Reduced-order models : convergence between scientific computing and data for fluid mechanics

Sébastien Riffaud

université de BORDEAUX

THÈSE

pour l'obtention du grade de

Docteur de l'Université de Bordeaux

École Doctorale de Mathématiques et d'Informatique

Spécialité : Mathématiques appliquées et calcul scientifique

présentée par

Sébastien Riffaud

Modèles réduits : convergence entre calcul et données pour la mécanique des fluides

---

sous la direction de **Angelo Iollo**

Soutenue le 18 décembre 2020

Membres du jury :

Rémi Abgrall	Professeur, Universität Zürich	Rapporteur (Pdt)
Andrea Ferrero	Professeur assistant, Politecnico di Torino	Invité
Angelo Iollo	Professeur, Université de Bordeaux	Directeur
Damiano Lombardi	Chargé de recherche, Inria Paris	Examinateur
Pierre Sagaut	Professeur, Aix-Marseille Université	Examinateur
Karen Veroy-Grepl	Professeure, Eindhoven University of Technology	Rapportrice

Modèles réduits : convergence entre calcul et données pour la mécanique des fluides

Résumé

L'objectif de cette thèse est de réduire significativement le coût de calcul associé aux simulations numériques gouvernées par des équations aux dérivées partielles. Dans ce but, nous considérons des modèles dits "réduits", dont la construction consiste typiquement en une phase d'apprentissage, au cours de laquelle des solutions haute-fidélité sont collectées pour définir un sous-espace d'approximation de faible dimension, et une étape de prédiction, qui exploite ensuite ce sous-espace d'approximation conduit par les données afin d'obtenir des simulations rapides voire en temps réel. La première contribution de cette thèse concerne la modélisation d'écoulements gazeux dans les régimes hydrodynamique et raréfié. Dans ce travail, nous développons une nouvelle approximation d'ordre réduite de l'équation de Boltzmann-BGK, basée sur la décomposition orthogonale aux valeurs propres dans la phase d'apprentissage et sur la méthode de Galerkin dans l'étape de prédiction. Nous évaluons la simulation d'écoulements instationnaires contenant des ondes de choc, des couches limites et des vortex en 1D et 2D. Les résultats démontrent la stabilité, la précision et le gain significatif des performances de calcul fournis par le modèle réduit par rapport au modèle haute-fidélité. Le second sujet de cette thèse porte sur les applications du problème de transport optimal pour la modélisation d'ordre réduite. Nous proposons notamment d'employer la théorie du transport optimal afin d'analyser et d'enrichir la base de données contenant les solutions haute-fidélité utilisées pour l'entraînement du modèle réduit. Les tests de reproduction et de prédiction d'écoulements gouvernés par l'équation de Boltzmann-BGK en 1D montrent l'amélioration de la précision et de la fiabilité du modèle réduit résultant de ces deux applications. Finalement, la dernière contribution de cette thèse concerne le développement d'une méthode de décomposition de domaine basée sur la méthode de Galerkin discontinue. Dans cette approche, le modèle haute-fidélité décrit la solution où un certain degré de précision est requis, tandis que le modèle réduit est employé dans le reste du domaine. La méthode de Galerkin discontinue pour le modèle réduit offre notamment une manière simple de reconstruire la solution globale en raccordant les solutions locales aux interfaces des cellules à travers les flux numériques. La méthode proposée est évaluée pour des problèmes paramétriques gouvernés par les équations d'Euler en 1D et 2D. Les résultats démontrent la précision de la méthode proposée et la réduction significative du coût de calcul par rapport aux simulations haute-fidélité.

Mots-clés : Réduction de modèles, Décomposition orthogonale aux valeurs propres, Écoulements raréfiés, Problème de transport optimal, Partitionnement de données, Décomposition de domaine, Méthode de Galerkin discontinue.

