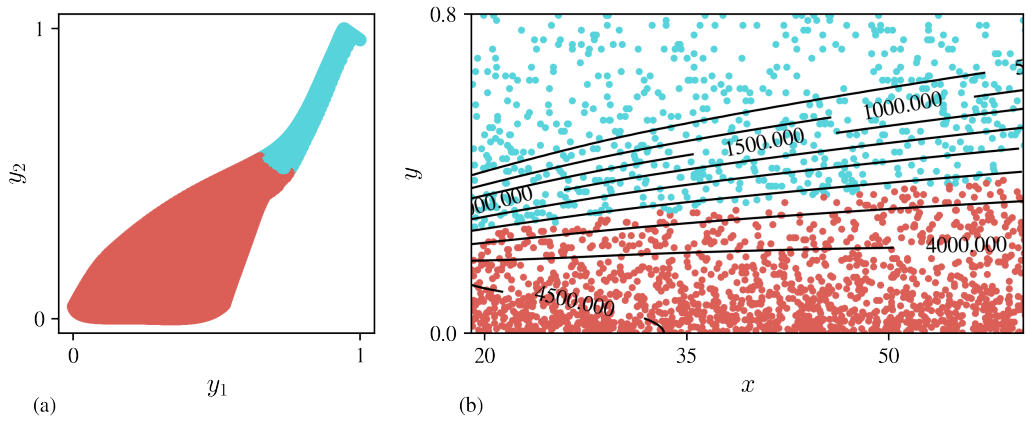


Figure 5.11: Training points $\mathbf{Y}$ colored by their cluster number. (a) In the latent space found by IO-E. (b) At the Cartesian location they were sampled from, with contours of temperature T in Kelvin.

5.4.1.3 Surrogate model construction

Once the clusters are determined, a separate RBF interpolant is trained for each cluster as described in §5.2.4.

To assess the number of centroids needed, the RBFs are trained simultaneously, for both clusters, with the same number of centroids N_{R} (note that this number can be varied to accommodate clusters of different sizes). The error of the model for the testing set $\|\widehat{\mathbf{Z}} - \mathbf{Z}\|_2$ is plotted against the number of centroids N_{R} in Figure 5.12 (dashed curve). As expected, the error decreases as the number of centroids is increased. To assess potential overfitting, one can follow the evolution of the mean of the squared RBF coefficients $\overline{\Lambda^2}$,

$$\overline{\Lambda^2} = \frac{1}{N_R} \sum_{i=1}^{N_R} \lambda_i^2 \tag{5.26}$$

where a high value will likely indicate overfitting. On the right axis of Figure 5.12, we see that $\overline{\Lambda^2}$ (dotted curve) increases with the number of RBF centroids. Therefore,

choosing the right N_{R} is a trade-off between low error and minimum overfitting. In this case, a value of $N_{\mathrm{R}} = 500$ for each cluster has been retained.

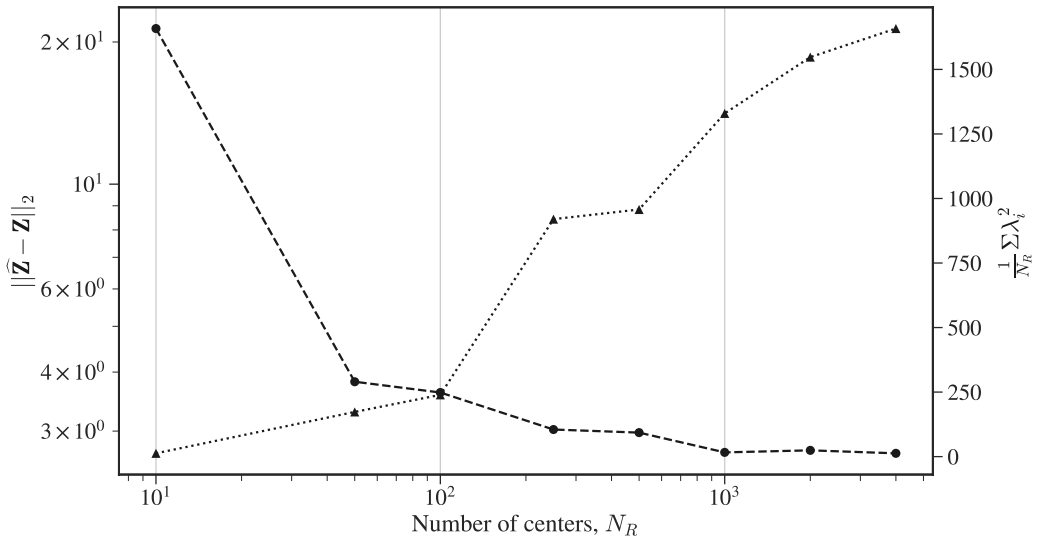


Figure 5.12: Left axis: reconstruction error of surrogate model $\|\widehat{\mathbf{Z}} - \mathbf{Z}\|_2$ on a testing set (dashed line). Right axis: mean of squared RBF coefficients $\overline{\Lambda^2}$ (dotted line). Both with respect to the number of RBF centers N_{R} used in training.

5.4.1.4 Model accuracy

The reduced library is tested (off-line) on a full snapshot (which includes also the training points used to build the model) to assess the capacity of the model to interpolate new points not encountered during training. Four configurations of the data-driven model are tested: (i) model 1, using the full IO-E for prediction (ii) model 2, with no dimensionality reduction and no clustering ($d = 6$, $Nc = 1$, $N_{\mathrm{R}} = 500$), (iii) model 3, with dimensionality reduction, but without clustering, ($d = 2$, $Nc = 1$, $N_{\mathrm{R}} = 500$), and (iv) model 4, with both dimensionality reduction and clustering ($d = 2$, $Nc = 2$, $N_{\mathrm{R}} = 500$).

Figure 5.13 displays the relative error (as a percentage) between the temperature T given by Mutation++ and the prediction of the data-driven model $\hat{T}$, for the four configurations enumerated above; contours of temperature in Kelvin are also added to highlight the evolution of the flow. First, the prediction of model 1 produces the highest error by a wide margin. This showcases the difficulty of properly training a neural network for prediction in high dimensions. The figure shows the maximum relative error in all three remaining models to be only a few percent, and located around the edge of the boundary layer, where the gradients are strongest. In addition, the error in model 2 is higher than the error of model 3, even though some information is lost in the latter due to the encoding step. This is a direct consequence

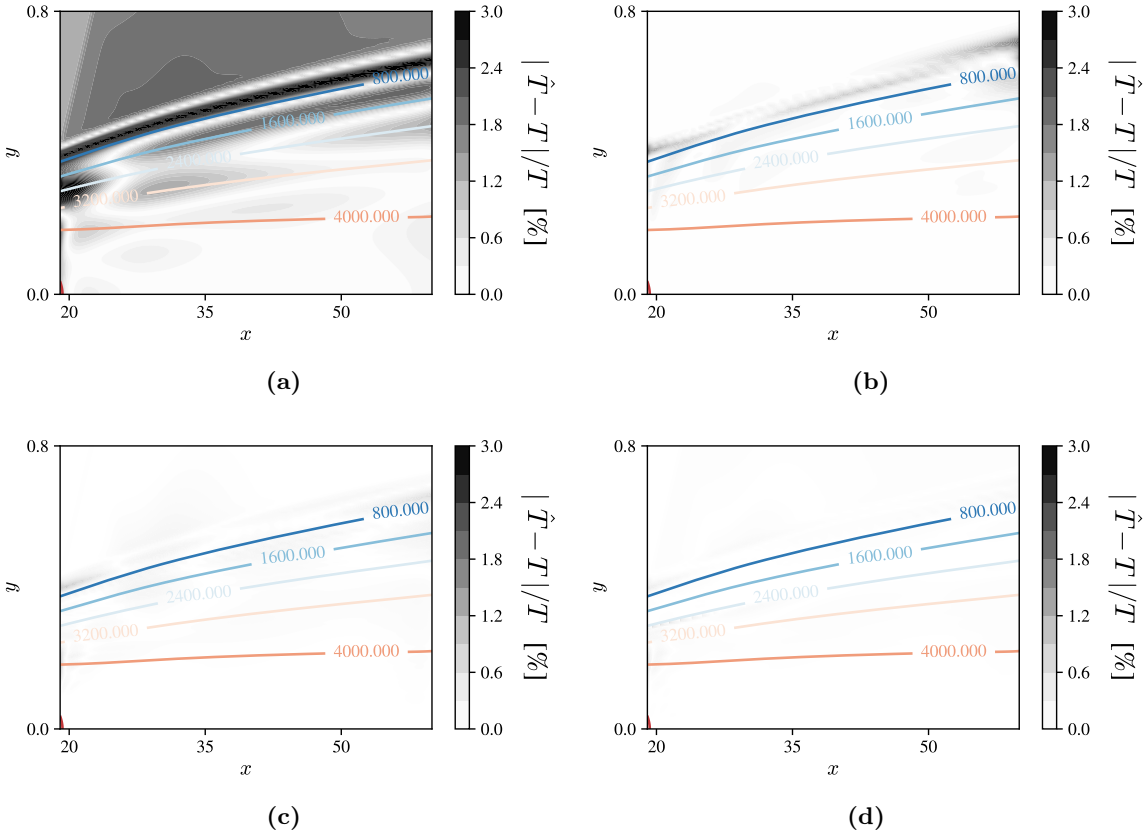


Figure 5.13: Comparison of the relative temperature error $|\hat{T} - T|/T$ in percent with contours of temperature. (a) Model 1: full IO-E (b) Model 2: $d = 6, N_c = 1, N_R = 500$ (c) Model 3: $d = 2, N_c = 1, N_R = 500$ (d) Model 4: $d = 2, N_c = 2, N_{\mathrm{R}} = 500$

of the curse of dimensionality: as the number of dimensions increases, the sampling volume in the input space increases exponentially. However, in this case, we kept the number of RBF centers N_R fixed. To get to the same level of accuracy, N_R should be increased in model 2, resulting in a performance loss. Finally, the error decreases slightly from model 3 to model 4. This improvement originates from the clustering step. In fact, each of the two clusters have $N_\mathrm{R} = 500$ centers. Hence, the input space is actually populated with $N_\mathrm{R} = 1000$ RBF centers in model 4. According to Figure 5.12, this improves the accuracy of the surrogate surface. The added number of centroids, however, does not result in a loss of performance. Assuming that a fraction α of the N_t query points are in cluster 1, then $1 - \alpha$ query points are in cluster 2. The evaluation time of the surrogate surface linked to cluster 1 then scales roughly as $\alpha d N_t N_\mathrm{R} C_{RBF}$. Similarly, for cluster 2, it scales as $(1 - \alpha) d N_t N_\mathrm{R} C_{RBF}$. Finally, the total evaluation time, i.e., the sum of the two, remains $d N_t N_\mathrm{R} C_{RBF}$. This can be easily generalized to a higher number of clusters.

These results demonstrate that the preprocessing steps involved in the construction of the model improve overall performance while maintaining a high level of accuracy.

5.4.1.5 Model stability

The resulting data-driven model (model 4, with all pre-processing steps) is coupled to the flow solver in a time-marching simulation. Starting from the solution obtained with Mutation++, the simulation is restarted using the reduced library only, also referred as a closed-loop prediction. After running for two flow-through times, the solution remains stable. The base-flow profiles are compared for various quantities of interest in Figure 5.14. Excellent agreement is found between the profiles. The only discrepancies observed are in the atomic oxygen profile in the freestream region with values in the range of 10^{-7}, which are not deemed important to the overall accuracy and stability of the model. This validates the accuracy and suitability of the data-driven model to simulate hypersonic flows in chemical non-equilibrium over the enthalpy range observed during the training step.

5.4.1.6 Model performance

To compare the performance of the data-driven model to the full library, we performed a scaling study.

Mutation++ is a serial library, hence its time complexity can be expressed as $O(C_{M++} N_t)$ where N_t is the number of independent, evaluated thermodynamic states, corresponding to each grid points in a numerical simulation for instance. Two variants of Mutation++ are considered here. The first one solves directly the Stefan-Maxwell diffusion problem and returns the diffusion velocity. In the second one, the diffusion coefficients D_s are returned and the diffusion fluxes are later computed using Eq. (1.21). See §1.2.1.3 for a more detailed discussion about the different modeling choice for multicomponent diffusion.

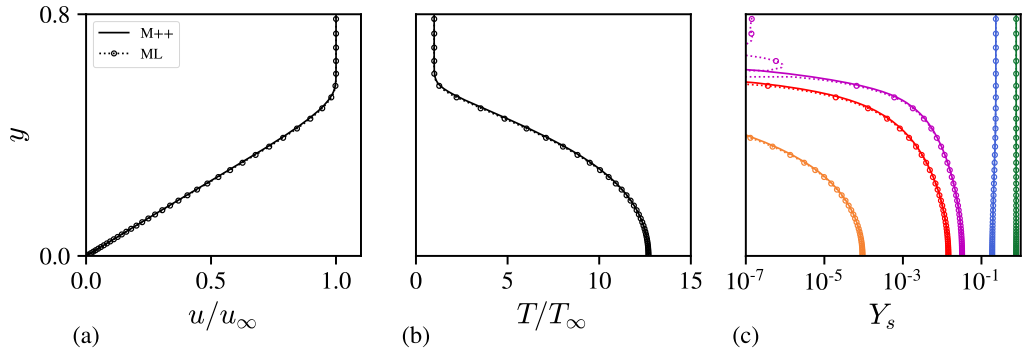


Figure 5.14: (a) Streamwise velocity, u, (b) temperature, T, and (c) species mass fraction, Y_s, profiles for case A at $R_x = 2,000$. The converged solution obtained with Mutation++ (solid lines) is compared to the solution restarted with the reduced-order thermochemical model (dotted lines with symbols) after 2 flow-through time. (c) From left to right : $\mathrm{N}, \mathrm{NO}, \mathrm{O}, \mathrm{O}_2, \mathrm{N}_2$

The diffusion fluxes are computed in the same fashion with the data-driven model. Moreover, we recall that its time complexity is $O(C_{ML}N_t)$, where $C_{ML} = O(HC_{ac}L + n_{\text{tree}}\text{depth} + N_{\text{R}}C_{RBF})$. For all three thermochemical models, the prefactor is empirically determined and shown in Figure 5.15.

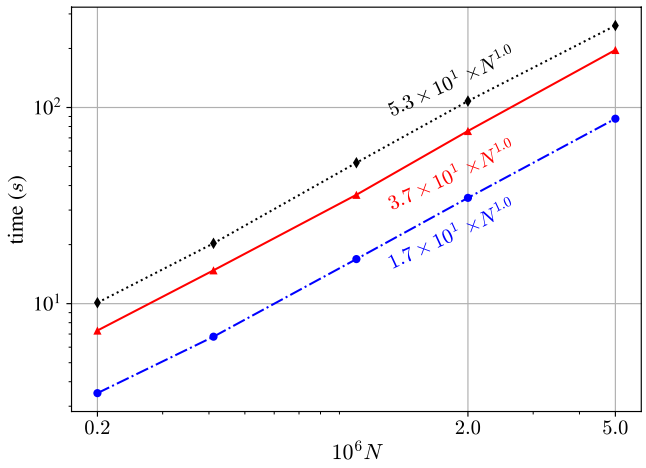


Figure 5.15: Comparison of the time complexity of Mutation++ with Stefan-Maxwell diffusion (dotted black line), the purely local version (solid red line) and the data-driven model (dash dotted blue line). The best non-linear least-squares fit of the form CN^α is added.

All curve fits, shown in Figure 5.15, suggest that, in practice, the models scale as $O(N_t)$ with exponents close to unity. The ratio of the prefactor is $C_{ML}/C_{M++} \approx 0.45$ when using Mutation++ in its local version. We can therefore expect a 55% CPU gain by using the data-driven model instead of Mutation++. In

fact, we assessed a CPU time reduction of 50% during the simulation, with a grid of size $N \approx 400,000$. This confirms a speedup through the use of a surrogate model. Secondly, the speed-up is even more significant when the library also solves for the Stefan-Maxwell diffusion problem at each grid point. Although not rigorously a one-to-one model comparison anymore, the data-driven model now performs 70% faster without any loss of accuracy. In fact, Fick's law based diffusion model have been shown to be highly accurate in hypersonic simulations. Moreover, fine tuning of the hyperparameters may allow even higher CPU gains as C_{ML} is proportional to a linear combination of the hyperparameters. Finally, we stress that the data-driven algorithm is a python implementation competing with a compiled C++ library. The speed-up reported here can be significantly increased by porting the model to a compiled language. It is also believed that the speed-up would be more significant when the dimensionality of the input space increases to include more chemical species.

5.4.2 Isothermal boundary layer at sub-orbital enthalpy

The second case chosen is the isothermal $Ma = 10$ boundary layer at suborbital enthalpy of Di Renzo and Urzay [Di Renzo & Urzay 2021]. Numerical details of this case were presented in the context of the locally self-similar solution in chemical nonequilibrium, §4.2.1. The application of a data-driven reduced order thermochemical model for this case was presented in [Scherding *et al.* 2022].

Aerodynamic heating produces temperatures of the order of $T_{max} \approx 4T_\infty$ at a wall-normal distance of $y = 0.8$. This temperature is sufficiently high to promote dissociation of O_2 into its atomic counterpart, as well as production of NO. However, it is too low to produce significant amounts of atomic N, as seen on Figures 4.1 and 5.17. Temperature then decreases towards the wall after the peak aerodynamic heating, due to wall cooling. Mass fractions of O and NO, on the other hand, stay relatively high, owing to the diffusion of radicals away from the wall and the lower convection speed. Hence, radical species are present in regions where they could not exist in case of chemical equilibrium. This demonstrates the strong coupling between hydrodynamics and chemistry and results in a more complex thermodynamic manifold. In fact, in the adiabatic case, temperature and concentrations of dissociation products increase monotonically with decreasing distance from the wall, resulting in a rather simplistic manifold of thermochemical states.

To train the data-driven surrogate model, $N = 10^5$ thermodynamic state vectors are sampled from the converged solution. The corresponding outputs of the library are collected simultaneously. The model obtained has the following specifications: $d = 2, N_c = 2$ and $N_\mathrm{R} = 250$. The resulting clusters are shown in Figure 5.16. Interestingly, the cooled near-wall region is assigned the same cluster as the freestream, despite the high-concentration of radical species. Indeed, Mutation++ outputs are more sensitive to internal energy rather than radical species concentration (Figure 5.2). Hence, during the encoding process, some

information about the radical concentration is lost and clusters rather demarcate ranges of density and internal energy.

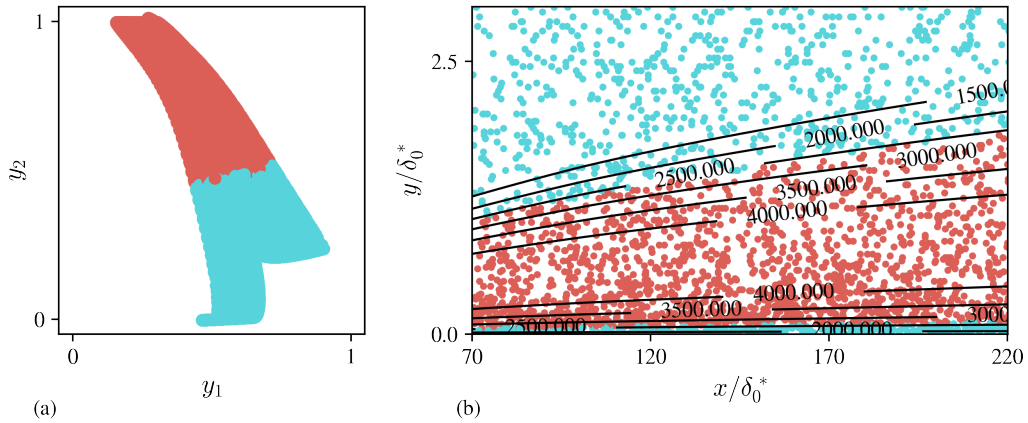


Figure 5.16: Training points $\mathbf{Y}$ colored by their cluster number. (a) In the latent space found by IO-E. (b) At the Cartesian location they were sampled from, with contours of temperature T in Kelvin.

In this case, the data-driven model performs 75% faster than the original library, drastically decreasing the time to solution of such calculations. The better performance is directly linked to a smaller number of RBF centers, $N_\mathrm{R} = 250$, compared to the adiabatic case where $N_\mathrm{R} = 500$ was a conservative choice. Restarting the simulation with the data-driven model (i.e, in a closed loop), the solution remains stable after four flow-through times using the reduced-order thermochemical model (Figure 5.17(a-c)).

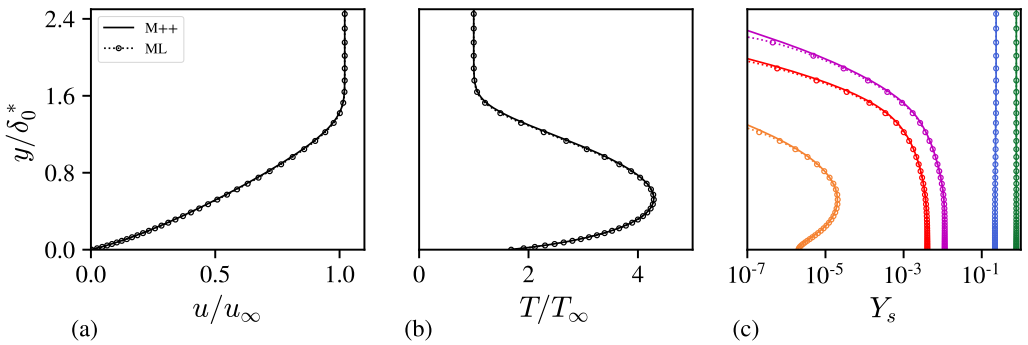


Figure 5.17: Profiles of (a) streamwise velocity, (b) temperature, (c) species mass fractions (from left to right: N, O, NO, N_2, O_2), and (d) temperature gradient at $R_x = 1000$. Solid lines and dotted lines with symbols correspond to the solutions generated using Mutation++ and the data-driven model, respectively.

5.4.3 SBLI with finite rate chemistry

Following the same steps, the model is trained on the $Ma = 5.92$ SBLI case with finite rate chemistry (see §3.2.2 for numerical details). The model has the following specifications: $d = 3$, $c = 3$, $N_{\mathrm{R}} = 500$. In this case, the dimensions of the latent space and the number of clusters are higher due to the more complex thermodynamic manifold learned by the IO-E. A two dimensional projection of the three dimensional manifold is presented in Figure 5.18(a). In this plane, distinct thermodynamic regions (i.e. different clusters) are wrapped around a scarcely populated center area. This can be explained by the impinging, recirculation and reflected shocks that induce abrupt change in the thermodynamic state. These regions and their borders become even more meaningful when reported in their physical location in the flow, as seen on Figure 5.18(b) where a numerical Schlieren is superposed. The green cluster corresponds to mildly hot conditions with high density, i.e. the freestream and post impinging shock conditions. After the recirculation shock, the thermodynamic states shift instantaneously to higher densities and temperatures, represented by the blue cluster. However, close to the apex of the recirculation bubble, the expansion fan decreases these thermodynamic variables, inducing a shift back to the green cluster. Finally, the red cluster, found in the boundary layer, corresponds to high temperatures and low densities. At the core of the recirculation bubble, temperature decreases and density increases, which brings the local state vector back to the green cluster.

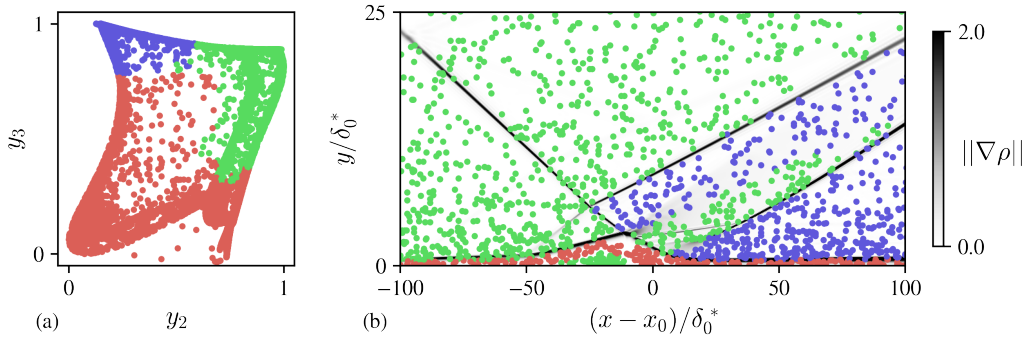


Figure 5.18: Training points $\mathbf{Y}$ colored by their cluster number. (a) In the latent space found by IO-E. (b) At the Cartesian location they were sampled from, with contours of magnitude of the density gradient $\|\nabla\rho\|$

In a closed-loop simulation, the model remains stable while maintaining a factor 2 speed-up in predicting thermochemical properties. In fact, Figure 5.19(a) and Figure 5.19(b) show that the wall pressure and skin friction remain in excellent agreement with the baseflow solution after 2 flow-through times. A slightly higher skin friction is observed after reattachment of the boundary layer, owing to a higher value of viscosity predicted by the model. Concerning the species, the only discrepancy with the initial solution is observed for atomic nitrogen mass concentration in Figure 5.19(c). However, it is present in such small quantities that it does not have an

impact on the stability of the solution. In fact, the thermochemical properties are not sensitive to small perturbations in atomic nitrogen concentration (cf. Figure 5.2), another fact motivating the dimensionality reduction performed in pre-processing.

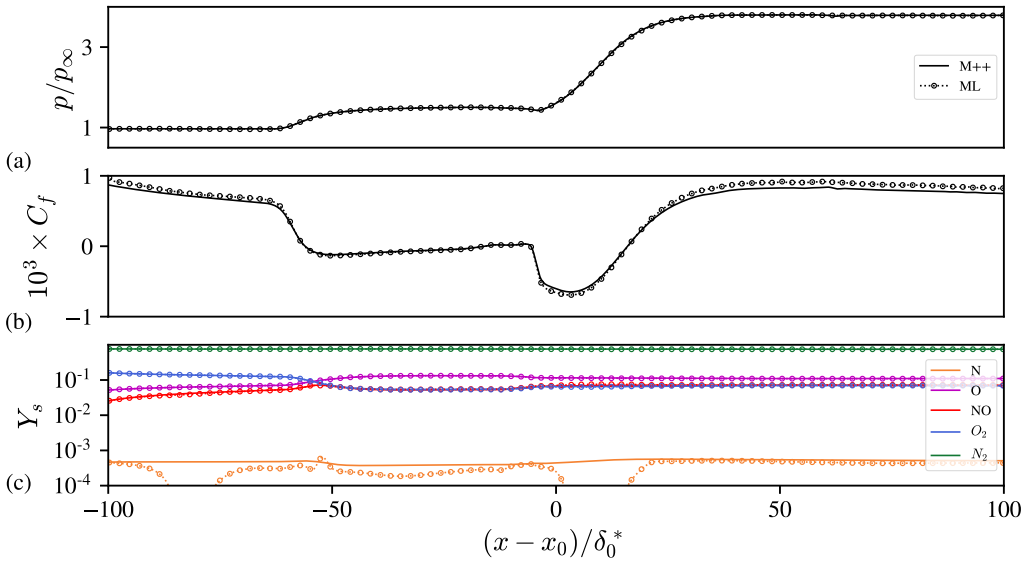


Figure 5.19: Comparison of wall-distribution of (a) skin-friction coefficient, (b) pressure, and (c) species mass fractions (from top to bottom on left side: N_2, $O2$, O, NO and N). Solid lines and dotted lines with symbols correspond to the solution obtained using Mutation++ and the data-driven model, respectively.

Chapter 6

Path to generalization

Contents

Up to this point, the reduced-order thermochemical models have exhibited the ability to preserve stable and consistent solutions, while enhancing the performance of evaluating physico-chemical properties. This outcome is unsurprising since the models have been trained on identical steady solutions without any alterations. However, hypersonic flows of interest can exhibit unsteady features such as hydrodynamic instabilities and turbulence. On one hand, turbulence enhances the average skin friction and heat-flux at the wall – two critical design parameters – compared to a laminar flow. On the other hand, hydrodynamic instabilities can cause even larger deviation from the mean flow. For example, Sayadi et al. [Sayadi *et al.* 2013] have shown that the instabilities present in a transitional boundary layer produce higher skin-friction and wall heat-flux than in the fully turbulent boundary layer.

High-order numerical methods are well suited to study such flows with their high accuracy and minimal modeling assumptions. However, the stringent requirement for performing direct numerical simulations of turbulent hypersonic flows and the additional cost incurred by the complex thermochemical nonequilibrium model have limited numerical studies to the lower Reynolds number regime and simple configurations [Di Renzo & Urzay 2021, Passiatore *et al.* 2022]. Thus, an optimized and tailored reduced-order thermochemical model could be utilized to study higher Reynolds number flows, for instance. Thus, the following question needs answering: (i) **can the model deal with unsteady hypersonic flows within its predictive capabilities ?**

The dynamic nature of unsteady flows introduces numerous thermodynamic states that cannot be fully predicted beforehand without a DNS. This poses a well-known issue for any reduced-order model [Huang & Duraisamy 2023] as they are likely to fail when applied in a parameter range not seen during training. One solution to increase the predictive capabilities of the model is to expand the range

of the training data by including data from many simulations using the full-order thermochemical model. However, this approach would severely increase the off-line training cost and counter its use for speeding up subsequent simulations, especially if a turbulent DNS with the full-order thermochemical model is needed.

To efficiently train the data-driven model, it is therefore important to obtain the training data at a low-computational cost, while being representative of the full flow system behavior. This raises the second question: (ii) **can we investigate techniques to approximate (at least partly) the unsteady thermodynamic manifold beforehand at a low computational cost?**

We first answer question (i) by considering simple unsteady regimes. Secondly, we consider using the steady DNS solution as the training set for an unsteady flow simulation, partially answering question (ii). Finally, we propose another training alternative, by leveraging the use of low-fidelity locally self-similar solutions in chemical nonequilibrium to generate the training set.

6.1 Extension for the simulation of unsteady hypersonic flows

Several issues arise by adding unsteadiness to the problem. First, pressure prediction becomes of paramount importance as the momentum equation contains the pressure gradient. Hence, any error in the pressure prediction would be enhanced by differentiation and could ultimately alter the flow dynamics. This problem was not encountered in the steady cases studied initially where pressure was mostly constant within the boundary layer (or a given thermodynamic cluster). Secondly, if the predicted pressure and temperature are not thermodynamically consistent, thermodynamic instability can be triggered, as demonstrated in [Pini *et al.* 2015].

The adaptation of the methodology to accurately compute the dynamics of unsteady hypersonics flows in chemical nonequilibrium is first detailed. Two cases are then considered for testing: (i) the adiabatic Mach 10 laminar boundary layer optimally forced with two-dimensional disturbances through a blowing-suction strip, as described in §3.1.3.2, and (ii) the thermochemical manifold is made more complex with a roughness-induced shock and expansion while keeping the optimal forcing of the laminar boundary layer (case R §3.1.3.3).

6.1.1 Modification to the algorithm

6.1.1.1 Pressure-advantaged dimensionality reduction

The goal of the dimensionality-reduction step is to find a lower dimensional subspace where all outputs have a smooth variation with respect to this new set of coordinates. The shape of the manifold as it unfolds (for example, existence of non-unique values in a given region of the low-dimensional subspace [Zdybał *et al.* 2022]), will affect the predictions of the RBF. Moreover, different outputs may lead to different shapes of the lower-dimensional manifold. For example, in our case of interest, kinetic terms

are active (non-zero) only in a small region of the thermodynamic input space while temperature displays variations over the entire explored space. During the training of the IO-E, all outputs of the library in the vector $\mathbf{z} \in \mathbb{R}^{D_Z}$ are scaled between 0 and 1 with a minimum-maximum scaler. Thus, they all have the same weight, and the unfolding becomes a trade-off between all the outputs, as the loss is based on the L_2 norm of the error $\|\tilde{\mathbf{z}} - \mathbf{z}\|_2$, where $\tilde{\mathbf{z}}$ denotes the prediction of the decoder.

This approach worked for steady cases but when considering unsteady flows where dynamics of the flow are governed by the momentum equation (Eq. 1.3), in which the pressure P directly appears through its gradient, any significant error in the pressure prediction would be enhanced through numerical differentiation, and the dynamics would be altered. Since our goal is to reproduce faithfully the dynamics of the boundary layer, it is therefore desirable to bias the dimensionality reduction step towards a better pressure unfolding. To this end, the scaled pressure value is multiplied by a scalar $\alpha > 1$ during the training of the IO-E. Hence, $P \in [0, \alpha]$ while other outputs lie in the $[0, 1]$ interval. The consequence is that more weight will be put on the pressure to reduce the IO-E loss, resulting in a smoother variation of pressure values on the low-dimensional manifold. This is conveyed in Fig. 6.1 where the low-dimensional manifold, colored by pressure, is presented for $\alpha = 1$ and $\alpha = 5$.

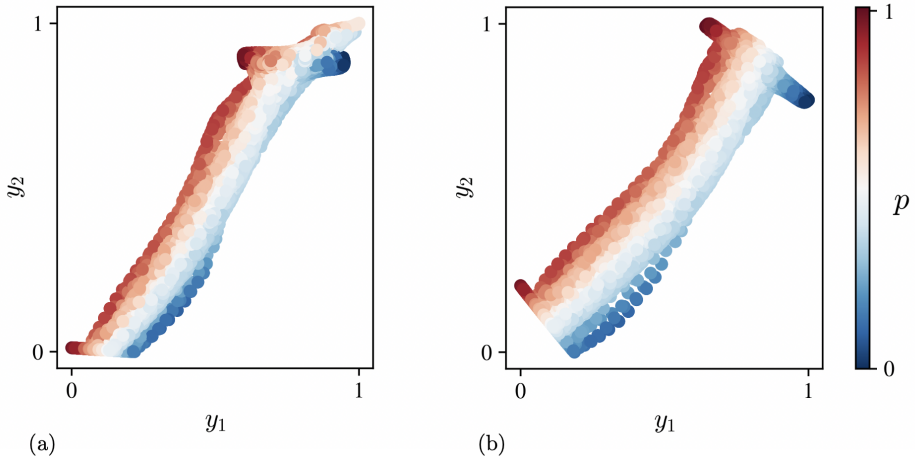


Figure 6.1: Low-dimensional representation $Y \in \mathbb{R}^{N \times d}$ $(d = 2)$ of $X \in \mathbb{R}^{N \times D}$, colored by pressure p. (y_1, y_2) denote the coordinates of the latent space. (a) $\alpha = 1$ (b) $\alpha = 5$. The same network architecture, random seed, training set and number of epochs were used during training.

6.1.1.2 Physics-based prediction of temperature

In order to maintain thermodynamic consistency, temperature is evaluated as in [Scherding *et al.* 2022], and is computed from the pressure prediction of the model $\hat{p}$ through the equation of state (Eq. (1.10)), which was shown to

improve the reconstruction of the temperature gradient during the simulation. The combination of pressure-advantaged dimensionality reduction and the reconstruction of temperature using the equation of state results in a highly accurate, stable and thermodynamically consistent prediction of the two primitive variables p and T.

6.1.2 Application

6.1.2.1 case A

The first unsteady test case is the adiabatic Mach 10 laminar boundary layer optimally forced with two-dimensional disturbances through a blowing-suction strip, as described in §3.1.3.2. An amplitude of perturbation of $A/u_\infty = 2 \times 10^{-2}$ is used.

The data-driven model is trained by sampling inputs and outputs from a time-periodic solution, containing perturbations. The hyper-parameters of the models are as follows: $d = 3$, $N_C = 2$ and $N_\mathrm{R} = 400$. They are determined by following the same metrics presented in §5.4.1. A parameter $\alpha = 2$ was used for the pressure unfolding. Note that the dimension of the latent space ($d = 3$) is higher compared to that of the steady case ($d = 2$). In fact, the perturbations add complexity to the thermodynamic manifold, and an additional dimension is necessary to properly unfold it to a lower-dimensional subspace. Even though the latent space has a higher dimension, the two clusters determined by Newman's algorithm are consistent with the one found in the steady case, see Figure 5.11. They represent the cold, frozen freestream and the hot and chemically reactive near-wall region, respectively. With this set of hyper-parameters, the model performs 60% faster than the original Mutation++ library.

Starting from the time-periodic solution that provided the training samples, the simulation is restarted for one forcing period with the true model (e.g. Mutation++ library) and the data-driven model. The RMS wall pressure and growth rates obtained are displayed in Figure 6.2. The agreement for the RMS wall-pressure signals are perfect. For the growth rates, the results of the data-driven model follow the same trend as the true solution but the curves are noisier. This noisiness is a direct consequence of the growth rate calculation via differentiation (Eq. 3.10). The data-driven model induces slight deviation in the envelope of the perturbation that is enhanced through differentiation. However, the error in the growth rate is no more than 5% across the streamwise domain.

Overall, the data-driven model provides a very good estimate of the dynamics compared to the true solution. This is expected as the perturbed states were already included in the training. Hence, over the forcing period, the model worked mainly in interpolation mode with high accuracy.

6.1.2.2 case R

Secondly, case R (§3.1.3.3) is chosen to showcase that the data-driven reduced-order thermochemical model can also deal with simulations of unsteady hypersonic flows

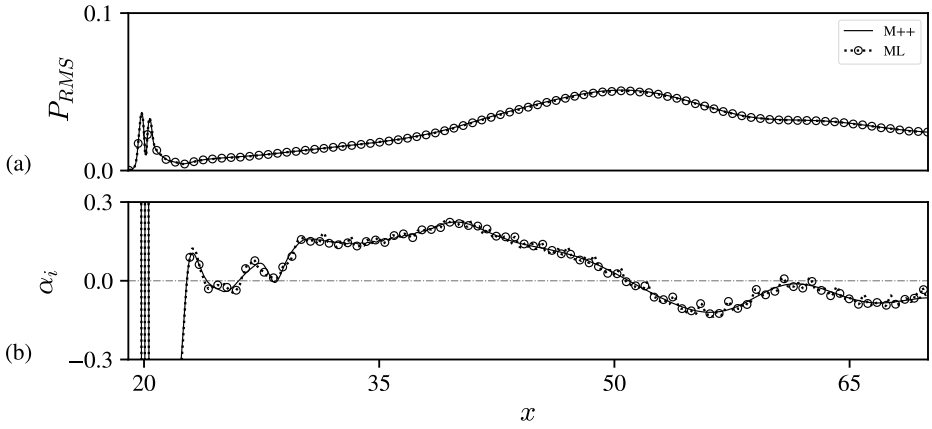


Figure 6.2: Evolution of (a) RMS wall pressure, and (b) growth rate α_i, computed using Eq. (3.9), as a function of streamwise position x for case A. Solid lines correspond to the baseline solution (Mutation++). Dotted lines with circles correspond to the result of the data-driven model.

in chemical non-equilibrium with characteristic compressible flow features: shocks and expansion fans.

Similar to the flat-plate case, the reduced-order thermochemical models is trained by sampling input-output pairs from a snapshot of the unsteady run. The following specifications are retained: $d = 3$, $\alpha = 5$, $N_C = 2$ and $N_{\mathrm{R}} = 400$. With this set of parameters, the data-driven model evaluates thermochemical properties 60% faster than the Mutation++ library. The simulation is then restarted using the data-driven model for one forcing period. The agreement between the RMS pressure signals at the wall is perfect (Fig. 6.3a). Concerning the growth rates (Fig. 6.3b), discrepancies are observed in the zone of influence of the roughness (between $x = 25$ and $x = 35$). In fact, the zone of influence of the roughness, where the shock and expansion fan interact with the two-dimensional disturbance, is relatively narrow compared to the full numerical domain. Subsequently, the corresponding thermodynamic states are proportionally less represented in the training set. Once the instabilities return to the growth pattern of the boundary layer, the agreement with Mutation++ is nearly perfect. Hence, there is no global influence of the error. This demonstrates the accuracy and robustness of the model in simulating unsteady hypersonic flows in chemical non-equilibrium with shocks and expansions.

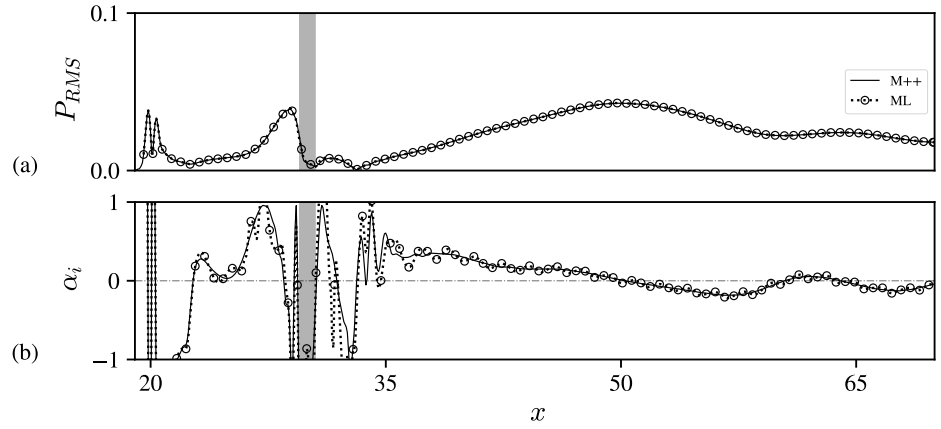


Figure 6.3: Evolution of (a) RMS wall pressure, and (b) growth rate α_i as a function of streamwise position x for case R. Solid lines correspond to the baseline solution (Mutation++). Dotted lines with circles corresponds to the results of the data-driven thermochemical model. The grey vertical band represents the extent of the roughness.