Reduced-order models: convergence between scientific computing and data for fluid mechanics

Abstract

The objective of this thesis is to significantly reduce the computational cost associated with numerical simulations governed by partial differential equations. For this purpose, we consider reduced-order models (ROMs), which typically consist of a training stage, in which high-fidelity solutions are collected to define a low-dimensional trial subspace, and a prediction stage, where this data-driven trial subspace is then exploited to achieve fast or real-time simulations. The first contribution of this thesis concerns the modeling of gas flows in both hydrodynamic and rarefied regimes. In this work, we develop a new reduced-order approximation of the Boltzmann-BGK equation, based on Proper Orthogonal Decomposition (POD) in the training stage and on the Galerkin method in the prediction stage. We investigate the simulation of unsteady flows containing shock waves, boundary layers and vortices in 1D and 2D. The results demonstrate the stability, accuracy and significant computational speedup factor delivered by the ROM with respect to the high-fidelity model. The second topic of this thesis deals with the optimal transport problem and its applications to model order reduction. In particular, we propose to use the optimal transport theory in order to analyze and enrich the training database containing the high-fidelity solution snapshots. The reproduction and prediction of unsteady flows governed by the 1D Boltzmann-BGK equation show the improvement of the accuracy and reliability of the ROM resulting from these two applications. Finally, the last contribution of this thesis concerns the development of a domain decomposition method based on the discontinuous Galerkin method. In this approach, the ROM approximates the solution where a significant dimensionality reduction can be achieved while the high-fidelity model is employed elsewhere. The discontinuous Galerkin method for the ROM offers a simple way to recover the global solution by linking the local solutions at cell interfaces through numerical fluxes. The proposed method is evaluated for parametric problems governed by the quasi-1D and 2D Euler equations. The results demonstrate the accuracy of the proposed method and the significant reduction of the computational cost with respect to the high-fidelity model.

Keywords: Model order reduction, Proper Orthogonal Decomposition, Rarefied flows, Optimal transport problem, Cluster analysis, Domain decomposition, Discontinuous Galerkin method.

Université de **B**ordeaux
École **D**octoral de **M**athématiques et d'**I**nformatique (ED 39)
Institut de **M**athématiques de **B**ordeaux (UMR 5251)

Acknowledgements

First of all, I would like to warmly thank my supervisor, Angelo Iollo, for giving me the opportunity to prepare this PhD thesis. You supported me over the last few years not only during this thesis, but also during my engineering internship, my second engineering internship, my undergraduate internship and my internship of excellence. Especially, you introduced me to the world of research, and it was a great adventure at your side.

I am also very grateful to Florian Bernard, Andrea Ferrero and Tommaso Taddei, with whom I talked many times about my work. You always took the time to answer my questions. Your expertise and advise was very helpful to me, and it was a great pleasure to work with you all.

I would like to sincerely thank Prof. Rémi Abgrall and Prof. Karen Veroy-Grepl for accepting to review my work. Their careful comments and suggestions helped me to improve this manuscript. I also thank Damiano Lombardi and Prof. Pierre Sagaut for their participation in my examining committee.

I would also like to warmly thank Prof. Charbel Farhat, Sebastian Grimberg and Spencer Anderson for welcoming me in their research team. It was a great experience to meet and work with you all. I really enjoyed the discussions, and I learned a lot from you.

I can't forget to thank my friends from my engineering school, bachelor's degree and high school, with whom I've spent so many great times. Many thanks also to the PhD students and post-docs of the MEMPHIS and MONC teams for bringing every day a pleasant and stimulating atmosphere in the laboratory.

A special thanks goes to Anne-Laure Gautier and to the administrative staff of the IMB for their help which greatly simplifies our daily life in the laboratory.

Thanks to the members of my thesis committee, Olivier Saut, Gautier Stauffer and François Vanderbeck, for their follow-up and advice.

The greatest thank-you goes to my family, and in particular, to my parents, Alain and Corinne, my grandparents, Christian and Jacqueline, my uncle and my aunt, Joel and Isabelle, and my brother and my sister, Julien and Estelle. You have always supported me during all these years, and I am proud of this nice family.

Contents

Introduction

Over the last few decades, numerical simulation has gained a growing interest in the fields of engineering and applied sciences. Thanks to the democratization of high-performance computing (HPC), numerical simulations currently provide an effective tool for solving models for which there is no simple analytical solution or experiments in real conditions are very expensive and difficult to perform. In addition, the constant increase in available computing power make nowadays possible the numerical modeling of complex, multiscale and multiphysics phenomena that were up to now inaccessible.

In numerical simulation, the dynamic of fluid flows or the deformation of mechanical structures are governed by mathematical models involving the resolution of partial-differential equations (PDEs). Since most of these equations are too complex to admit closed-form solutions, numerical methods are employed to transform the continuum problem into its discrete counterpart, leading to the resolution of a large-scale system. However, the computational complexity of the resulting high-dimensional model (HDM) can be problematic due to the large number of degrees of freedom $N \approx O(10^6, \ldots, 10^9)$ to be determined. In many industrial applications, efficient simulations are required, either due to runtime constraints in the case of extremely large-scale models or due to the large number of simulations to perform for different input parameters in the case of many-query problems.