6.2 Generating training data efficiently

Now that we have addressed the initial question posed at the start of this chapter, we can conclude that if a data-driven model has been exposed to every conceivable state found within a specific configuration, it possesses the capability to faithfully replicate the dynamics and quantities of interest. Moreover, the utilization of the data-driven reduced order thermochemical model significantly reduced the performance disparity between equivalent simulations using the CNEQ and perfect-gas thermochemical model. In fact, simulations in CNEQ with Mutation++ and the Stefan-Maxwell multicomponent diffusion model are up to 5.5 times more expensive than the equivalent simulations with a perfect gas model, see Figure 2.7. This gap is decreased to a factor 3 with the simplified diffusion model (Eq. (1.21)). With the data-driven surrogate model, this value is brought down to 1.2 approximately. Moreover, its lower bound is theoretically 0.8 with an infinitely fast thermochemical model. This hard limit is due to the additional $N_s - 1 = 4$ additional species conservation equations (Eqs. (1.2)) that have to be solved alongside the five original equations (Eqs. (1.1,1.3,1.4)) for simulations in CNEQ.

However, a complete training set is not always readily available for more complex unsteady hypersonic flow simulations. Providing training data directly from a first unsteady simulation with the full thermochemical model would represent a sub-optimal approach and hinders its use in speeding up the calculation. It is therefore of interest to investigate techniques to approximate (at least partly) the unsteady thermodynamic manifold beforehand from a steady calculation.

6.2.1 Artificially augmented training set

In the following, two different training sets $\mathbf{X}_1 \in \mathbb{R}^{N\times D}$ and $\mathbf{X}_2 \in \mathbb{R}^{N\times D}$ with $N = 3 \times 10^5$ training points are considered. In the first set, thermodynamic states are sampled from the converged steady-state solution of case A, i.e. the base flow without any perturbation, as in §5.4.1. The second set is sampled from a single flow snapshot taken during one oscillation period and contains perturbed states, as in §6.1.2.1.

In order to verify whether the addition of perturbations yields new thermodynamic states, we compare the distribution of the library outputs with respect to the input thermodynamic state in the different training sets. To this end, we use kernel density estimation (KDE) [Rosenblatt 1956]. First, the inputs and outputs of either training dataset $\mathbf{X}_j$ and $\mathbf{Z}_j,\ j \in \{1, 2\}$ are concatenated into new vectors $\mathbf{m}_i, i = 1, ..., N \in \mathbb{R}^{D+D_Z}$ and assumed to be independent and identically distributed samples drawn from the same multivariate distribution f^m. The kernel density estimator of f^m is derived as

$$\hat{f}_h^m(\mathbf{m}) = \frac{1}{hN} \sum_{i=1}^{N} K\left(\frac{\|\mathbf{m} - \mathbf{m}_i\|}{h}\right), \tag{6.1}$$

where $K(r) = e^{-r^2}$ denotes the Gaussian kernel and h is the bandwidth. The latter is estimated according to [Scott 2015] as $h = N^{-1/(D+4)}$. This results in $(D + D_Z)^2$ bivariate distributions that can be plotted to track the distribution of any output with respect to the different component of the input thermodynamic state vector.

In Figure 6.4, the bivariate distributions of pressure p and temperature T against all inputs of the library are plotted. Black lines represent the training set sampled on the steady solution, while blue lines represent the training set containing perturbations. The distributions of temperature cover mostly the same area, with slightly higher temperature values found in the blue training set. However, wide differences are observed for pressure distributions. Indeed, pressure is nearly constant in a boundary layer. Hence, the baseflow distribution is mostly constant around the freestream value of $p_\infty = 3596$ Pa, with variations of no more than 3% with respect to all inputs. However, in the perturbed solution (blue lines), pressure variations are up to 12% in amplitude with respect to the freestream value. A model trained on the baseflow solution would not be able to predict accurately this pressure range. Since the momentum equations contain the pressure force term, the dynamics would be altered if pressure is not correctly computed by the data-driven model.

Since the changes are not so significant, the training set can be artificially augmented in order to increase the domain of predictability of the model trained on the base-flow solution. To this end, M random thermodynamic states are sampled from the input training set. To each of these randomly selected points $\mathbf{x}_i$, additive

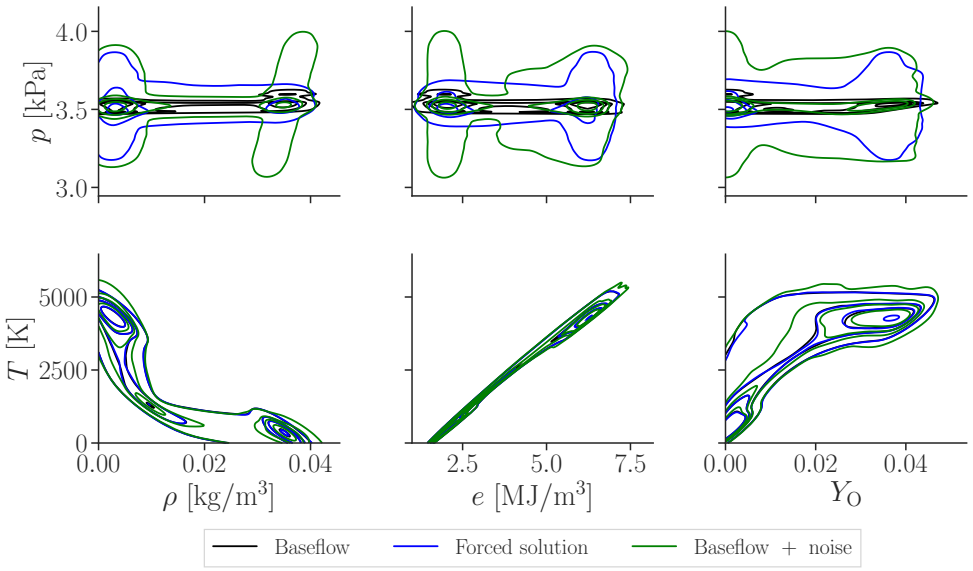


Figure 6.4: Pairwise kernel density of pressure P and temperature T with respect to the thermodynamic state vector components ρ, e, Y_N,Y_O, and Y_{NO}. Black lines represent the baseflow training set; blue lines represent the training set with perturbations included and an isothermal wall; green lines represent the baseflow training set, artificially augmented. Extreme contour lines represent a probability of 1%.

white noise is added in all input directions, generating new points $\mathbf{x}_i^r$,

$$\mathbf{x}_i^r = \mathbf{x}_i \odot (1 + \mathbf{A} \odot \mathbf{R}) \tag{6.2}$$

where $\mathbf{A} = [0.05, 0.2, 0, 0, 0, 0]^T \in \mathbb{R}^D$ is an input dependent amplitude vector, $\mathbf{R} \in \mathbb{R}^D$ is a random vector drawn from a uniform distribution in the interval $[-1, 1]$, and $\odot$ denotes the Hadamard (element-wise) product. This process is repeated several times ($N_{rand} = O(10^2)$) for each sampled point. This results in a set $\mathbf{X}^r \in \mathbb{R}^{MN_{rand} \times D}$ of artificial inputs. The corresponding outputs are obtained by calling the library, $\mathbf{Z}^r = f(\mathbf{X}^r)$. Looking at the artificially augmented distribution in Fig. 6.4 (green lines), the pressure range of the perturbed solution is mostly recovered. One can therefore expect the model to work well in simulating the growth of second-mode waves in the boundary layer.

6.2.1.1 Application

Using the green training set, a model is trained using the same specifications as in §5.4.1. Starting from the time-periodic solution that provided the training samples of the blue training set, the simulation is restarted for one forcing period with the new data-driven model. The resulting RMS wall-pressure signals and growth rates are shown in Figure 6.5, (triangular markers). The agreement between the RMS

wall pressure remains almost perfect. However, the error on the growth rate is now higher with this model than the one trained with full-state information (see Figure 6.2), with a maximum error of 10% at $x = 40$. Nevertheless, it still describes accurately the trend inside the boundary layer. These oscillations are again due to (now larger) modulations of the envelope of the wall-normal RMS velocities that are enhanced through differentiation. This higher error can be explained by the fact that some information is lacking in the artificially augmented base-flow training set. For example, looking at the density-pressure bivariate distribution in Figure 6.4, the green distribution (artificially augmented base-flow) does not fully recover the pressure range of the blue distribution (solution with instabilities) around $\rho = 0.02$ [kg/m^3]. Hence, pressure will be extrapolated in this domain, ultimately altering the dynamics.

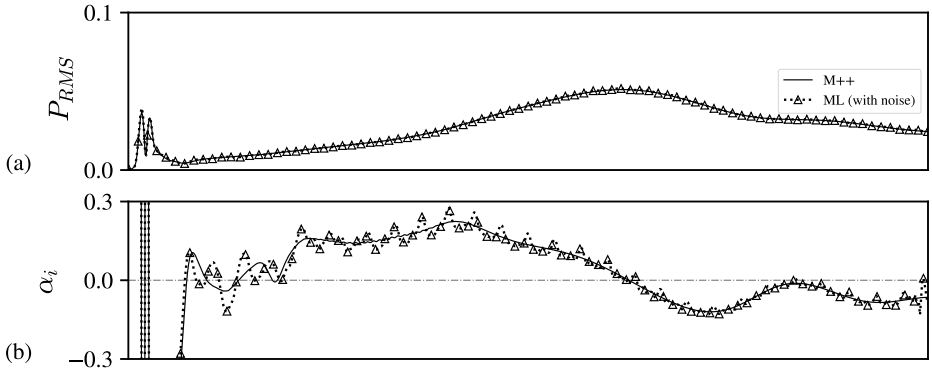


Figure 6.5: Evolution of (a) RMS wall pressure, and (b) growth rate α_i, computed using Eq. (3.9), as a function of streamwise position x for case A. Solid lines correspond to the baseline solution (Mutation++). Dotted lines with triangles correspond to the result of the data-driven model, trained on the jittered baseflow.

Separately, as a last test, a model was trained on the baseflow solution without artificial states (black distribution in Fig. 6.4). The dynamics could not be recovered as the simulation became unstable and quickly diverged. While not perfect, artificially augmented base-flow provides an easy solution to partially recover the flow dynamics and recover stability in the model.

Fine tuning the artificial points could result in a better recovery of the full pressure range of the unsteady flow, yielding a more accurate growth-rate prediction. This would require extensive knowledge about the unsteady features and their numerical range. It is however not clear how much the thermodynamic manifold would change in a case of a complex turbulent flows, for instance.

On the other hand, a brute-force inclusion of artificial points with higher standard deviations from their mean (higher-amplitude noise) would prevent the training of a tailored model. For instance, in the dimensionality-reduction step, the high-to-low dimensional mapping could become too challenging to learn, and the model would be under-fitting.

Thus, while attractive, this methodology should rather be considered as a preprocessing step on the training set to improve robustness and generalizability of the model. Indeed, injecting artificial noise (jittering) into the training set is a common technique in neural network training to improve robustness and reduce overfitting, as it is equivalent to a Tikhonov regularization [Bishop 1995, An 1996].

Therefore, a more prudent strategy would be to derive reduced-order model that can actively learn the new states on-the-fly during the simulation, which will be studied in the next chapter. This improved strategy will help tackle more challenging flows while keeping a model tailored to the specific case.

6.2.2 Using self-similar solution for training

In the previous section, the training set still came from a costly high-fidelity solution. We propose here to leverage locally self-similar solutions in chemical non-equilibrium to generate the training dataset, eliminating the previous computational bottleneck associated with DNS (see Chapters 4 for a description of these solutions). We consider here the conditions of the $Ma = 10$ isothermal boundary layer at surborbital enthalpy of Di Renzo and Urzay [Di Renzo & Urzay 2021], §4.2.1. This case was successfully simulated using a reduced-order thermochemical model in §5.4.2 with a model trained with full-state knowledge of the DNS.

In §4.2.1, Figure 4.1 displays a comparison of the DNS and self-similar profiles at $R_x = 1,000$. Good agreement is observed for the velocity, temperature, and mass fraction profiles. The most notable difference is in the species mass fractions, where higher dissociation is observed in the self-similar solution than in the DNS. This can be explained by the streamwise species diffusion flux that is neglected in the locally self-similar solution. The self-similar solutions were generated using 0.5 core hours on the Yellowstone cluster, compared with 1,500 core hours needed to obtain the converged laminar DNS.

Locally self-similar profiles are then interpolated on the numerical grid of the DNS, and input/output pairs are sampled. To assess the viability of using the self-similar solutions as a training basis for the thermochemical model, we compare the distribution of the input thermodynamic states from the DNS with that of the self-similar solutions, using again kernel density estimation (KDE), Eq. (6.1).

Figure 6.6 shows the pairwise KDE plots of the selected thermodynamic state vector components obtained when sampling points from either the laminar DNS (blue) or the self-similar solutions (black). Considering the diagonal plots, both distributions are bimodal along each component of the thermodynamic state vector. This bimodality comes from the freestream and the peak aerodynamic heating regions. Very good agreement is found for density (Figure 6.6(a)). A slight deviation in the high energy e and high radical's partial density (e.g. ρ_O) is observed. The latter is a direct consequence of the higher level of dissociated species shown in Figure 4.1. Since temperature distributions are almost identical (Figure 4.1 (b)), the internal energy increases (Eq. (1.11)), due to higher specific enthalpies of atomic species compared with their molecular counterpart. Looking at the bimodal

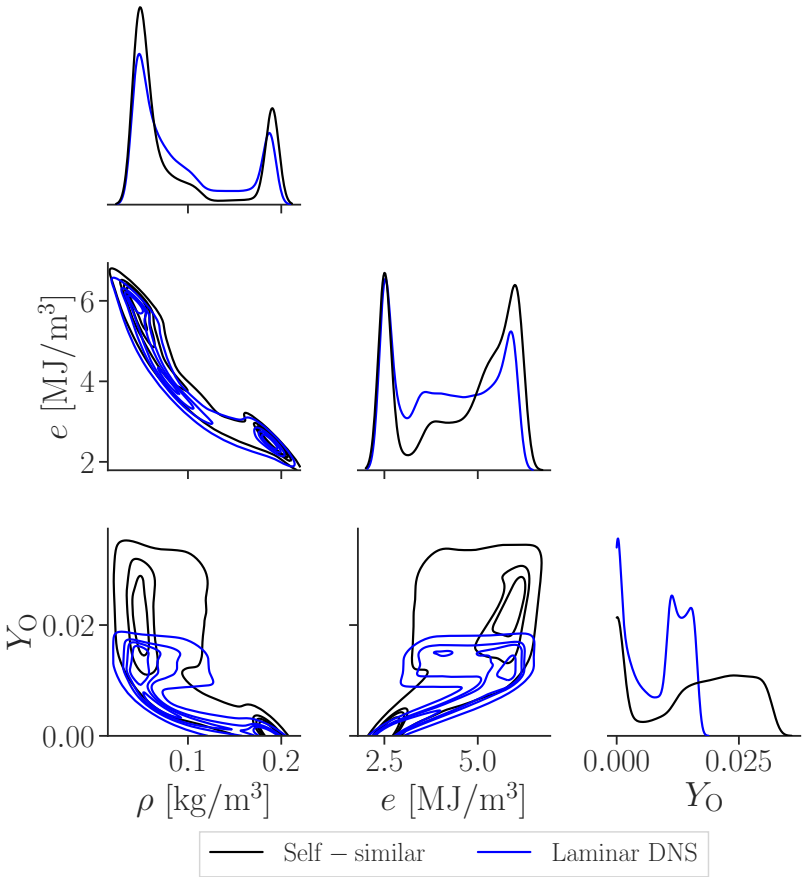


Figure 6.6: Pairwise kernel density of the selected thermodynamic state vector components ρ, e and ρ_O: blue lines represent laminar DNS; black lines represents self-similar solutions. Extreme contour lines represent a probability of 1%.

distribution (Figure 6.6(b,d,e)), the self-similar distribution (blue) extends to higher e and ρ_O.

Overall, the thermodynamic distribution spanned by the laminar flow is mostly contained in the region spanned by the locally self-similar solutions, resulting in reasonable accuracy of the surrogate model. However, a minority of states not seen during training would require extrapolation of the model in a time-marching simulation, which could be detrimental to the convergence and accuracy of the converged solution. Hence, this emphasizes again the need for active-learning capability to obtain truly predictive reduced-order thermochemical gas models.

6.2.2.1 Extension to turbulent DNS

Transition to turbulence is a critical consideration in the design of successful aerospace missions in the hypersonic regime. In fact, skin friction and heat flux at the wall are enhanced in a turbulent boundary layer compared with a laminar one. However, the DNS of turbulent boundary layers requires billions of degrees of freedom [Di Renzo & Urzay 2021, Passiatore *et al.* 2022]. The additional computational cost incurred by a complex thermochemical model could be better utilized to study higher Reynolds number hypersonic reacting flows, for instance. Hence, an efficient data-driven thermochemical model would be highly beneficial in these scenarios. Providing training data directly from the turbulent simulations, however, would represent a suboptimal approach. It is thus prudent to investigate the possibility of approximating the manifold (at least partly) using the laminar self-similar solutions in chemical non-equilibrium.

To this end, we consider a fully turbulent section of the boundary layer simulated by [Di Renzo & Urzay 2021]. The velocity and temperature fields on a streamwise slice (at half the spanwise extent) are plotted in Figure 6.7(a,b). We see that higher temperatures are present within the boundary layer compared with the laminar solution, partly due to the presence of shocklets within the boundary layer, increasing local temperatures.

N thermodynamic states are sampled out of the domain and an approximation of the probability density function is obtained with KDE (Eq. (6.1)). We compare the distribution from the self-similar (black) and turbulent (red) solution in Figure 6.8. The turbulent boundary layer presents a bimodal distribution along each component of the thermodynamic state vector except energy. In fact, the high-energy peak disappears in the turbulent boundary layer due to sweeps and injection of cold air in the peak heating region. Higher energy levels are also observed due to the presence of shocklets. The most notable difference is a shift of the freestream density peak to lower densities. This is the result of a succession of pressure waves emanating from the boundary layer to the freestream as it transitions to turbulence. The self-similar manifold predominantly covers the thermodynamic states seen in the turbulent boundary layer section considered. These results also suggest that locally self-similar solutions could be used to warm-start the training of a data-driven model adapted to hypersonic turbulent boundary layer simulations with finite-rate

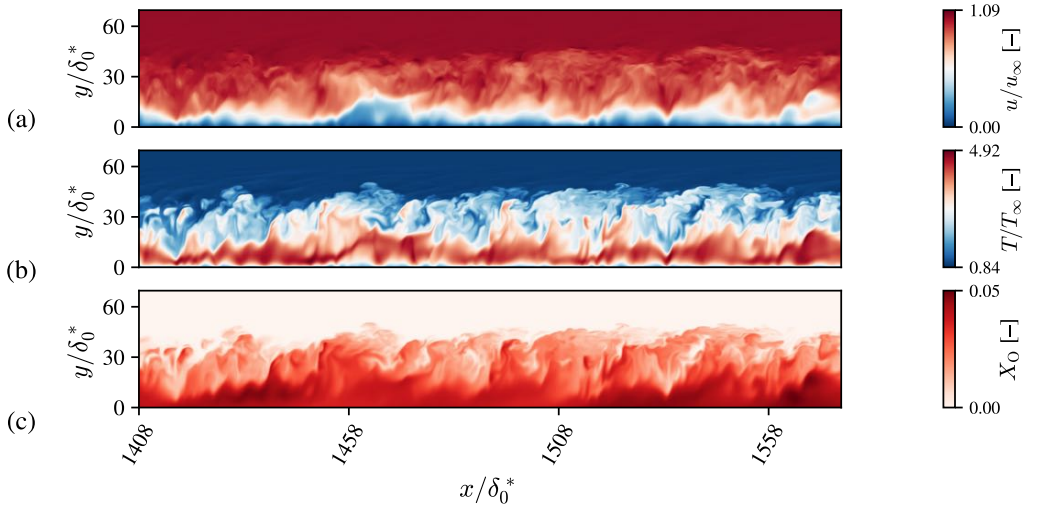


Figure 6.7: Contours of normalized (a) streamwise velocity, (b) temperature, and (c) atomic oxygen mole fraction at $z = 10\pi\delta_0^*$, in the turbulent cooled-wall Mach 10 boundary layer [Di Renzo & Urzay 2021].

chemistry. In such a scenario, if the model possesses the ability to dynamically learn and incorporate new thermodynamic states in real-time, its predictive capacity could enable turbulent simulations of hypersonic flows in chemical nonequilibrium at a significantly reduced computational cost.

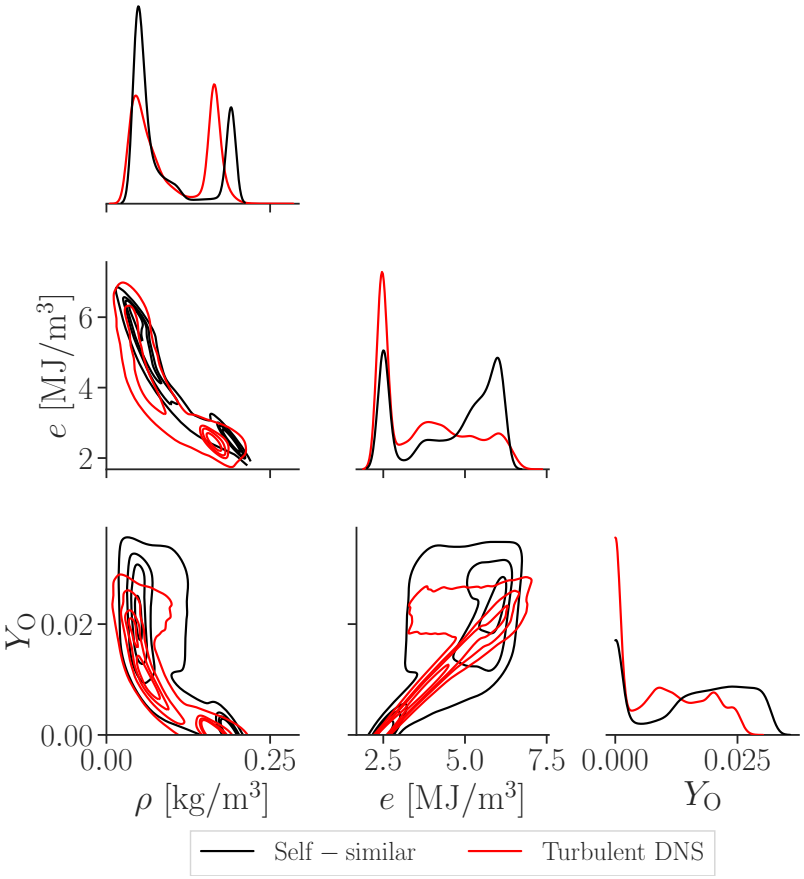


Figure 6.8: Pairwise kernel density of selected thermodynamic state vector components ρ, e and ρ_O: blue lines represent self-similar solutions; red lines represent turbulent DNS. Extreme contour lines represent a probability of 1%.

Chapter 7

Active learning for reduced-order thermochemical model

Contents

7.1 Motivation

The results of the last chapter have proven that reduced-order thermochemical models need an on-the-fly learning procedure to be truly predictive and efficient in speeding up the simulations of hypersonic flows in chemical non-equilibrium.

In fact, to achieve optimal performance, it is crucial to efficiently train the data-driven model in the first place. One approach to accomplish this is by warm-starting the training process using representative thermodynamic data that are more affordable to obtain compared to DNS. These data could be derived from a self-similar solution, which provides a reasonable approximation of the system's behavior, as seen in §6.2.2. This warm-starting process provides a starting point for the model's training, enabling it to capture essential features and relationships within the data.

However, during real-time usage of the model, new flow features may arise, such as perturbations or changes in parameters. For example, a model trained on a self-similar solution will cover most of the thermodynamic states encountered in the high-fidelity simulation. However, a minority of state will still require extrapolation of the model, as seen in §6.2.1.1, Figures 6.6 and 6.8. Secondly, in §6.2.1.1, a model trained on a high-fidelity steady simulation did not recover the dynamics of the forced flow (Figure 6.5) as faithfully as the model trained on the forced solution directly (Figure 6.2), even though artificial noise was included in the training to extend the range of applicability of the model. In both cases, these additions result in new thermodynamic states that were not present in the initial training set.

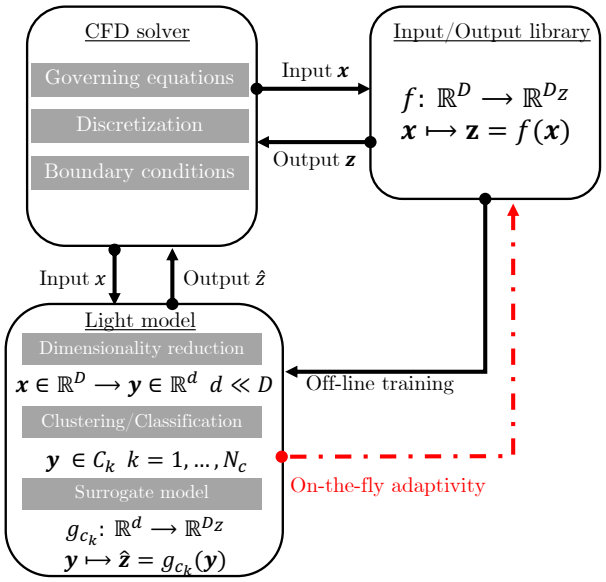


Figure 7.1: General schematic of the model training and coupling to replace any expensive Input/Output library.

In such cases, the data-driven model needs to adapt to these new states to provide accurate predictions. To address this issue, a continuous learning framework has to be implemented. This framework allows the data-driven model to dynamically update and adapt its internal representation based on new incoming data. This way, as new flow features or changes occur, the model can adjust its predictions accordingly, incorporating the updated thermodynamic states into its learned patterns.

By combining a warm-start training with a continuous learning framework, the data-driven model can effectively handle both the initial training phase with limited data and the adaptation to new thermodynamic states during real-time usage. This approach enhances the model's versatility and ensures its accuracy even in the presence of evolving flow features. More generally, adaptive reduced-order models are a promising solution to improve the predicition capability of reduced-order model. Therefore, this subject has gained attention in the scientific community recently [Peherstorfer 2020, Yano *et al.* 2021, Ramezanian *et al.* 2021, Huang & Duraisamy 2023].

In this chapter, we first define the methodology to enable active learning capability for the reduced-order thermochemical gas model. It corresponds to the parts highlighted in red in the schematic of the data-driven model in Figure 7.1. Secondly, this novel method is tested in two time marching simulations that generate thermodynamic states unseen during training. Namely, a low-fidelity (self-similar) to high-fidelity DNS transient simulation and an optimally disturbed laminar boundary layer.

7.2 Methodology

In this section, we describe the methodology for on-the-fly active learning of the model during a time-marching simulation. The strategy is divided in three steps:

- **§7.2.1. Extrapolation detection**: We first define a metric to detect when the model is extrapolating.
- **§7.2.2.1. Online k-means**: Secondly, new centers are added to the RBF thanks to an online clustering procedure.
- **§7.2.2.2. Update procedure of the RBF**: Finally, the RBF is efficiently updated by taking into account these new centers.

The different steps are showcased using case A: the model is trained on the self-similar solution, the black distribution in Figure 6.6, and tested directly on the converged laminar DNS, the blue distribution in Figure 6.6.

During a simulation, the user sets an updating frequency f_{up} (or a number of iterations) at which all the steps described below are performed and the model is updated.

7.2.1 Extrapolation detection

To overcome the generalization problem, one should first describe the region of the embedded space where training data is available. In fact, a correct characterization of this region would allow us to identify the regions where the outputs of the RBF are not reliable, which would subsequently signal the need for retraining.

One way to detect the borders of the space spanned by the training data is to use an α-shape of this set. Briefly, an α-shape is a generalization of the convex-hull of a set of points and were introduced by Edelsbrunner et al. [Edelsbrunner *et al.* 1983]. However, the hyperparameter α is difficult to tune, and the method lacks generalization in higher dimensions. Leonard et al. [Leonard *et al.* 1992] instead estimated the local density of training data using kernel density estimation (KDE). A low density of probability indicates possible extrapolation. An even simpler method to detect extrapolation in an online procedure was proposed by Lohninger [Lohninger 1993] and is based on RBF with gaussian kernels. In fact, evaluation of the RBF is based on the distance of the evaluation point $\mathbf{x}^t$ to the centers $\mathbf{X}^c = [\mathbf{x}^c_1, \dots, \mathbf{x}^c_{N_\text{R}}]$, of the RBF (Eq. (5.24)). The difference of the maximum of the activation functions to 1 can then be used as a parameter to flag extrapolation

$$f_e(\mathbf{x}^t) = 1 - \max_{\mathbf{x}^c \in \mathbf{X}^c} \phi(\|\mathbf{x}^t - \mathbf{x}^c\|). \tag{7.1}$$

Using the above relation, if the evaluation point is geometrically close to a center, the second term on the right-hand side tends to 1 and f_e tends to 0. However, if the evaluation point lies far from any center, f_e tends to 1, indicating extrapolation. A threshold on the value of f_e is then used to distinguish between extrapolation and

interpolation regions, respectively. However, a drawback of this method is that it relies on monotonic kernel functions, which is not the case of the thin-plate spline kernels, (Eq. (5.19)), used here.

Hence, we propose an alternative approach based on the minimum distance of the evaluation point $\mathbf{x}^t$ to the set of centers $\mathbf{X}^c$,

$$f_e(\mathbf{x}^t) = \min_{\mathbf{x}^c \in \mathbf{X}^c} \|\mathbf{x}^t - \mathbf{x}^c\|. \tag{7.2}$$

Let $\mathbf{X}^c_{i,k} = [\mathbf{x}^c_{i,1} \dots \mathbf{x}^c_{i,k}]$ be the matrix containing the k-nearest neighbors of centroid $\mathbf{x}^c_i$ in $\mathbf{X}^c$. The threshold for extrapolation detection, d_e, is then computed as

$$d_e = \frac{1}{N_\mathrm{R}} \sum_{i=1}^{N_\mathrm{R}} \left(\frac{1}{k} \sum_{j=1}^{k} \|\mathbf{x}^c_i - \mathbf{x}^c_{i,j}\| \right) \tag{7.3}$$

A demonstration of this method is plotted on Figure 7.2 where the points outside of the α-shape of the training data (shown in black) are correctly flagged in extrapolation.

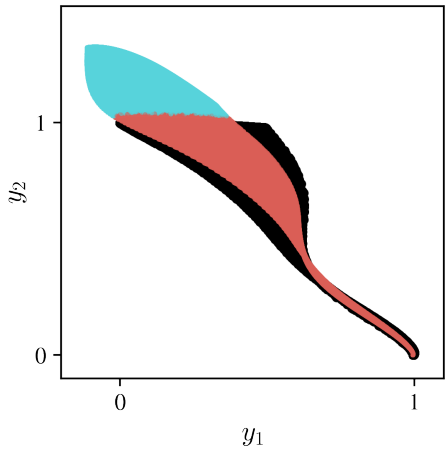


Figure 7.2: Evaluation points $\mathbf{Y}^t$ colored by the extrapolation flag: blue and red represent points detected outside and inside the training range, respectively. The α-shape of the training data is drawn underneath in black.

7.2.2 Growing RBF

The input/output pairs detected in extrapolation have to be learned by the RBF to improve the function mapping in that newly explored region of the latent space. This topic is known as growing RBF, or more generally as resource-allocating network (RAN), and several training techniques have been proposed in the literature for their on-line training [Platt 1991, Kadirkamanathan & Niranjan 1993, Karayiannis & Mi 1997, Huang *et al.* 2005, Bortman & Aladjem 2009]. The common strategy is to sequentially feed new observations to the network. If an

observation makes a significant contribution to the overall performance of the model (it fulfills the novelty criterion given a certain metric), then a new center (or neuron) is added in the hidden layer. If not, the parameters of the network in the vicinity of the observation are updated. Recent improvement even allow to prune the network, providing an optimal architecture in terms of complexity (number of units in the hidden layer) [Huang *et al.* 2005, Bortman & Aladjem 2009].

The original implementation used least mean squares filters (LMS) [Platt 1991] for the update procedure of the network parameters. While the LMS algorithm iteratively updates the filter coefficients efficiently, it does not guarantee optimality in the least-square sense. In fact, it is a stochastic gradient descent algorithm and can therefore converge to a local minimum or exhibit some residual error even after convergence. In contrast, the optimal solution in the least-square sense can be obtained using methods like the Recursive Least Squares (RLS) algorithm. This method provides a closed-form optimal solution in the least-square sense. However, they are computationally more expensive (due to the necessity of performing a matrix inversion) and may not be suitable for real-time applications. Kadirkamanathan and Niranjan instead used extended Kalman filters (EKF) for the update procedure [Kadirkamanathan & Niranjan 1993, Bortman & Aladjem 2009]. This provided a good trade-off between computational efficiency and optimality of the solution (even though not strictly) in the least-square sense. Finally, these update procedures also require the tuning of many hyperparameters.

Here, we propose a novel and efficient update technique, optimal in the least-square sense, that follows the two-step training procedure of the off-line training. First, we obtain the new units in the hidden layer by performing a sequential *k-means* clustering of the observations detected outside the training range. Secondly, we efficiently retrain the whole RBF in "one-go" using the Schur complement.

7.2.2.1 Online *k-means*

The fist step in the update procedure is to generate new centers on the new subspace defined by the data points detected outside of the initial training range. This set of points is denoted as $\mathbf{X}^e \subset \mathbf{X}^t$. During the off-line initial training stage, the tesselation of the latent space was generated using the *k-means* algorithm, resulting in the set of centroids (centers for the RBF) $\mathbf{X}^c$. The count of the number of training points associated with each centroid is saved in matrix $\mathbf{C} \in \mathbb{R}^{N_\mathrm{R}}$.

In the online stage, we use an in-house adapted version of the k-means algorithm for sequential data, namely the *sequential k-means* algorithm, which is close to the original formulation of the *k-means* algorithm by MacQueen [MacQueen 1965], see also [Duda & Hart 2006]. The pseudo-code of the algorithm is described in Alg. 1.

A clear advantage of the distance-based formulation is that the numbers of new centers is automatically determined by the algorithm and does not come as an extra hyper-parameter. The algorithm is applied on the set $\mathbf{X}^e$ (blue points in Figure 7.2) in Figure 7.3.

Algorithm 1: Online k-means pseudo-code

for $\mathbf{x}^e \in \mathbf{X}^e$ **do**
  Find the closest centroids $\mathbf{x}_j^c \in \mathbf{X}^c$ to $\mathbf{x}^e$
  Compute $r = \|\mathbf{x}^e - \mathbf{x}_j^c\|$
  if $r < d_e$ **then**
    $C[j] \leftarrow C[j] + 1$
    $\mathbf{x}_j^c \leftarrow \mathbf{x}_j^c + (\mathbf{x}^e - \mathbf{x}_j^c)/C[j]$
  else
    Append $\mathbf{x}^e$ to $\mathbf{X}^c$
    $N_\mathrm{R} \leftarrow N_\mathrm{R} + 1$
    Append 1 to $\mathbf{C}$
  end
end

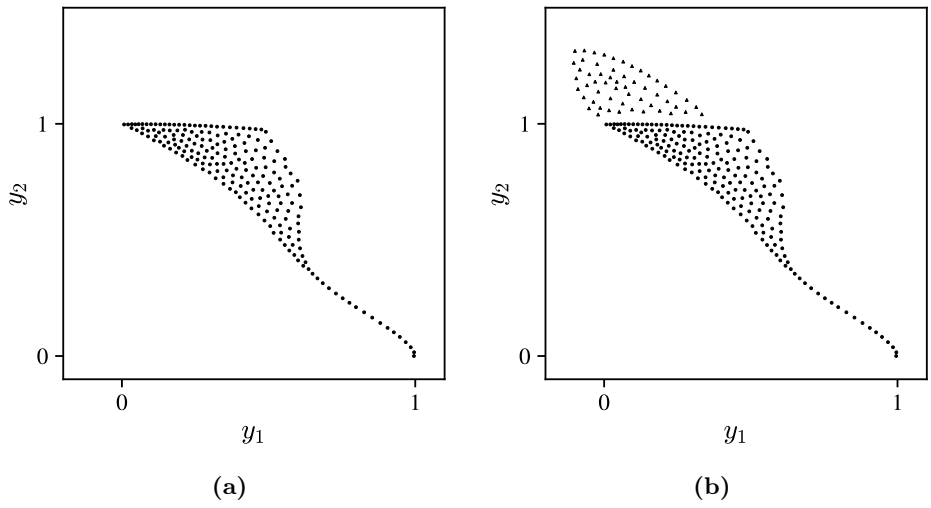


Figure 7.3: (a) Initial tesselation of the low-dimensional space after applying the *k-means* algorithm with $N_\mathrm{R} = 200$ where each circular dot represents a cluster centroids $\mathbf{y}^c \in \mathbf{Y}^c$. (b) Application of the online *k-means* algorithm on the set of states detected in extrapolation, $\mathbf{X}^e$ (blue region in Fig. 7.2, generating 55 new centroids, represented by triangles.

7.2.2.2 Update procedure of the RBF

Let us first consider an RBF with gaussian kernel. As long as all centers are distinct, the kernel matrix is always a symmetric positive definite real matrix and is therefore invertible. Let $\mathbf{\Phi}_{1,1} \in \mathbb{R}^{n\times n}$ represent the initial kernel matrix, $\mathbf{\Phi}_{2,2} \in \mathbb{R}^{m\times m}$ the kernel matrix of the new centers added by the online k-means algorithm and $\mathbf{\Phi}_{1,2} = \mathbf{\Phi}_{2,1}^T \in \mathbb{R}^{n\times m}$ the cross kernel matrix between initial and new centers, respectively. Hence, optimally updating the model parameters in the least-square sense requires the inversion of the augmented kernel matrix, of size $(n+m)\times(n+m)$, defined as,

$$\mathbf{\Phi} = \begin{pmatrix} \mathbf{\Phi}_{1,1} & \mathbf{\Phi}_{1,2} \\ \mathbf{\Phi}_{2,1} & \mathbf{\Phi}_{2,2} \end{pmatrix}, \tag{7.4}$$

However, as the number of centers increases, this task can become computationally intensive, especially for an on-line procedure with high retrain frequency.

Schur complement and matrix inversion

Let M be a square matrix of size $(n+m)\times(n+m)$, written in terms of block partitions as

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}, \tag{7.5}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{m\times n}$ and $D \in \mathbb{R}^{m\times m}$. If A is invertible, the Schur complement of block A of matrix M is defined as

$$M/A = D - CA^{-1}B. \tag{7.6}$$

We then have the following theorem, see for instance Gallier [Gallier 2011] for a proof.

Theorem A (Invertibility of Schur complement) *If A is invertible, then*

$$M \text{ is invertible} \iff M/A \text{ is invertible,}$$

which implies that

$$M^{-1} = \begin{pmatrix} A^{-1} + A^{-1}B(M/A)^{-1}CA^{-1} & -A^{-1}B(M/A)^{-1} \\ -(M/A)^{-1}CA^{-1} & (M/A)^{-1} \end{pmatrix}. \tag{7.7}$$

Application of the Schur complement to the update procedure of the RBF

Based on the definition of the augmented kernel matrix, Eq. (7.4), it is straightforward to apply the Schur complement for efficient matrix inversion. In fact, since $\mathbf{\Phi}_{1,1}^{-1}$ is known from the initial RBF training and $\mathbf{\Phi}$ is invertible (as a symmetric positive definite real matrix), theorem A states that the Schur complement of

the block $\mathbf{\Phi}_{1,1}$ of matrix $\mathbf{\Phi}$ is invertible. We can then compute its inverse and immediately construct $\mathbf{\Phi}^{-1}$ with Eq. (7.7). Hence, inverting the whole matrix requires only the inversion of a $m \times m$ matrix instead of a $(n+m) \times (n+m)$ one.

In the case of thin-plate spline kernel used here, the poly-harmonic terms complicate the processus. The augmented system matrix with poly-harmonic terms can be written as,

$$M = \begin{pmatrix} \mathbf{\Phi}_{1,1} & \mathbf{\Phi}_{1,2} & \mathbf{P}_1 \\ \mathbf{\Phi}_{2,1} & \mathbf{\Phi}_{2,2} & \mathbf{P}_2 \\ \mathbf{P}_1^T & \mathbf{P}_2^T & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{\Phi} & \mathbf{P} \\ \mathbf{P}^T & 0 \end{pmatrix}, \tag{7.8}$$

where $\mathbf{P}_1$ and $\mathbf{P}_2$ are the polynomial matrices computed on the initial and new centers, respectively. See §5.2.4 for the definition of the polynomial matrix. We can circumvent the problem by first using the Schur complement to invert the first block, $\mathbf{\Phi}$, composed of the $\mathbf{\Phi}_{i,j}$ terms, as described above for a gaussian kernel. It is worth noting that due to the nature of the thin-plate spline kernel, $\mathbf{\Phi}_{i,i}$ are only conditionnaly positive definite matrices. Thus, invertibility is not always guaranteed. However, we did not encounter any non-invertible matrix in the numerical tests realized. Secondly, we compute the full inverse by using again the Schur complement of block $\mathbf{\Phi}$ of M. In total, this requires first the inversion of an $m \times m$ matrix and secondly a $d_p \times d_p$ matrix, where d_p is the number of monomial in the space of polynomials Π_p^d of d variables and degrees up to p. Since usually the degree p is chosen small (i.e, $p < 3$) and $m < n$, these two inversions are more efficient that the inversion of the full $(n+m+d_p) \times (n+m+d_p)$ matrix directly. This is shown in Figure 7.4, where run-time is compared with that of the direct inversion. When m is proportionally small compared to n, direct inversion is systematically slower. During a simulation, since f_{retrain} is set to a low value, m is always small compared to n (less than 5% of m). This provides efficient inversion of the RBF system matrix while maintaining optimality of the model in the least-square sense.

Finally, the RBFs corresponding to each cluster are retrained and the resulting error in latent space and physical space with the true value of Mutation++ are shown in Figure 7.5. Clearly, error drastically decreases in the extrapolation region, demonstrating the efficient and optimal active learning capability of the model.