This thesis aims at developing accurate and efficient reduced-order models (ROMs) in order to significantly decrease the computational complexity of the simulations. Instead of discretizing the solution without any knowledge about the dynamical system, ROMs [48, 81, 103, 49, 95, 34] use *a posteriori* information to considerably reduce the number of unknowns $M \approx O(10^1)$. The construction of ROMs is similar to the machine learning approach to achieve dimensionality reduction ($M \ll N$). It first consists of a training stage in which high-fidelity solutions are acquired for some training parameters to learn the system behaviour and to extract a low-dimensional trial subspace representing accurately the high-dimensional solution manifold. Then, during the prediction stage, the large-scale system is projected onto the test subspace, leading to the resolution of a small-scale system that enables fast or real-time simulations for new input parameters.

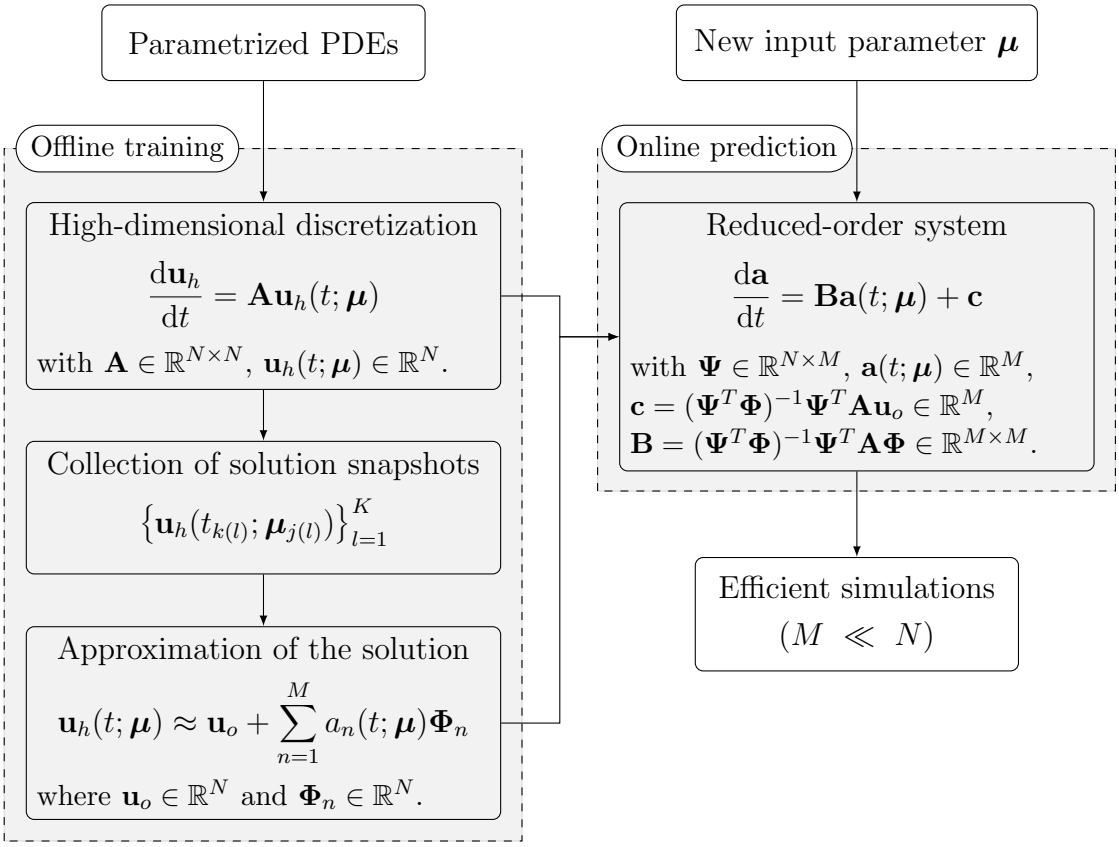


Figure 1: Schematic diagram illustrating the construction of ROMs.