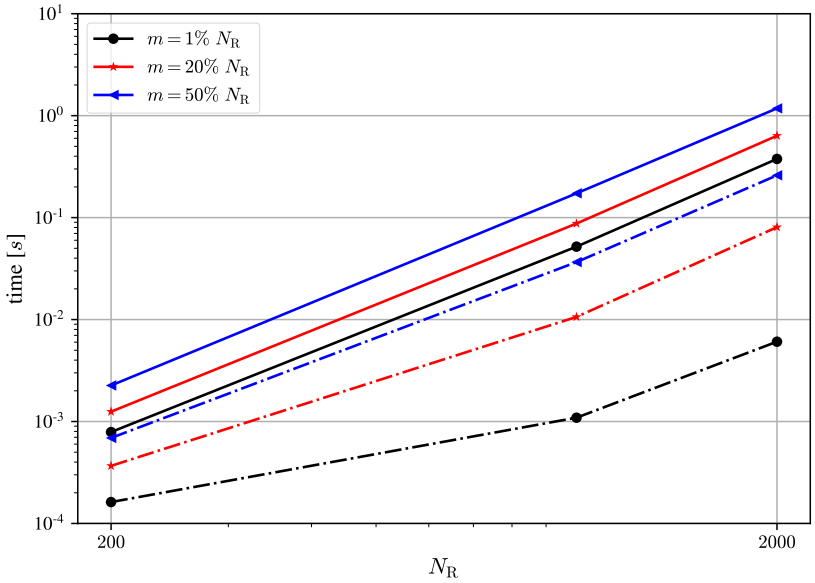


Figure 7.4: Comparison of the retraining run-time of a RBF with N_R numbers of centers initially and m (defined as percentage of N_R) new centers. Solid lines correspond to direct inversion and dashed-dotted lines to the technique using the Schur complement.

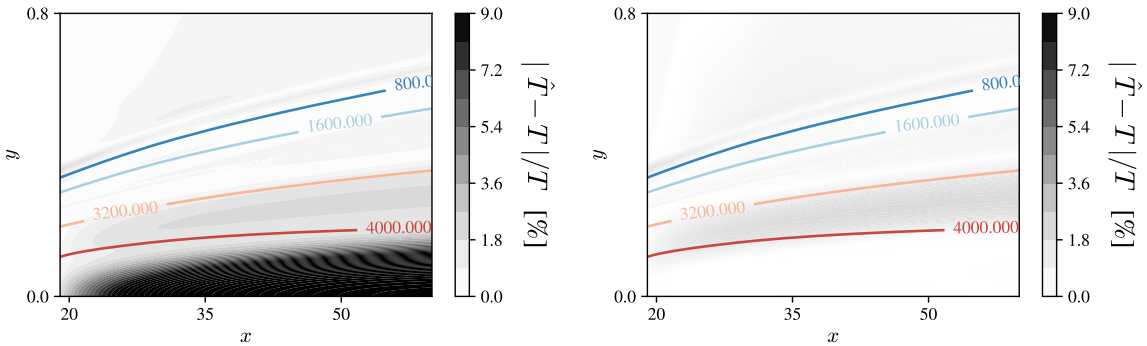


Figure 7.5: Comparison of the relative temperature error $|\hat{T} - T|/T$ in percent with contours of temperature in the physical space. (a) Original model, (b) model after retraining.

7.3 Results

7.3.1 Low-fidelity to high-fidelity transient simulation

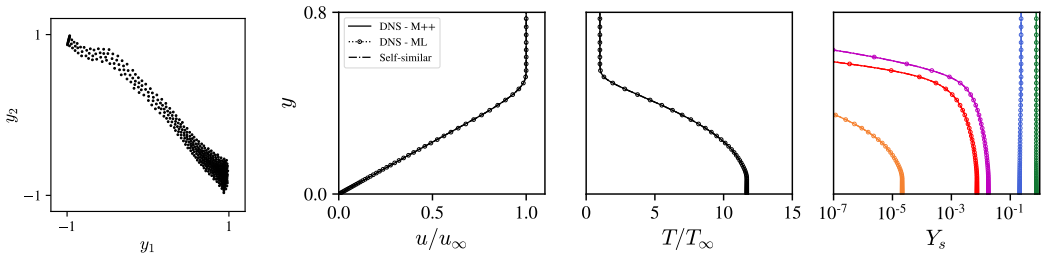
The first test chosen is a low-fidelity to high-fidelity transient simulation based on case A (see §4.2.2 for a complete description). The simulation is initialized with a set of locally self-similar solutions in chemical nonequilbrium. The flow then transiently evolves until converging to the steady solution to the Navier-Stokes equations. This case is chosen since large differences are observed between the low-fidelity and high-fidelity solutions using the full-order thermochemical gas model (in this case Mutation++). Hence, a model trained only on the low-fidelity simulation will encounter many new states that have to be actively learned throughout the transient simulation.

The base model is trained on the locally self-similar solution, and white noise with 2% amplitude is added to improve the robustness of the model. The hyper-parameters of the models are : $d = 2,\ N_C = 2,\ N_R = 200$. During the simulation, the model is updated every 100 iterations.

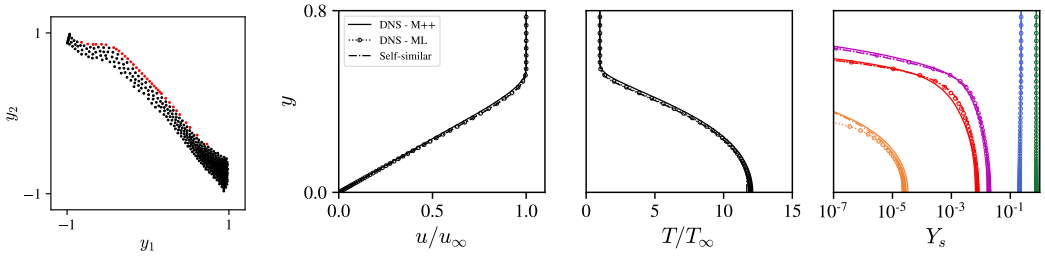
Figure 7.6 presents the evolution of the population of the RBF centers in the latent space as well as boundary layer profiles at different instant during the simulation. As the simulation advances in-time, more centers are added as new thermodynamic states are encountered. It is important to note that the number of centers rapidly plateaus after a first transient where many new states are discovered. The final total number of centers is 602. This proves the convergence of the active learning process during the simulation. Correspondingly, the boundary layer profiles evolve towards the converged solution. Interestingly, at a given time, the solution using the reduced-order model lags slightly behind the one with the full-order model. However, it eventually converges very closely to the reference simulation using Mutation++, as supported by the residual plots for both simulation in Figure 7.7. The main difference is in the radical mass fractions. However, they are present in such small quantities that they do not alter more relevant quantities of interest such as the velocity profiles and maximum temperature within the boundary layer. Furthermore, a reduced-order model is not expected to be precise in the range $[0, 10^{-5}]$.

Figure 7.8 presents the evaluation time of thermochemical properties for all grid points. The model initially performs 80% faster than Mutation++. As the solution progresses and new centers are added, the performance of the model slightly degrades until reaching a final performance that is 75% faster than Mutation++. This loss of performance is due to the higher evaluation cost after progressively growing the RBF. In fact, we recall here that the time complexity of the evaluation step of the RBF is $O(C_{RBF} \times N_t \times N_{\mathrm{R}} \times d)$. Hence, as N_{R} increases during the active learning process, so does the time complexity. However, since the number of new centers at each update is small and the load is split between two clusters, this additional cost is not detrimental to the overall performance of the data-driven model.

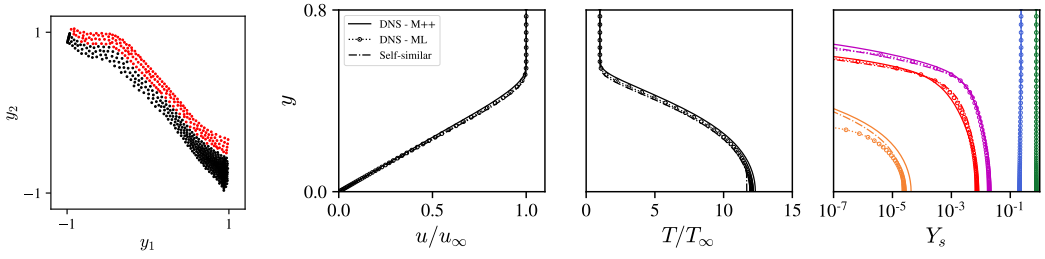
The time spent during the updating step is however hard to evaluate since it


y_2
y_1
y
u/u_∞
DNS - M++
DNS - ML
Self-similar
T/T_∞
Y_s

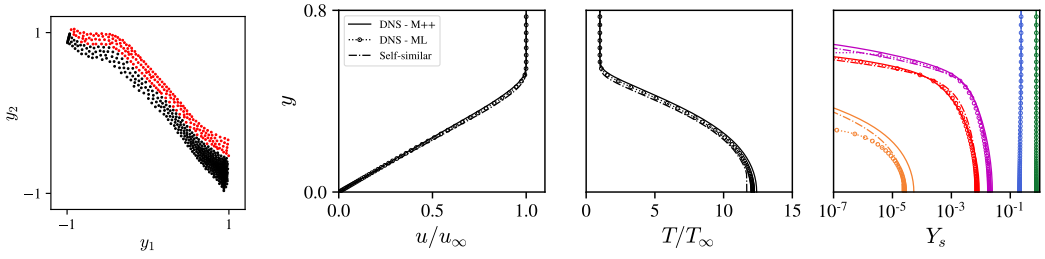
(a) (b)


y_2
y_1
y
u/u_∞
DNS - M++
DNS - ML
Self-similar
T/T_∞
Y_s

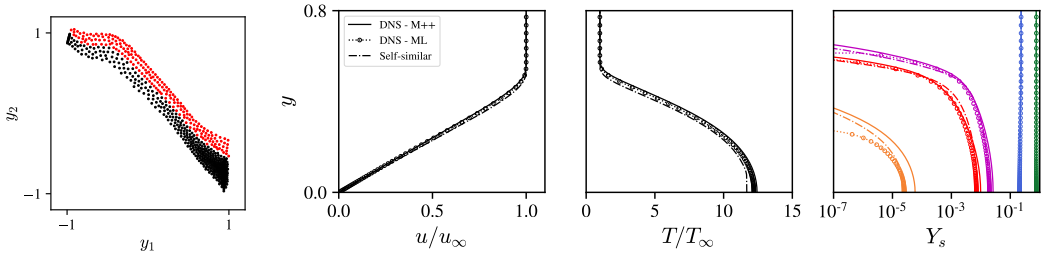
(c) (d)


y_2
y_1
y
u/u_∞
DNS - M++
DNS - ML
Self-similar
T/T_∞
Y_s

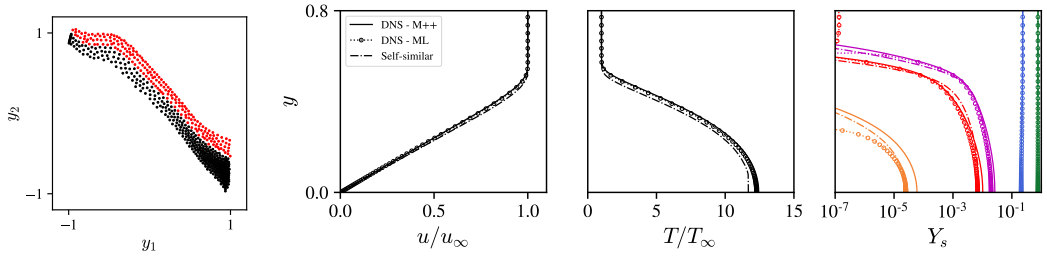
(e) (f)


y_2
y_1
y
u/u_∞
DNS - M++
DNS - ML
Self-similar
T/T_∞
Y_s

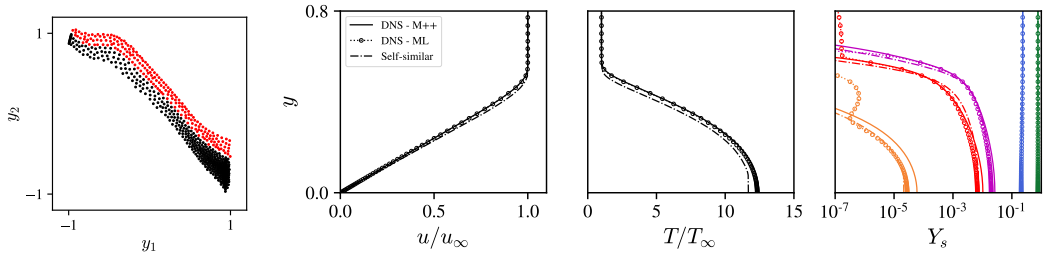
(g) (h)


y_2
y_1
y
u/u_∞
DNS - M++
DNS - ML
Self-similar
T/T_∞
Y_s

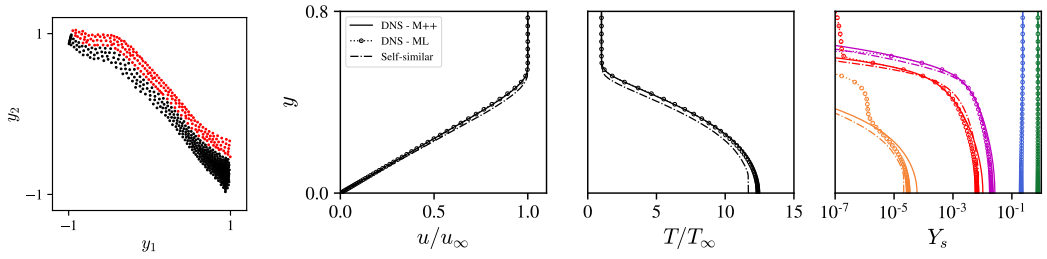
(i) (j)


DNS - M++
DNS - ML
Self-similar
y_1
y_2
y
u/u_∞
T/T_∞
Y_s

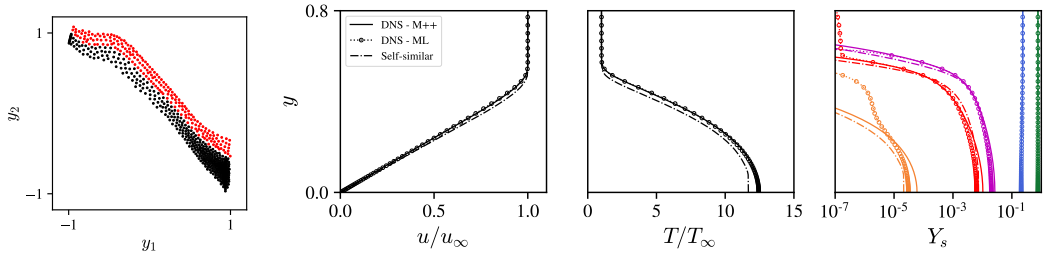
(k) **(l)**


DNS - M++
DNS - ML
Self-similar
y_1
y_2
y
u/u_∞
T/T_∞
Y_s

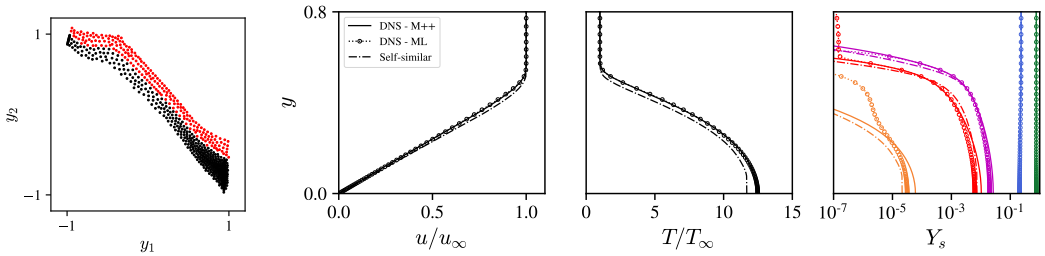
(m) **(n)**


DNS - M++
DNS - ML
Self-similar
y_1
y_2
y
u/u_∞
T/T_∞
Y_s

(o) **(p)**

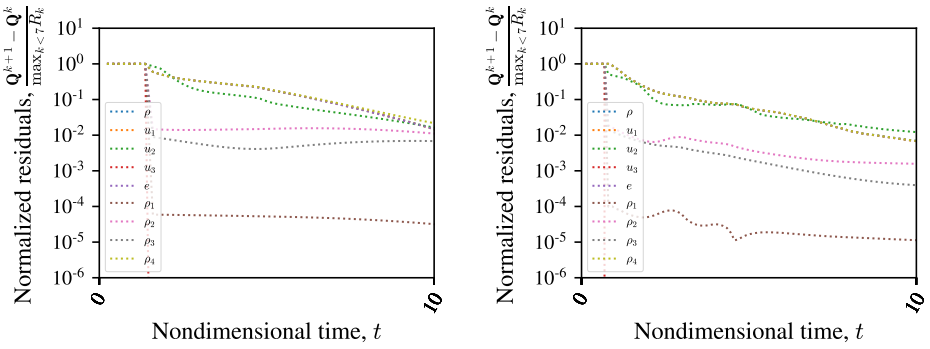

DNS - M++
DNS - ML
Self-similar
y_1
y_2
y
u/u_∞
T/T_∞
Y_s

(q) **(r)**



(s) **(t)**

Figure 7.6: Transient low to high-fidelity simulation of case A using both full and reduced-order thermochemical models. Left column: RBF centers in the latent space. Black and red dots represent initial and newly added centers during the simulation, respectively. Right column: evolution of the boundary layer profiles. Solid and dotted lines with markers correspond to the solution using the full and reduced-order thermochemical model, respectively. Each line represents uniformly sampled instantaneous snapshot at a non-dimensional time $t \in [0, 10]$, ordered chronologically.



(a) **(b)**

Figure 7.7: Evolution of residuals during the transient simulation using (a) full-order and (b) reduced-order thermochemical model, respectively. In both case, all residual curves decrease by at least 2 order of magnitude, which is deemed sufficient to assess convergence of the solution.

depends on the number of points detected during extrapolation. To evaluate it empirically, two simulations have been run for 1000 iterations, with and without update every 100 iterations, respectively. The simulation with updates was 1.05 times slower, even though it was initialized with the locally self-similar solutions and the update load is higher early on during the transient, as seen on Figures 7.6 and 7.8. Hence, the update time is marginal in the total time to solution and can be omitted as a first approximation. Hence, directly integrating the curve of the data-driven reduced order model leads to a time to solution 77% times faster than Mutation++.

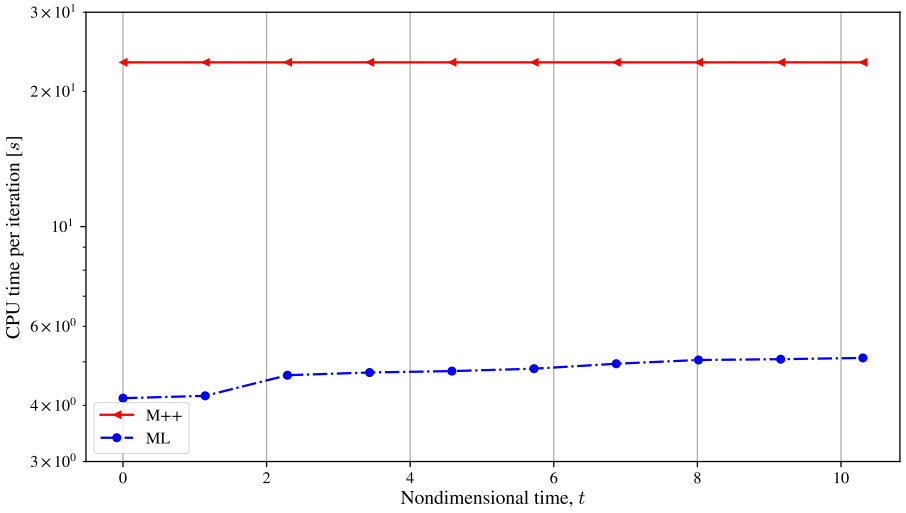


Figure 7.8: Comparison of time per iteration during the transient simulation using both full (solid line) and reduced-order thermochemical model (dashed-dotted).

This first test proves that the data-driven reduced order thermochemical model can actively learn new-states on-the-fly. Thus, the models are predictive, generalizable and can help reduce the high cost associated with high-fidelity simulation of hypersonic flows in chemical nonequilibrium, with minimal impact on the accuracy of the solution.

7.3.2 Optimally disturbed boundary layer with blowing-suction

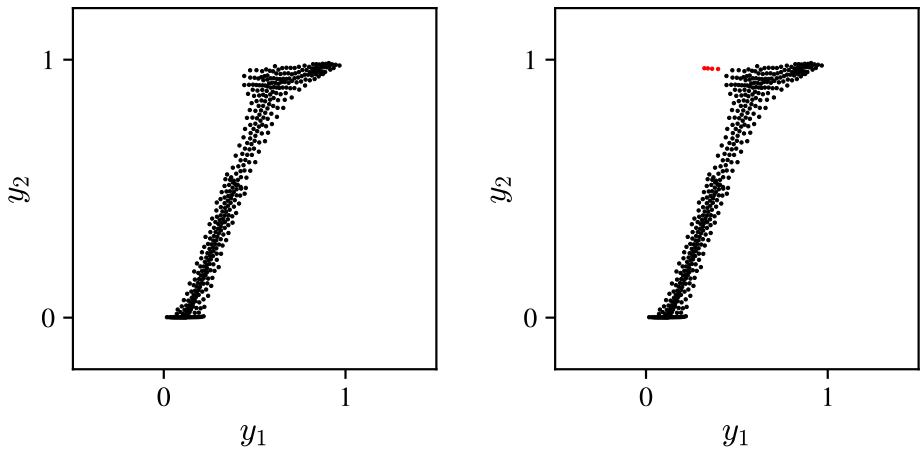
The second test case is the optimally disturbed boundary layer, described in §3.1.3.2. The model is trained on the steady solution, supplemented with white noise of 2% amplitude to increase robustness. The hyper-parameters of the model are : $d = 2,\ N_C = 2,\ N_R = 200$.

Disturbances are triggered in the boundary layer at the wall, through a blowing-suction strip, as described in §3.1.1. Starting from the steady solution, the simulation is advanced with the data-driven model and an update frequency of 100

iterations until a time-periodic state is reached. Flow snapshots are then collected and the post-processed in the same way as described in §3.1.2.

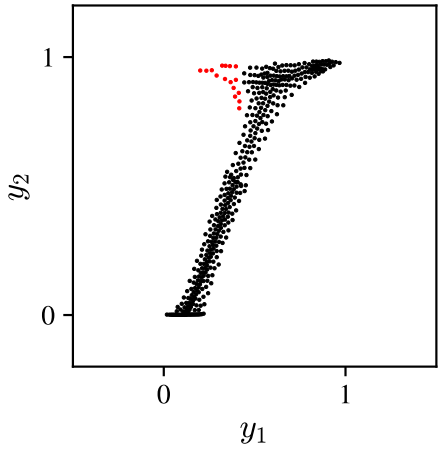
Figure 7.9 shows the evolution of the population of the RBF centers in the latent space during the transient phase all the way up to the time-periodic state, showing that the model learns online during the simulation.

Figure 7.10 depicts the RMS wall pressure and growth rates obtained. However, the level of agreement observed in this figure is not as close to the reference solution as that of the two previous models illustrated in Figures 6.2 and 6.5. In fact, the aforementioned models began from a time-periodic solution achieved with Mutation++, resulting in a shorter integration time. In contrast, the simulation presented in Figure 7.10 underwent a transient phase until reaching a time-periodic state. The time-periodicity of the solution is proven by the similarity in results between the post-processed dynamics at intervals of eight forcing periods, as demonstrated in Figure 7.10, wherein the RMS wall pressure and growth rate of the second-mode instability are identical. However, the time-periodic state slightly deviates from the reference solution, explaining the higher discrepancies. The overall dynamic remains close to the reference, noting that the thermochemical model (i.e TPG or CNEQ) has a higher impact on the dynamics, as shown in 3.5. This demonstrates the algorithm's capability to actively learn a reduced-order thermochemical model that is both accurate and efficient during a time-marching simulation, resulting in decreased CPU time required to obtain reliable results. In fact, Figure 7.11 shows that by integrating the time per iteration, the solution was obtained using 75% less computational resources than when using Mutation++.

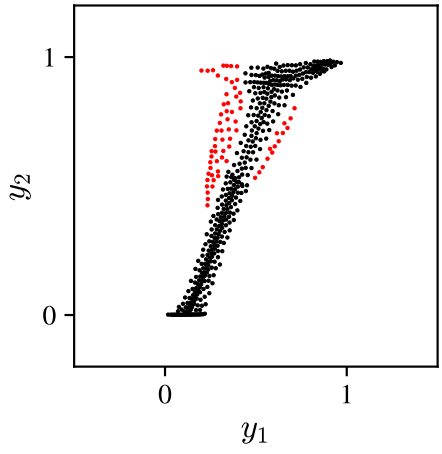


(a)

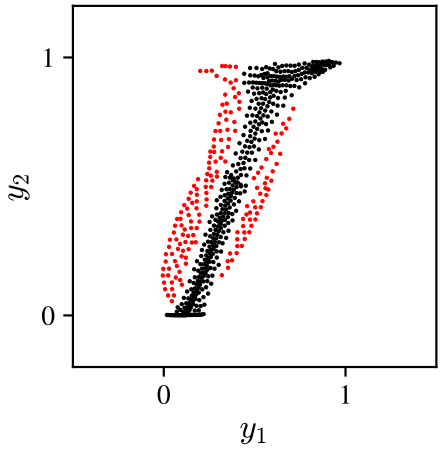
(b)



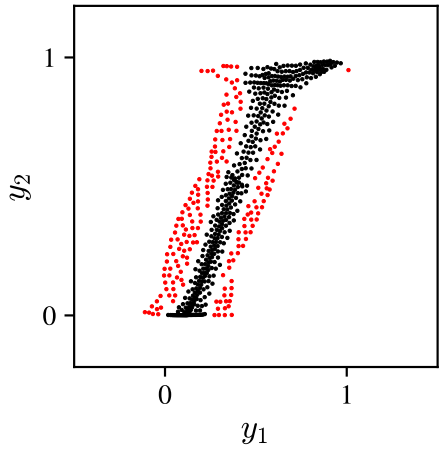
(c)



(d)



(e)



(f)

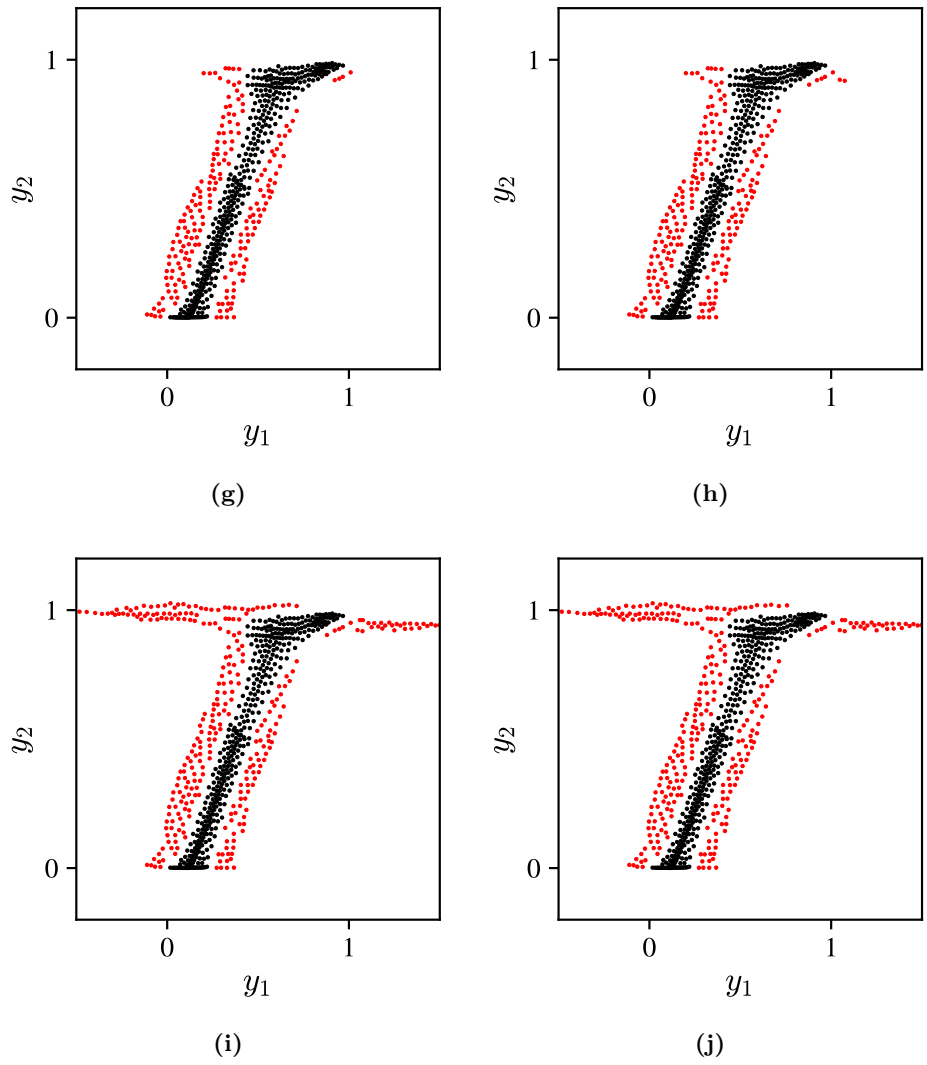


Figure 7.9: Evolution of RBF centers in the latent space during the unsteady simulation of second-mode growth in case A boundary layer. Black and red dots represent initial and newly added centers during the simulation, respectively.

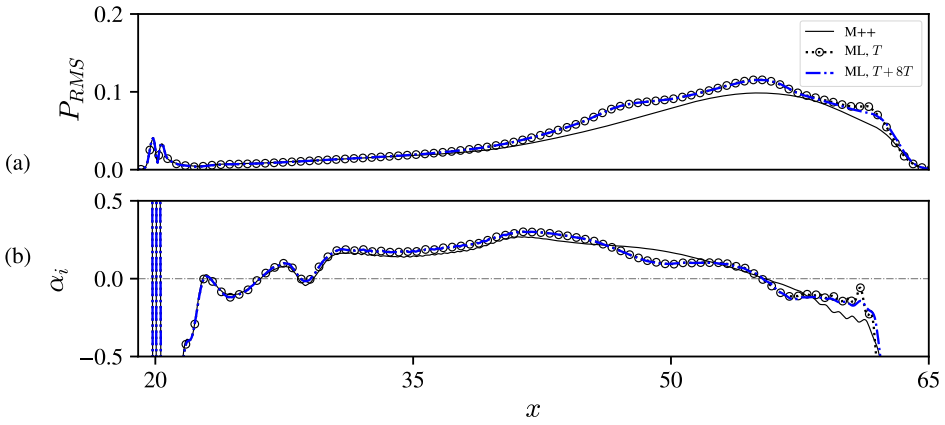


Figure 7.10: Evolution of (a) RMS wall pressure, and (b) growth rate α_i, computed using Eq. (3.9), as a function of streamwise position x for case A. Solid lines correspond to the baseline solution (Mutation++). Dotted lines with circles and blue dashed-dotted correspond to the result of the active data-driven model 8 forcing periods apart, proving the convergence of the data-driven model to a time-periodic state.

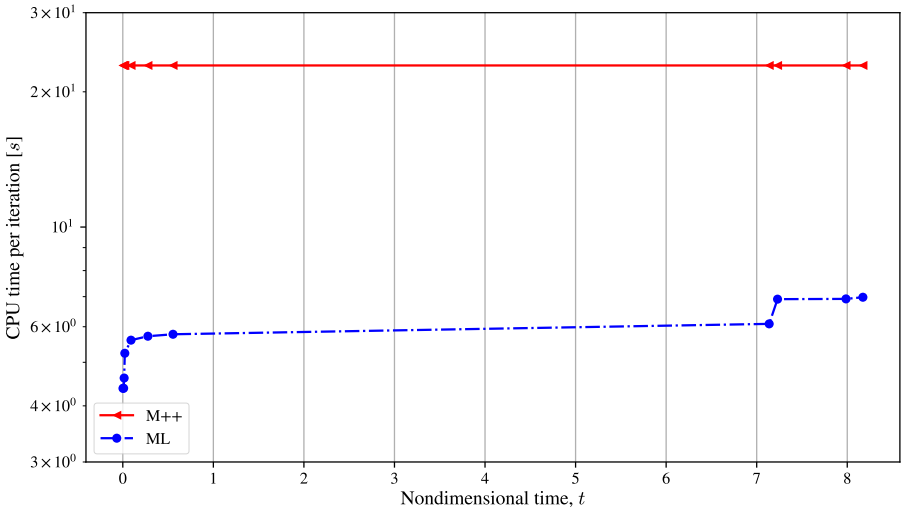


Figure 7.11: Comparison of time per iteration during the unsteady simulation using both full (solid line) and reduced-order thermochemical model (dashed-dotted).

Conclusion

The major challenges posed by the need for space exploration have undeniably revitalized research on the practical applications of hypersonic flows. At such high speed and high enthalpy regimes as those encountered during atmospheric (re-)entry or sustained hypersonic cruise, gases exhibit behaviors that deviate from predictions made by conventional models used in CFD solvers. As a result, enhanced models become imperative to accurately capture the complex interactions among various phenomena prevalent in these configurations.

In the first part of this dissertation, we have presented a computational tool that can accurately simulate flows at hypersonic speed, including high-enthalpy gas effects in the presence of weak and strong shockwaves. This is achieved by augmenting the flow solver with additional conservation equations for each chemical species while the Mutation++ library models the thermodynamic, transport and kinetics terms.

Extensive code verification has been completed against previous results for reacting boundary layers, SBLI cases, and jet injections in high-speed crossflow. The applications have been extended towards novel, higher-enthalpy cases where the effects of gas thermochemistry are pronounced. Our findings are in agreement with the literature, and provide insight in cases and regimes that have not been thoroughly investigated before.

Supplementing the flow solver with complex thermochemical models has, however, a negative impact on its performance. A simulation in chemical nonequilibrium is in general one order of magnitude slower than its equivalent with a perfect gas model. More generally, many engineering applications rely on expensive high-dimensional functions such as Mutation++.

Therefore, we presented in the second part of this work a novel technique to reduce any high-dimensional look-up library to a lower-dimensional surrogate, and thus reduce the CPU costs of numerical simulations that rely on these libraries. Several machine learning techniques have been used: encoding based on deep neural networks, community clustering, surrogate modeling and classification in a three-step learning phase: in the first step, the proposed input/output-encoder architecture has been shown to outperform partial least-squares (PLS) for dimensionality reduction of input/output relations. Clustering was performed using Newman's algorithm. It discovered physically consistent clusters in the low-dimensional latent space without *a-priori* knowledge of the number of clusters. Then, a random-forest classifier was trained, which reliably predicted the cluster of previously unencountered data points. Finally, a radial basis function network was constructed on each cluster to obtain a continuous and local representation of the library via a reduced-order surrogate model.

The combination of these pre-processing steps has been shown to improve the efficiency of the simulation while maintaining high accuracy on several canonical

cases, such as reacting boundary layers and shock wave boundary layer interaction. During this demonstration, we observed up to 70% CPU time decrease to compute the thermochemical properties of the mixture. In steady cases, the newly computed base flows were recovered accurately when compared to the true solution (obtained with Mutation++). In unsteady cases, the reduced-order thermochemical models faithfully reproduced the dynamics of the flow.

A high level of accuracy was only achievable when the training set contained all possible flow features. In cases where the model was trained on an incomplete training set, solutions developed instabilities and quickly diverged due to the poor performance of the model in extrapolation. In tandem, we also demonstrated that when a complete training set is not available or too expensive to obtain, self-similar solutions provide a good approximation of the thermodynamic state manifold, at a fraction of the cost, thus proving to be a viable substitute for training.

These observations led to the augmentation of the model with an on-the-fly learning procedure, which was successfuly demonstrated for two types of transient simulations, removing the need for a complete training set and making the technique truly predictive and generalizable. In the first test, namely a low to high-fidelity transient simulation, the data-driven model converged quickly to a high-fidelity solution with good agreement to the reference while decreasing the time-to solution by almost 80% compared to using Mutation++. The second test started from the high-fidelity steady solution and the boundary layer was optimally forced wiuth blowing and suction. The simulation using the data-driven model converged to a time-periodic state that provided good approximation of the general dynamics of the boundary layer while using 75% less computation time.

Outlook

There are several directions in which there is significant potential for improvement and extension of this tools' capabilities.

First, this computational framework can be readily ported into other application fields to accelerate simulations that rely on high-dimensional look-up tables to model complex flow behavior such as combustion, phase-change or fluid-particle interactions. In the context of hypersonic flows, we could first readily apply the technique to a more ambitious DNS simulation, such as the transitional boundary layer of Di Renzo and Urzay [Di Renzo & Urzay 2021]. In fact, we observed in §6.2.2.1 that the self-similar solutions provided a good estimation of the thermodynamic manifold of the turbulent section. With active learning capability, we expect the statistics to remain close to the one of the full-order thermochemical model, while the latter are significantly more expensive to obtain. There are also more accurate options for gas modeling, including thermal nonequilibrium and state-to-state transitions, that can also benefit from adaptive reduced-order gas model to accelerate such simulations. There is also great interest in extending our results for different gas mixtures, representing atmospheric compositions of celestial

objects interesting for space exploration, or more complex geometries. Application to ablation and more detailed chemical mechanisms can also be pursued in the context of future work.

Secondly, several improvements can be made to improve the technique performances and accuracy. They are non-exhaustively listed here as a potential basis for future work;

- **Active learning of the low-dimensional manifold.** During the online stage, the embedding of the input thermodynamics states does not evolve and is solely based on the off-line training. However, in complex scenarios, different thermodynamic states could be projected onto the same location if the off-line training was not sufficient to properly learn the low-dimensional manifold. It might therefore be of interest to adapt the low-dimensional manifold on-the-fly as well.

- **Initialization of new 'Newman' clusters.** The new RBF centers are added to the corresponding Newman cluster given by the classifier. However, if one of this Newman clusters grows too much, the performance of the corresponding surrogate model would decrease drastically. More importantly, the Newman cluster would lose its inner thermodynamic consistensy of similar states. An interesting approach would be to initialize new Newman clusters online based on a specific criterion. One potential criterion could be the maximum number of centers allowed in the surrogate model or the maximum variation observed in an output within a cluster. By monitoring these criteria, new clusters can be initialized when they are exceeded.

- **Pruning of RBF.** Alternatively, the states encountered during a time-marching simulation could shift from one region to another of the manifold. In this case, some RBF centers would become useless (too far from the evaluation region) while still impacting its evaluation cost. It could be interesting to prune these centers to improve the performance of the surrogate model while not altering its accuracy. One can potentially use the work of [Bortman & Aladjem 2009] as a starting point.

- **Criterion for update frequency $f_{\mathbf{up}}$.** The update of the algorithm is done at a constant update frequency. However, the update load (i.e the numbers of new states encountered at a given time) is different throughout the simulation and is usually higher early on in the transient. Therefore, the update frequency could be instead based on a tailored criterion, such as done in Adaptive Mesh Refinement.

Bibliography

[An 1996] Guozhong An. *The effects of adding noise during backpropagation training on a generalization performance.* Neural computation, vol. 8, no. 3, pages 643–674, 1996. (Cited on page 108.)

[Anderson 2019] John D. Anderson. Hypersonic and high-temperature gas dynamics. American Institute of Aeronautics and Astronautics (AIAA), Washington, DC, 3 édition, Apr 2019. (Cited on pages 3, 11 and 38.)

[Attili *et al.* 2014] Antonio Attili, Fabrizio Bisetti, Michael E Mueller and Heinz Pitsch. *Formation, growth, and transport of soot in a three-dimensional turbulent non-premixed jet flame.* Combustion and flame, vol. 161, no. 7, pages 1849–1865, 2014. (Cited on page 6.)

[Baldi & Hornik 1989] Pierre Baldi and Kurt Hornik. *Neural networks and principal component analysis: Learning from examples without local minima.* Neural networks, vol. 2, no. 1, pages 53–58, 1989. (Cited on page 76.)

[Bansal *et al.* 2015] Gaurav Bansal, Ajith Mascarenhas and Jacqueline H Chen. *Direct numerical simulations of autoignition in stratified dimethyl-ether (DME)/air turbulent mixtures.* Combustion and Flame, vol. 162, no. 3, pages 688–702, 2015. (Cited on page 6.)

[Ben-Yakar *et al.* 2006] Adela Ben-Yakar, MG Mungal and RK Hanson. *Time evolution and mixing characteristics of hydrogen and ethylene transverse jets in supersonic crossflows.* Physics of Fluids, vol. 18, no. 2, page 026101, 2006. (Cited on pages 49 and 60.)

[Bertin & Cummings 2003] John J. Bertin and Russell M. Cummings. *Fifty years of hypersonics: where we've been, where we're going.* Progress in Aerospace Sciences, vol. 39, no. 6, pages 511–536, 2003. (Cited on page 1.)

[Bettebghor *et al.* 2011] Dimitri Bettebghor, Nathalie Bartoli, Stéphane Grihon, Joseph Morlier and Manuel Samuelides. *Surrogate modeling approximation using a mixture of experts based on EM joint estimation.* Structural and multidisciplinary optimization, vol. 43, no. 2, pages 243–259, 2011. (Cited on pages 6, 72 and 82.)

[Bhagatwala *et al.* 2014] Ankit Bhagatwala, Jacqueline H Chen and Tianfeng Lu. *Direct numerical simulations of HCCI/SACI with ethanol.* Combustion and flame, vol. 161, no. 7, pages 1826–1841, 2014. (Cited on page 6.)

[Bishop 1995] Chris M Bishop. *Training with noise is equivalent to Tikhonov regularization.* Neural computation, vol. 7, no. 1, pages 108–116, 1995. (Cited on page 108.)

[Bortman & Aladjem 2009] Maria Bortman and Mayer Aladjem. *A growing and pruning method for radial basis function networks.* IEEE Transactions on Neural Networks, vol. 20, no. 6, pages 1039–1045, 2009. (Cited on pages 7, 116, 117 and 135.)

[Bottin *et al.* 1999] Benoit Bottin, David Vanden Abeele, Mario Carbonaro, Gerard Degrez and Gabbita SR Sarma. *Thermodynamic and transport properties for inductive plasma modeling.* Journal of thermophysics and heat transfer, vol. 13, no. 3, pages 343–350, 1999. (Cited on page 3.)