The first contribution of this thesis concerns the simulation of gas flows in both hydrodynamic and rarefied regimes. In fluid dynamics, the regime of a gas flow is characterized by the Knudsen number Kn, defined as the ratio between the mean free path of the particles and the characteristic length of the problem. When the Knudsen number is low ($Kn \ll 1$), the gas particles are close to each other with respect to the characteristic length of the problem. The behaviour of the particles is similar to the macroscopic flow, and the regime is said hydrodynamic. Conversely in the rarefied regime ($Kn \gtrsim 1$), the behaviour of each particle can significantly differ from the macroscopic flow due to the large distance between the particles. For the simulation of hydrodynamic flows, it is generally sufficient to consider the macroscopic flow as in the Euler or Navier-Stokes equations. However in the rarified regime, this approach can fail to properly describe the dynamic of the fluid. In this work, we consider a model developed during the PhD thesis of F. Bernard [21] to simulate gas flows in both hydrodynamic and rarefied regimes. In this model, the dynamic of the gas flow is governed by the Boltzmann-BGK equation [35, 23], which is known to be sufficient for moderate and small Knudsen numbers ($Kn < 1$). However, the large number of degrees of freedom to be determined leads to a computationally expensive model, whose simulations require weeks on supercomputers. For this reason, we develop in this thesis a stable, ac-

curate and efficient ROM [22] which employs a new reduced-order approximation of the Boltzmann-BGK equation to considerably decrease the computational complexity of these simulations.

The second topic of this thesis is about the optimal transport problem [80, 66] and its applications to model order reduction [64, 22]. To construct an accurate and robust ROM over a wide range of input parameters, high-fidelity snapshots of the solution are collected at different time instances and input parameters in order to learn the solution manifold. However, the number of high-fidelity simulations for sampling the solution manifold is limited due to the expensive computational cost of the HDM. In particular, if the training snapshots are too different from the new predicted solution, the ROM may lead to unreliable predictions. In addition, since the trial subspace is constructed to approximate the previously collected snapshots, the accuracy of the ROM also depends on its ability to represent all these snapshots characterized by different physical regimes and moving features. For these reasons, we propose to employ the optimal transport problem in order to enrich and partition the snapshot database resulting from the sampling of the solution manifold. Notably, the optimal transport theory provides powerful tools to analyze and manipulate the snapshots of the solution. The transportation distance, commonly known as the Wasserstein distance, defines a robust metric to quantify the notion of proximity between two distribution functions. Compared to the classical L^2-norm which corresponds to the pointwise difference of the two distributions, the Wasserstein distance measures the minimal effort needed to push forward one distribution onto the other. In addition, this distance gives rise to realistic interpolations, referred to as Wasserstein barycenters, which preserve the features of the interpolated distribution functions.

Finally, the last contribution of this thesis concerns the development of a domain decomposition method for model order reduction [76, 73]. Perhaps the most common approach for constructing the low-dimensional trial subspace is the Proper Orthogonal Decomposition [87, 48, 103, 20], which hierarchically rearranges the high-dimensional solution manifold according to an energy criterion so that redundant information can be discarded to achieve dimensionality reduction. However, the nature of the problem strongly determines the extent to which one can reduce the dimensionality of the trial subspace. As the problem parameters are varied, singular solution features (e.g. discontinuities and fronts) or compact support phenomena can change their position and shape such that dimensionality reduction is limited. In this work [92], we adopt the strategy of employing the ROM only in those subdomains where a significant dimensionality reduction can be achieved. Instead of modeling the flow by a global ROM, the fluid problem is spatially partitioned to isolate the subdomains containing shocks or compact support phenomena. Local ROMs then predict the solution where a low-dimensional trial subspace is sufficiently accurate, while the HDM is employed elsewhere.

In addition to the introductory and concluding chapters, this manuscript consists of four main chapters organized as follows.

1. The first **Chapter** I gives a quick overview of the model order reduction framework. The goal is to present the main techniques later employed in this manuscript for the construction of ROMs. In particular, we introduce the Proper Orthogonal Decomposition (POD) and the Petrov-Galerkin method used during the training and prediction stages, respectively. In addition, we present hyper-reduction techniques to deal with nonlinear problems.

2. The first contribution of this thesis is presented in the second **Chapter** II. In this work, we develop a new reduced-order approximation of the Boltzmann-BGK equation to significantly decrease the computational cost associated with the numerical simulation of gas flows in both hydrodynamic and rarefied regimes. To this end, we adopt an approach based on POD in the training stage and on the Galerkin method in the prediction stage. This approach is then adapted to the case of the Boltzmann-BGK equation. The performance of the resulting ROM is evaluated on the simulation of unsteady flows governed by the Boltzmann-BGK equation in 1D and 2D.