[Bouhlel *et al.* 2016] Mohamed Amine Bouhlel, Nathalie Bartoli, Abdelkader Otsmane and Joseph Morlier. *Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction.* Structural and Multidisciplinary Optimization, vol. 53, no. 5, pages 935–952, 2016. (Cited on pages 6, 72, 75, 77 and 80.)

[Breiman 2001] Leo Breiman. *Random forests.* Machine learning, vol. 45, no. 1, pages 5–32, 2001. (Cited on page 84.)

[Broomhead & Lowe 1988] David S Broomhead and David Lowe. *Radial basis functions, multi-variable functional interpolation and adaptive networks.* Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988. (Cited on pages 5 and 80.)

[Brunton *et al.* 2016] Steven L Brunton, Joshua L Proctor and J Nathan Kutz. *Discovering governing equations from data by sparse identification of nonlinear dynamical systems.* Proceedings of the national academy of sciences, vol. 113, no. 15, pages 3932–3937, 2016. (Cited on page 5.)

[Buhmann 2000] Martin D Buhmann. *Radial basis functions.* Acta numerica, vol. 9, pages 1–38, 2000. (Cited on pages 5, 80 and 82.)

[Candler 2019] Graham V. Candler. *Rate Effects in Hypersonic Flows.* Annual Review of Fluid Mechanics, vol. 51, no. 1, pages 379–402, jan 2019. (Cited on pages 2 and 3.)

[Carpenter & Kennedy 1994] Mark Carpenter and Christopher Kennedy. *Fourth-order 2N-storage Runge-Kutta schemes.* Technical report, NASA, 07 1994. (Cited on page 27.)

[Cassel 2003] Louis A Cassel. *Applying jet interaction technology.* Journal of Spacecraft and Rockets, vol. 40, no. 4, pages 523–537, 2003. (Cited on page 54.)

[Cenedese *et al.* 2022] Mattia Cenedese, Joar Axås, Bastian Bäuerlein, Kerstin Avila and George Haller. *Data-driven modeling and prediction of non-linearizable dynamics via spectral submanifolds.* Nature communications, vol. 13, no. 1, page 872, 2022. (Cited on page 5.)

[Chai *et al.* 2015] Xiaochuan Chai, Prahladh S Iyer and Krishnan Mahesh. *Numerical study of high speed jets in crossflow.* Journal of Fluid Mechanics, vol. 785, pages 152–188, 2015. (Cited on pages 49, 52 and 60.)

[Chang *et al.* 1997] C. L. Chang, H. Vinh and Mujeeb R. Malik. *Hypersonic boundary-layer stability with chemical reactions using PSE.* In 28th Fluid Dynamics Conference, pages 1–23, Reston, Virginia, Jun 1997. American Institute of Aeronautics and Astronautics (AIAA). (Cited on page 2.)

[Chang *et al.* 2001] F-J Chang, Jin-Ming Liang and Yen-Chang Chen. *Flood forecasting using radial basis function neural networks.* IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 31, no. 4, pages 530–535, 2001. (Cited on page 6.)

[Chu 1965] Boa-Teh Chu. *On the energy transfer to small disturbances in fluid flow (Part I).* Acta Mechanica, vol. 1, no. 3, pages 215–234, 1965. (Cited on page 58.)

[Cook & Cabot 2004] Andrew W Cook and William H Cabot. *A high-wavenumber viscosity for high-resolution numerical methods.* Journal of Computational Physics, vol. 195, no. 2, pages 594–601, 2004. (Cited on pages 28 and 30.)

[Cybenko 1989] George Cybenko. *Approximation by superpositions of a sigmoidal function.* Mathematics of control, signals and systems, vol. 2, no. 4, pages 303–314, 1989. (Cited on pages 5 and 76.)

[Di Renzo & Urzay 2021] M. Di Renzo and J. Urzay. *Direct numerical simulation of a hypersonic transitional boundary layer at suborbital enthalpies.* Journal of Fluid Mechanics, vol. 912, page A29, 2021. (Cited on pages vi, 2, 15, 42, 63, 64, 66, 67, 86, 94, 99, 108, 110, 111 and 134.)

[Di Renzo *et al.* 2020] Mario Di Renzo, Lin Fu and Javier Urzay. *HTR solver: An open-source exascale-oriented task-based multi-GPU high-order code for hypersonic aerothermodynamics.* Computer Physics Communications, vol. 255, page 107262, 2020. (Cited on pages 4 and 42.)

[Duan & Martin 2011] L Duan and MP Martin. *Direct numerical simulation of hypersonic turbulent boundary layers. Part 4. Effect of high enthalpy.* Journal of Fluid Mechanics, vol. 684, pages 25–59, 2011. (Cited on page 2.)

[Duan *et al.* 2010] L Duan, I Beekman and MP Martin. *Direct numerical simulation of hypersonic turbulent boundary layers. Part 2. Effect of wall temperature.* Journal of Fluid Mechanics, vol. 655, pages 419–445, 2010. (Cited on page 1.)

[Duan *et al.* 2011] L Duan, I Beekman and MP Martin. *Direct numerical simulation of hypersonic turbulent boundary layers. Part 3. Effect of Mach number.* Journal of Fluid Mechanics, vol. 672, pages 245–267, 2011. (Cited on page 1.)

[Duan *et al.* 2013] Le Duan, Xiaowen Wang and Xiaolin Zhong. *Stabilization of a Mach 5.92 boundary layer by two-dimensional finite-height roughness.* AIAA journal, vol. 51, no. 1, pages 266–270, 2013. (Cited on page 43.)

[Ducros *et al.* 1999] F Ducros, Valerie Ferrand, Franck Nicoud, C Weber, D Darracq, C Gacherieu and Thierry Poinsot. *Large-eddy simulation of the shock/turbulence interaction.* Journal of Computational Physics, vol. 152, no. 2, pages 517–549, 1999. (Cited on page 29.)

[Duda & Hart 2006] Richard O Duda and Peter E Hart. Pattern classification. John Wiley & Sons, 2006. (Cited on page 117.)

[Duraisamy *et al.* 2019] Karthik Duraisamy, Gianluca Iaccarino and Heng Xiao. *Turbulence modeling in the age of data.* Annual review of fluid mechanics, vol. 51, pages 357–377, 2019. (Cited on page 5.)

[Edelsbrunner *et al.* 1983] Herbert Edelsbrunner, David Kirkpatrick and Raimund Seidel. *On the shape of a set of points in the plane.* IEEE Transactions on information theory, vol. 29, no. 4, pages 551–559, 1983. (Cited on page 115.)

[Erdem 2011] Erinc Erdem. *Active flow control studies at Mach 5: measurement and computation.* PhD thesis, The University of Manchester, 2011. (Cited on pages vii, 49, 52, 53 and 54.)

[Ern & Giovangigli 1994] Alexandre Ern and Vincent Giovangigli. Multicomponent transport algorithms, volume 24. Springer Science & Business Media, 1994. (Cited on page 16.)

[Ern & Giovangigli 2004] A Ern and V Giovangigli. *Eglib: A general-purpose fortran library for multicomponent transport property evaluation.* Manual of EGlib version, vol. 3, page 12, 2004. (Cited on page 3.)

[Fiorina & Lele 2007] B. Fiorina and S.K. Lele. *An artificial nonlinear diffusivity method for supersonic reacting flows with shocks.* Journal of Computational Physics, vol. 222, no. 1, pages 246–264, 2007. (Cited on page 29.)

[Frank *et al.* 2004] Eibe Frank, Mark Hall, Len Trigg, Geoffrey Holmes and Ian H Witten. *Data mining in bioinformatics using Weka.* Bioinformatics, vol. 20, no. 15, pages 2479–2481, 2004. (Cited on page 84.)

[Franko & Lele 2013] Kenneth J Franko and Sanjiva K Lele. *Breakdown mechanisms and heat transfer overshoot in hypersonic zero pressure gradient boundary layers.* Journal of Fluid Mechanics, vol. 730, pages 491–532, 2013. (Cited on page 1.)

[Franzelli *et al.* 2013] B Franzelli, B Fiorina and N Darabiha. *A tabulated chemistry method for spray combustion.* Proceedings of the Combustion Institute, vol. 34, no. 1, pages 1659–1666, 2013. (Cited on page 5.)

[Fu *et al.* 2021] Lin Fu, Michael Karp, Sanjeeb T Bose, Parviz Moin and Javier Urzay. *Shock-induced heating and transition to turbulence in a hypersonic boundary layer.* Journal of Fluid Mechanics, vol. 909, page A8, 2021. (Cited on page 1.)

[Furumoto *et al.* 1997] Gregory H Furumoto, Xiaolin Zhong and John C Skiba. *Numerical studies of real-gas effects on two-dimensional hypersonic shock-wave/boundary-layer interaction.* Physics of Fluids, vol. 9, no. 1, pages 191–210, 1997. (Cited on page 47.)

[Gallier 2011] Jean Gallier. Geometric methods and applications: for computer science and engineering, volume 38. Springer Science & Business Media, 2011. (Cited on page 119.)

[Gkimisis *et al.* 2023] Leonidas Gkimisis, Bruno Dias, James B Scoggins, Thierry Magin, Miguel A Mendez and Alessandro Turchi. *Data-Driven Modeling of Hypersonic Reentry Flow with Heat and Mass Transfer.* AIAA Journal, pages 1–18, 2023. (Cited on pages 4 and 6.)

[Gottlieb & Shu 1998] Sigal Gottlieb and Chi-Wang Shu. *Total variation diminishing Runge-Kutta schemes.* Mathematics of computation, vol. 67, no. 221, pages 73–85, 1998. (Cited on page 27.)

[Grandhi & Roy 2017] Raj Kiran Grandhi and Arnab Roy. *Effectiveness of a reaction control system jet in a supersonic crossflow.* Journal of Spacecraft and Rockets, vol. 54, no. 4, pages 883–891, 2017. (Cited on page 49.)

[Gupta *et al.* 1990] Roop N Gupta, Jerrold M Yos, Richard A Thompson and Kam-Pui Lee. *A review of reaction rates and thermodynamic and transport properties for an 11-species air model for chemical and thermal nonequilibrium calculations to 30000 K.* Technical report 1232, NASA, 1990. (Cited on page 15.)

[Hakkinen *et al.* 1959] R J Hakkinen, I Greber, L Trilling and S S Abarbanel. *The interaction of an oblique shock wave with a laminar boundary layer.* Technical report, NASA, 1959. (Cited on page 46.)

[Hanifi *et al.* 1996] Ardeshir Hanifi, Peter J Schmid and Dan S Henningson. *Transient growth in compressible boundary layer flow.* Physics of Fluids, vol. 8, no. 3, pages 826–837, 1996. (Cited on page 58.)

[Hastie *et al.* 2009] Trevor Hastie, Robert Tibshirani and Jerome H Friedman. The elements of statistical learning: data mining, inference, and prediction, volume 2. Springer, 2009. (Cited on page 6.)

[Hawchar *et al.* 2017] Lara Hawchar, Charbel-Pierre El Soueidy and Franck Schoefs. *Principal component analysis and polynomial chaos expansion for time-variant reliability problems.* Reliability Engineering & System Safety, vol. 167, pages 406–416, 2017. (Cited on pages 6 and 72.)

[Heitmann & Radespiel 2013] Dirk Heitmann and Rolf Radespiel. *Simulations of boundary-layer response to laser-generated disturbances at Mach 6.* Journal of Spacecraft and Rockets, vol. 50, no. 2, pages 305–316, 2013. (Cited on page 43.)

[Hildebrand *et al.* 2018] Nathaniel Hildebrand, Anubhav Dwivedi, Joseph W Nichols, Mihailo R Jovanović and Graham V Candler. *Simulation and stability analysis of oblique shock-wave/boundary-layer interactions at Mach 5.92.* Physical Review Fluids, vol. 3, no. 1, page 013906, 2018. (Cited on pages 1, 46, 47 and 48.)

[Hirschfelder *et al.* 1964] Joseph Oakland Hirschfelder, Charles F Curtiss and R Byron Bird. Molecular theory of gases and liquids. Wiley, 1964. (Cited on pages 14 and 16.)

[Holden 1986] Michael S. Holden. *Review of Aerothermal Problems Associated With Hypersonic Flight.* In AIAA Paper, pages 1–40, Reston, Virigina, Jan 1986. American Institute of Aeronautics and Astronautics (AIAA). (Cited on page 1.)

[Huang & Duraisamy 2023] Cheng Huang and Karthik Duraisamy. *Predictive reduced order modeling of chaotic multi-scale problems using adaptively sampled projections.* arXiv preprint arXiv:2301.09006, 2023. (Cited on pages 7, 99 and 114.)

[Huang *et al.* 2005] Guang-Bin Huang, Paramasivan Saratchandran and Narasimhan Sundararajan. *A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation.* IEEE transactions on neural networks, vol. 16, no. 1, pages 57–67, 2005. (Cited on pages 7, 116 and 117.)

[Jeong & Hussain 1995] Jinhee Jeong and Fazle Hussain. *On the identification of a vortex.* Journal of fluid mechanics, vol. 285, pages 69–94, 1995. (Cited on page 55.)

[Jin *et al.* 2001] Ruichen Jin, Wei Chen and Timothy W Simpson. *Comparative studies of metamodelling techniques under multiple modelling criteria.* Structural and multidisciplinary optimization, vol. 23, no. 1, pages 1–13, 2001. (Cited on page 6.)

[Johnson & Candler 2005] Heath B. Johnson and Graham V. Candler. *Hypersonic boundary layer stability analysis using PSE-chem.* In 35th AIAA Fluid Dynamics Conference and Exhibit, Reston, Virginia, Jun 2005. American Institute of Aeronautics and Astronautics (AIAA). (Cited on page 2.)

[Johnson *et al.* 1998] Heath B. Johnson, Trevor G. Seipp and Graham V. Candler. *Numerical study of hypersonic reacting boundary layer transition on cones.*

Physics of Fluids, vol. 10, no. 10, pages 2676–2685, oct 1998. (Cited on page 2.)

[Kadirkamanathan & Niranjan 1993] Visakan Kadirkamanathan and Mahesan Niranjan. *A function estimation approach to sequential learning with neural networks.* Neural computation, vol. 5, no. 6, pages 954–975, 1993. (Cited on pages 7, 116 and 117.)

[Karagozian 2014] Ann R Karagozian. *The jet in crossflow.* Physics of Fluids, vol. 26, no. 10, pages 1–47, 2014. (Cited on page 50.)

[Karayiannis & Mi 1997] Nicolaos B Karayiannis and Glenn Weiqun Mi. *Growing radial basis neural networks: Merging supervised and unsupervised learning with network growth techniques.* IEEE Transactions on Neural networks, vol. 8, no. 6, pages 1492–1506, 1997. (Cited on pages 7 and 116.)

[Katzer 1989] Edgar Katzer. *On the lengthscales of laminar shock/boundary-layer interaction.* Journal of Fluid Mechanics, vol. 206, pages 477–496, 1989. (Cited on page 46.)

[Kawai & Lele 2008] Soshi Kawai and Sanjiva K. Lele. *Localized artificial diffusivity scheme for discontinuity capturing on curvilinear meshes.* Journal of Computational Physics, vol. 229, page 9498–9526, 2008. (Cited on pages 28 and 29.)

[Kawai & Lele 2010] Soshi Kawai and Sanjiva K Lele. *Large-eddy simulation of jet mixing in supersonic crossflows.* AIAA Journal, vol. 48, no. 9, pages 2063–2083, 2010. (Cited on pages 49, 52 and 60.)

[Kawai & Shankar 2010] Soshi Kawai and Sanjiva K. Shankar Santhosh K.and Lele. *Assessment of localized artificial diffusivity scheme for large-eddy simulation of compressible turbulent flows.* Journal of Computational Physics, vol. 227, page 1739–1762, 2010. (Cited on pages 28 and 29.)

[Kee *et al.* 2000] RJ Kee, FM Rupley, JA Miller, ME Coltrin, JF Grcar, E Meeks, HK Moffat, AE Lutz, G Dixon-Lewis, MD Smooke*et al.* *CHEMKIN collection, release 3.6, reaction design.* Inc., San Diego, CA, vol. 20, no. 0, page 0, 2000. (Cited on page 3.)

[Kennedy *et al.* 2000] Christopher A. Kennedy, Mark H. Carpenter and R.Michael Lewis. *Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations.* Applied Numerical Mathematics, vol. 35, no. 3, pages 177–219, 2000. (Cited on page 27.)

[Kleijnen 2009] Jack PC Kleijnen. *Kriging metamodeling in simulation: A review.* European journal of operational research, vol. 192, no. 3, pages 707–716, 2009. (Cited on pages 6 and 80.)

[Krige 1951] Daniel G Krige. *A statistical approach to some basic mine valuation problems on the Witwatersrand.* Journal of the Southern African Institute of Mining and Metallurgy, vol. 52, no. 6, pages 119–139, 1951. (Cited on page 6.)

[Lapeyre *et al.* 2019] Corentin J Lapeyre, Antony Misdariis, Nicolas Cazard, Denis Veynante and Thierry Poinsot. *Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates.* Combustion and Flame, vol. 203, pages 255–264, 2019. (Cited on page 5.)

[LeCun 1985] Yann LeCun. *Une procedure d'apprentissage pour reseau a seuil asymetrique.* Proceedings of Cognitiva 85, pages 599–604, 1985. (Cited on page 5.)

[Lees 1956] Lester Lees. *Laminar heat transfer over blunt-nosed bodies at hypersonic flight speeds.* Journal of Jet Propulsion, vol. 26, no. 4, pages 259–269, 1956. (Cited on pages 63, 64 and 72.)

[Lele 1992] Sanjiva K. Lele. *Compact finite difference schemes with spectral-like resolution.* Journal of Computational Physics, vol. 103, no. 1, pages 16–42, 1992. (Cited on page 23.)

[Leonard *et al.* 1992] JA Leonard, Mark A Kramer and LH Ungar. *A neural network architecture that computes its own reliability.* Computers & chemical engineering, vol. 16, no. 9, pages 819–835, 1992. (Cited on page 115.)

[Leyva 2017] Ivett A. Leyva. *The relentless pursuit of hypersonic flight.* Physics Today, vol. 70, no. 11, pages 30–36, Nov 2017. (Cited on page 1.)

[Liu & Batill 2000] Weiyu Liu and S Batill. *Gradient-enhanced neural network response surface approximations.* In 8th Symposium on Multidisciplinary Analysis and Optimization, page 4923, 2000. (Cited on page 5.)

[Lohninger 1993] Hans Lohninger. *Evaluation of neural networks based on radial basis functions and their application to the prediction of boiling points from structural parameters.* Journal of chemical information and computer sciences, vol. 33, no. 5, pages 736–744, 1993. (Cited on page 115.)

[Lu 2002] Frank K Lu. Advanced hypersonic test facilities, volume 198. AIAA, 2002. (Cited on page 1.)

[Lumley 1967] John Leask Lumley. *The structure of inhomogeneous turbulent flows.* Atmospheric turbulence and radio wave propagation, pages 166–178, 1967. (Cited on page 57.)

[Mack 1975] Leslie M Mack. *Linear stability theory and the problem of supersonic boundary-layer transition.* AIAA journal, vol. 13, no. 3, pages 278–289, 1975. (Cited on page 37.)

[Mack 1984] Leslie M Mack. *Boundary-layer linear stability theory.* Technical report, Jet Propulsion Lab, California Institute of Technology, Pasadena, CA, USA, 1984. (Cited on page 37.)

[MacQueen 1965] J MacQueen. *Some methods for classification and analysis of multivariate observations.* In Proc. 5th Berkeley Symposium on Math., Stat., and Prob, page 281, 1965. (Cited on page 117.)

[Magin & Degrez 2004] Thierry E Magin and Gérard Degrez. *Transport algorithms for partially ionized and unmagnetized plasmas.* Journal of computational Physics, vol. 198, no. 2, pages 424–449, 2004. (Cited on pages 14, 15 and 16.)

[Malik & Anderson 1991] Mujeeb R. Malik and E. C. Anderson. *Real gas effects on hypersonic boundary-layer stability.* Physics of Fluids A: Fluid Dynamics, vol. 3, no. 5, pages 803–821, may 1991. (Cited on pages 2, 18, 36 and 37.)

[Malik *et al.* 1990] Mujeeb R. Malik, T. A. Zang and David M. Bushnell. *Boundary layer transition in hypersonic flows.* In AIAA 2nd International Aerospace Planes Conference, 1990, pages 1–21, Reston, Virginia, Oct 1990. American Institute of Aeronautics and Astronautics (AIAA). (Cited on page 2.)

[Malik 1989] Mujeeb R. Malik. *Prediction and control of transition in supersonic and hypersonic boundary layers.* AIAA Journal, vol. 27, no. 11, pages 1487–1493, 1989. (Cited on page 2.)

[Mani *et al.* 2009] Ali Mani, Johan Larsson and Parviz Moin. *Suitability of artificial bulk viscosity for large-eddy simulation of turbulent flows with shocks.* Journal of Computational Physics, vol. 228, no. 19, pages 7368–7374, 2009. (Cited on pages 28 and 29.)

[Mao *et al.* 2021] Zhiping Mao, Lu Lu, Olaf Marxen, Tamer A Zaki and George Em Karniadakis. *DeepM&Mnet for hypersonics: Predicting the coupled flow and finite-rate chemistry behind a normal shock using neural-network approximation of operators.* Journal of computational physics, vol. 447, page 110698, 2021. (Cited on pages 4 and 6.)

[Martin 2007] M Pino Martin. *Direct numerical simulation of hypersonic turbulent boundary layers. Part 1. Initialization and comparison with experiments.* Journal of Fluid Mechanics, vol. 570, pages 347–364, 2007. (Cited on page 1.)

[Marxen *et al.* 2010] Olaf Marxen, Gianluca Iaccarino and Eric S. G. Shaqfeh. *Disturbance evolution in a Mach 4.8 boundary layer with two-dimensional roughness-induced separation and shock.* Journal of Fluid Mechanics, vol. 648, pages 435–469, 2010. (Cited on pages 43 and 45.)

[Marxen *et al.* 2011a] Olaf Marxen, Thierry Magin, Gianluca Iaccarino and Eric SG Shaqfeh. *A high-order numerical method to study hypersonic boundary-layer*

instability including high-temperature gas effects. Physics of Fluids, vol. 23, no. 8, page 084108, 2011. (Cited on page 2.)

[Marxen *et al.* 2011b] Olaf Marxen, Thierry E. Magin, Gianluca Iaccarino and Eric S. G. Shaqfeh. *A high-order numerical method to study hypersonic boundary-layer instability including high-temperature gas effects.* Physics of Fluids, vol. 23, no. 8, page 084108, Aug 2011. (Cited on pages 5, 18, 22 and 37.)

[Marxen *et al.* 2013] Olaf Marxen, Thierry E. Magin, Eric S. G. Shaqfeh and Gianluca Iaccarino. *A method for the direct numerical simulation of hypersonic boundary-layer instability with finite-rate chemistry.* Journal of Computational Physics, vol. 255, pages 572–589, 2013. (Cited on pages vii, 2, 16, 22, 35, 36, 37, 38, 39, 40, 41 and 42.)

[Marxen *et al.* 2014a] Olaf Marxen, Gianluca Iaccarino and Thierry E. Magin. *Direct numerical simulations of hypersonic boundary-layer transition with finite-rate chemistry.* Journal of Fluid Mechanics, vol. 755, pages 35–49, Sep 2014. (Cited on pages 22 and 37.)

[Marxen *et al.* 2014b] Olaf Marxen, Gianluca Iaccarino and Eric S. G. Shaqfeh. *Nonlinear instability of a supersonic boundary layer with two-dimensional roughness.* Journal of Fluid Mechanics, vol. 752, no. 4, pages 497–520, 2014. (Cited on pages 2, 22, 35 and 45.)

[McBride *et al.* 2002] Bonnie J McBride, Michael J Zehe and Gordon Sanford. *NASA Glenn coefficients for calculating thermodynamic properties of individual species.* Technical report 2002-211556, NASA, 2002. (Cited on page 14.)

[Miller *et al.* 2018] Warrick A Miller, Paul R Medwell, Con J Doolan and Minkwan Kim. *Transient interaction between a reaction control jet and a hypersonic crossflow.* Physics of Fluids, vol. 30, no. 4, page 046102, 2018. (Cited on pages 49, 52, 57 and 60.)

[Miró Miró & Pinna 2017] Fernando Miró Miró and Fabio Pinna. *Linear Stability Analysis of a Hypersonic Boundary Layer in Equilibrium and Non-Equilibrium.* In 47th AIAA Fluid Dynamics Conference, Reston, Virginia, Jun 2017. American Institute of Aeronautics and Astronautics (AIAA). (Cited on page 2.)

[Miró Miró & Pinna 2019] Fernando Miró Miró and Fabio Pinna. *On Decoupling Ablation Effects on Boundary-Layer Stability and Transition.* In AIAA Aviation 2019 Forum, Reston, Virginia, Jun 2019. American Institute of Aeronautics and Astronautics (AIAA). (Cited on page 2.)

[Miró Miró *et al.* 2018a] Fernando Miró Miró, Ethan S. Beyak, Daniel Mullen, Fabio Pinna and Helen L. Reed. *Ionization and Dissociation Effects on*

Hypersonic Boundary-Layer Stability. In 31st Congress of the International Council of the Aeronautical Sciences (ICAS) 2018, 2018. (Cited on page 2.)

[Miró Miró *et al.* 2018b] Fernando Miró Miró, Fabio Pinna, Ethan S. Beyak, Paolo Barbante and Helen L. Reed. *Diffusion and chemical non-equilibrium effects on hypersonic boundary-layer stability*. In 2018 AIAA Aerospace Sciences Meeting, Reston, Virginia, Jan 2018. American Institute of Aeronautics and Astronautics (AIAA). (Cited on page 2.)

[Mitchner & Kruger Jr 1973] Morton Mitchner and Charles H Kruger Jr. Partially ionized gases. John Wiley and Sons, Inc., New York, 1973. (Cited on page 14.)

[Morgan *et al.* 2010] Brandon Morgan, Soshi Kawai and Sanjiva Lele. *Large-eddy simulation of an oblique shock impinging on a turbulent boundary layer*. In 40th Fluid Dynamics Conference and Exhibit, page 4467, 2010. (Cited on pages 46 and 47.)

[Munafò *et al.* 2015] A Munafò, Y Liu and M Panesi. *Modeling of dissociation and energy transfer in shock-heated nitrogen flows*. Physics of Fluids, vol. 27, no. 12, page 127101, 2015. (Cited on page 73.)

[Munafò *et al.* 2020] Alessandro Munafò, Andrea Alberti, Carlos Pantano, Jonathan B Freund and Marco Panesi. *A computational model for nanosecond pulse laser-plasma interactions*. Journal of Computational Physics, vol. 406, page 109190, 2020. (Cited on page 3.)

[Nagarajan *et al.* 2003] S. Nagarajan, S. K. Lele and J. H. Ferziger. *A robust high-order compact method for large eddy simulation*. Journal of Computational Physics, vol. 191, no. 2, pages 392–419, 2003. (Cited on page 22.)

[Nagarajan 2004] S. Nagarajan. *Leading edge effects in bypass transition*. PhD thesis, Stanford University, 2004. (Cited on pages 22, 23, 24 and 27.)

[Nair *et al.* 2019] Vedanth Nair, Benjamin Wilde, Benjamin Emerson and Tim Lieuwen. *Shear layer dynamics in a reacting jet in crossflow*. Proceedings of the Combustion Institute, vol. 37, no. 4, pages 5173–5180, 2019. (Cited on page 50.)

[Newman 2006] Mark EJ Newman. *Modularity and community structure in networks*. Proceedings of the national academy of sciences, vol. 103, no. 23, pages 8577–8582, 2006. (Cited on pages 79 and 80.)

[Novello *et al.* 2022] Paul Novello, Gaël Poëtte, David Lugato, Simon Peluchon and Pietro Marco Congedo. *Accelerating hypersonic reentry simulations using deep learning-based hybridization (with guarantees)*. arXiv preprint arXiv:2209.13434, 2022. (Cited on pages 4 and 6.)

[Park 1989] Chul Park. *A review of reaction rates in high temperature air.* In 24th thermophysics conference, page 1740, 1989. (Cited on pages 16 and 35.)

[Passiatore *et al.* 2021] D. Passiatore, L. Sciacovelli, P. Cinnella and G. Pascazio. *Finite-rate chemistry effects in turbulent hypersonic boundary layers: A direct numerical simulation study.* Phys. Rev. Fluids, vol. 6, page 054604, May 2021. (Cited on pages 2, 4 and 42.)

[Passiatore *et al.* 2022] D. Passiatore, L. Sciacovelli, P. Cinnella and G. Pascazio. *Thermochemical non-equilibrium effects in turbulent hypersonic boundary layers.* Journal of Fluid Mechanics, vol. 941, page A21, 2022. (Cited on pages 2, 15, 99 and 110.)

[Passiatore *et al.* 2023] Donatella Passiatore, Luca Sciacovelli, Paola Cinnella and Giuseppe Pascazio. *Shock impingement on a transitional hypersonic high-enthalpy boundary layer.* Physical Review Fluids, vol. 8, no. 4, page 044601, 2023. (Cited on page 2.)

[Peherstorfer 2020] Benjamin Peherstorfer. *Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling.* SIAM Journal on Scientific Computing, vol. 42, no. 5, pages A2803–A2836, 2020. (Cited on pages 7 and 114.)

[Peter *et al.* 2007] J Peter, M Marcelet, S Burguburu and Valentino Pediroda. *Comparison of surrogate models for the actual global optimization of a 2D turbomachinery flow.* In Proceedings of the 7th WSEAS international conference on simulation, modelling and optimization, pages 46–51. Citeseer, 2007. (Cited on page 6.)

[Peterson *et al.* 2006] David Peterson, Pramod Subbareddy and Graham Candler. *Assessment of Synthetic Inflow Generation for Simulating Injection Into a Supersonic Crossflow.* In 14th AIAA/AHI Space Planes and Hypersonic Systems and Technologies Conference, page 8128, 2006. (Cited on pages 49, 52 and 60.)

[Pini *et al.* 2015] M Pini, Andrea Spinelli, G Persico and S Rebay. *Consistent look-up table interpolation method for real-gas flow simulations.* Computers & Fluids, vol. 107, pages 178–188, 2015. (Cited on pages 5 and 100.)

[Platt 1991] John Platt. *A resource-allocating network for function interpolation.* Neural computation, vol. 3, no. 2, pages 213–225, 1991. (Cited on pages 7, 116 and 117.)

[Pope 1997] Stephen B Pope. *Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation.* Combustion Theory and Modelling, vol. 1, pages 41–63, 1997. (Cited on pages 5 and 7.)

[Powell 1992] Michael JD Powell. *The theory of radial basis function approximation in 1990.* Advances in numerical analysis, pages 105–210, 1992. (Cited on page 80.)

[Ramezanian *et al.* 2021] Donya Ramezanian, Arash G Nouri and Hessam Babaee. *On-the-fly reduced order modeling of passive and reactive species via time-dependent manifolds.* Computer Methods in Applied Mechanics and Engineering, vol. 382, page 113882, 2021. (Cited on pages 7 and 114.)

[Ramshaw 1990] John D Ramshaw. *Self-consistent effective binary diffusion in multicomponent gas mixtures.* Journal of Non-Equilibrium Thermodynamics, vol. 15, pages 295–300, 1990. (Cited on page 16.)

[Rosenblatt 1956] Murray Rosenblatt. *Remarks on some nonparametric estimates of a density function.* The annals of mathematical statistics, pages 832–837, 1956. (Cited on page 105.)

[Sacks *et al.* 1989] Jerome Sacks, William J Welch, Toby J Mitchell and Henry P Wynn. *Design and analysis of computer experiments.* Statistical science, vol. 4, no. 4, pages 409–423, 1989. (Cited on page 6.)

[Santiago & Dutton 1997] Juan Gabriel Santiago and J Craig Dutton. *Velocity measurements of a jet injected into a supersonic crossflow.* Journal of propulsion and power, vol. 13, no. 2, pages 264–273, 1997. (Cited on page 49.)

[Sayadi *et al.* 2013] Taraneh Sayadi, Curtis W. Hamman and Parviz Moin. *Direct numerical simulation of complete H-type and K-type transitions with implications for the dynamics of turbulent boundary layers.* Journal of Fluid Mechanics, vol. 724, pages 480–509, 2013. (Cited on pages 22, 32, 53 and 99.)

[Scanlon *et al.* 2015] Thomas J Scanlon, Craig White, Matthew K Borg, Rodrigo C Palharini, Erin Farbar, Iain D Boyd, Jason M Reese and Richard E Brown. *Open-source direct simulation Monte Carlo chemistry modeling for hypersonic flows.* AIAA journal, vol. 53, no. 6, pages 1670–1680, 2015. (Cited on page 2.)

[Scherding *et al.* 2022] C Scherding, T Sayadi, C Williams, M Di Renzo, G Rigas and PJ Schmid. *Data-driven reduction of lookup libraries for the simulation of non-equilibrium effects in hypersonic flows.* In Center of Turbulence Research, Proceedings of the Summer Program, pages 57–66, 2022. (Cited on pages 94 and 101.)

[Scherding *et al.* 2023] Clément Scherding, Georgios Rigas, Denis Sipp, Peter J Schmid and Taraneh Sayadi. *Data-driven framework for input/output lookup tables reduction: Application to hypersonic flows in chemical nonequilibrium.* Physical Review Fluids, vol. 8, no. 2, page 023201, 2023. (Cited on page 71.)

[Schmid *et al.* 2002] Peter J Schmid, Dan S Henningson and DF Jankowski. *Stability and transition in shear flows. applied mathematical sciences, vol. 142.* Appl. Mech. Rev., vol. 55, no. 3, pages B57–B59, 2002. (Cited on page 58.)

[Schmid 2010] Peter J Schmid. *Dynamic mode decomposition of numerical and experimental data.* Journal of fluid mechanics, vol. 656, pages 5–28, 2010. (Cited on page 5.)

[Schmisseur 2015] John D. Schmisseur. *Hypersonics into the 21st century: A perspective on AFOSR-sponsored research in aerothermodynamics.* Progress in Aerospace Sciences, vol. 72, pages 3–16, jan 2015. (Cited on page 1.)

[Schneider 2008] Steven P. Schneider. *Summary of hypersonic boundary-layer transition experiments on blunt bodies with roughness.* Journal of Spacecraft and Rockets, vol. 45, no. 6, pages 1090–1105, 2008. (Cited on page 1.)

[Schwenker *et al.* 2001] Friedhelm Schwenker, Hans A Kestler and Günther Palm. *Three learning phases for radial-basis-function networks.* Neural networks, vol. 14, no. 4-5, pages 439–458, 2001. (Cited on page 82.)

[Scoggins & Magin 2014] James B. Scoggins and Thierry E. Magin. *Development of Mutation++: Multicomponent Thermodynamic and Transport Properties for Ionized Plasmas written in C++.* In 11th AIAA/ASME Joint Thermophysics and Heat Transfer Conference. American Institute of Aeronautics and Astronautics (AIAA), 2014. ID: AIAA 2014-2966. (Cited on pages 14 and 15.)

[Scoggins *et al.* 2020] James B. Scoggins, Vincent Leroy, Georgios Bellas-Chatzigeorgis, Bruno Dias and Thierry E. Magin. *Mutation++: MUlticomponent Thermodynamic And Transport properties for IONized gases in C++.* SoftwareX, vol. 12, page 100575, 2020. (Cited on pages 3, 13, 17, 31 and 72.)

[Scoggins 2017] James B. Scoggins. *Development of numerical methods and study of coupled flow, radiation, and ablation phenomena for atmospheric entry.* PhD thesis, Université Paris-Saclay and von Karman Institute for Fluid Dynamics, 2017. (Cited on page 14.)

[Scott 2015] David W Scott. Multivariate density estimation: theory, practice, and visualization. John Wiley & Sons, 2015. (Cited on page 105.)

[Semper *et al.* 2012] Michael Semper, Brandon Pruski and Rodney Bowersox. *Freestream turbulence measurements in a continuously variable hypersonic wind tunnel.* In 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, page 732, 2012. (Cited on page 47.)

[Shlens 2014] Jonathon Shlens. *A tutorial on principal component analysis.* arXiv:1404.1100 [cs.LG], 2014. (Cited on page 75.)

[Soize & Ghanem 2004] Christian Soize and Roger Ghanem. *Physical systems with random uncertainties: chaos representations with arbitrary probability measure.* SIAM Journal on Scientific Computing, vol. 26, no. 2, pages 395–410, 2004. (Cited on page 6.)

[Spaid & Zukoski 1968] FW Spaid and EE Zukoski. *A study of the interaction of gaseous jets from transverse slots with supersonic external flows.* AIAA journal, vol. 6, no. 2, pages 205–212, 1968. (Cited on pages 50 and 51.)

[Stemmer *et al.* 2017] Christian Stemmer, Marcel Birrer and Nikolaus A Adams. *Hypersonic boundary-layer flow with an obstacle in thermochemical equilibrium and nonequilibrium.* Journal of Spacecraft and Rockets, vol. 54, no. 4, pages 899–915, 2017. (Cited on page 43.)

[Sun & Wang 2019] Gang Sun and Shuyue Wang. *A review of the artificial neural network surrogate modeling in aerodynamic design.* Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, vol. 233, no. 16, pages 5863–5872, 2019. (Cited on page 80.)

[Taira *et al.* 2017] Kunihiko Taira, Steven L Brunton, Scott TM Dawson, Clarence W Rowley, Tim Colonius, Beverley J McKeon, Oliver T Schmidt, Stanislav Gordeyev, Vassilios Theofilis and Lawrence S Ukeiley. *Modal analysis of fluid flows: An overview.* Aiaa Journal, vol. 55, no. 12, pages 4013–4041, 2017. (Cited on pages 5, 57 and 58.)

[Theofilis *et al.* 2022] Vassilis Theofilis, Sergio Pirozzoli and Pino Martin. *Special issue on the fluid mechanics of hypersonic flight.* Theoretical and Computational Fluid Dynamics, vol. 36, no. 1, pages 1–8, 2022. (Cited on page 1.)

[Tumuklu *et al.* 2018] Ozgur Tumuklu, Vassilis Theofilis and Deborah A Levin. *On the unsteadiness of shock–laminar boundary layer interactions of hypersonic flows over a double cone.* Physics of Fluids, vol. 30, no. 10, page 106111, 2018. (Cited on page 51.)

[Urzay & Di Renzo 2021] J Urzay and M Di Renzo. *Engineering aspects of hypersonic turbulent flows at suborbital enthalpies.* Annual Research Briefs, Center for Turbulence Research, pages 7–32, 2021. (Cited on pages 2 and 3.)

[Van Driest & Blumer 1968] ER Van Driest and CB Blumer. *Boundary-layer transition at supersonic speeds-Roughness effects with heat transfer.* AIAA Journal, vol. 6, no. 4, pages 603–607, 1968. (Cited on page 43.)

[Volpiani 2021] Pedro Stefanin Volpiani. *Numerical strategy to perform direct numerical simulations of hypersonic shock/boundary-layer interaction in*

chemical nonequilibrium. Shock Waves, vol. 31, no. 4, pages 361–378, 2021. (Cited on pages 47 and 48.)

[Wartemann *et al.* 2018] Viola Wartemann, Alexander Wagner, Ross Wagnild, Fabio Pinna, Fernando Miró Miró and Hideyuki Tanno. *Code to code comparison on hypersonic high enthalpy transitional boundary layers.* In 2018 AIAA Aerospace Sciences Meeting, pages 1–14, Reston, Virginia, Jan 2018. American Institute of Aeronautics and Astronautics (AIAA). (Cited on page 2.)

[Wartemann *et al.* 2019] Viola Wartemann, Alexander Wagner, Ross Wagnild, Fabio Pinna, Fernando Miró Miró, Hideyuki Tanno and Heath B. Johnson. *High-Enthalpy Effects on Hypersonic Boundary-Layer Transition.* Journal of Spacecraft and Rockets, vol. 56, no. 2, pages 347–356, Mar 2019. (Cited on page 2.)

[Wegelin 2000] Jacob A. Wegelin. Technical report 371, University of washington, 2000. (Cited on page 77.)

[Wilke 1950] CR Wilke. *A viscosity equation for gas mixtures.* The journal of chemical physics, vol. 18, no. 4, pages 517–519, 1950. (Cited on page 15.)

[Williams *et al.* 2021] C Williams, M Di Renzo, P Moin and J Urzay. *Locally self-similar formulation for hypersonic laminar boundary layers in thermochemical nonequilibrium.* Annual Research Briefs, Center for Turbulence Research, pages 119–128, 2021. (Cited on pages 63, 64 and 66.)

[Wood 2003] Simon N Wood. *Thin plate regression splines.* Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 65, no. 1, pages 95–114, 2003. (Cited on page 81.)

[Yang 2003] Yuhong Yang. *Regression with multiple candidate models: selecting or mixing?* Statistica Sinica, pages 783–809, 2003. (Cited on page 6.)

[Yano *et al.* 2021] Masayuki Yano, Tianci Huang and Matthew J Zahr. *A globally convergent method to accelerate topology optimization using on-the-fly model reduction.* Computer Methods in Applied Mechanics and Engineering, vol. 375, page 113635, 2021. (Cited on pages 7 and 114.)

[Zanardi *et al.* 2022] Ivan Zanardi, Simone Venturi and Marco Panesi. *Towards efficient simulations of non-equilibrium chemistry in hypersonic flows: a physics-informed neural network framework.* In AIAA SCITECH 2022 Forum, page 1639, 2022. (Cited on pages 4 and 6.)

[Zanus *et al.* 2017] Ludovico Zanus, Fernando Miró Miró and Fabio Pinna. *Parabolized stability analysis of chemically reacting boundary layer flows in equilibrium conditions.* In 7th European Conference for Aeronautics and Aerospace Sciences (EUCASS) 2017, pages 1–15, 2017. (Cited on page 2.)

[Zdybał *et al.* 2022] Kamila Zdybał, Elizabeth Armstrong, James C Sutherland and Alessandro Parente. *Cost function for low-dimensional manifold topology assessment*. Scientific Reports, vol. 12, no. 1, page 14496, 2022. (Cited on page 100.)