3. In the third **Chapter** III, we presents two applications of the optimal transport problem in order to improve the ROM described in **Chapter** II. In the first application, the snapshot database is enriched with additional snapshots interpolated by optimal transport. These artificial snapshots allow to complete the sampling of the solution manifold in order to perform reliable predictions. In the second application, the snapshot database is partitioned into clusters by the k-means algorithm combined with the Wasserstein distance. The solution is then represented by several local trial subspaces, which are more appropriate and accurate than a single global trial subspace to approximate the snapshots associated with each cluster. These two applications are evaluated on the reproduction and prediction of shock waves described by the 1D Boltzmann-BGK equation.

4. The fourth **Chapter** IV presents a domain decomposition method based on the discontinuous Galerkin method. In this approach, the ROM approximates the solution in regions where significant dimensionality reduction can be achieved while the HDM is employed elsewhere. The Discontinuous Galerkin (DG) method for the ROM offers a simple way to recover the global solution by linking the local solutions at the interface of subdomains though the numerical flux. Compared to the standard DG method, the polynomial shape functions are replaced by empirical modes constructed by POD during the training stage. The performance of the resulting method is evaluated on the prediction of unsteady flows governed by the quasi-1D and 2D Euler equations in the presence of shocks.

Chapter I

Model order reduction

I.1 Introduction

In numerical simulation, the dynamic of fluid flows or the deformation of mechanical structures are governed by mathematical models involving the resolution of parametrized partial-differential equations (PDEs). Since most of these equations are too complex to admit simple analytical solutions, numerical methods are employed to transform the continuum problem into its discrete counterpart, leading to the resolution of a large-scale system. However, the computational complexity of the resulting high-dimensional model (HDM) can be problematic due to the large number of degrees of freedom $N \approx O(10^6, \dots, 10^9)$ to be determined. In many industrial applications, efficient simulations are required, either due to runtime constraints in the case of extremely large-scale models or due to the large number of simulations to perform for different input parameters in the case of many-query problems.

Reduced-order models (ROMs) [48, 81, 103, 49, 95, 33] have been developed in order to decrease the computational complexity of the simulations. Instead of discretizing the solution without any knowledge about the dynamical system, ROMs use *a posteriori* information to drastically reduce the number of unknowns $M \approx O(10^1)$. The construction of ROMs is similar to the machine learning approach to achieve dimensionality reduction ($M \ll N$). It first consists of a training stage in which high-fidelity solutions are acquired for some training parameters to learn the system behaviour and to extract a low-dimensional trial subspace representing accurately the solution manifold. Then, during the prediction stage, the large-scale system is projected onto the test subspace, leading to the resolution of a small-scale system that enables fast or real-time simulations for new input parameters. Moreover, in the case of nonlinear systems, an additional hyper-reduction approximation is introduced to ensure the computational complexity of the ROM is independent of the dimension N of the HDM.

This chapter presents the main techniques later employed in this manuscript for the construction of ROMs. In Section I.2, we introduce the HDM resulting from the discretization of the parametrized PDEs. Then, Section I.3 describes the Petrov-Galerkin method employed in the prediction stage to obtain the reduced-order system. In Section I.4, we present the Proper Orthogonal Decomposition allowing during the training stage to find the low-dimensional trial subspace representing accurately the solution manifold. Finally, Section I.5 details hyper-reduction techniques for the model order reduction of nonlinear problems.

I.2 High-dimensional model

In numerical simulation, the dynamic of fluid flows or the deformation of mechanical structures are described by mathematical models involving the resolution of parametric PDEs. In particular, we will focus on the Euler and Boltzmann equations in the next chapters. These models depends on input parameters $\boldsymbol{\mu}$, which may characterize geometric features of the domain, fluid or material properties, or initial and boundary conditions. The PDEs connect the input parameters to the dynamical system solution and possibly to some outputs of interest. The solution may represent, for example, the deformation of a structure or fluid quantities such as the density, the velocity and the pressure. Since most of PDEs do not admit closed-form solutions, numerical methods are used to transform the continuum problem into its discrete counterpart. This discretization step leads to the resolution of a large-scale system often referred to as the high-dimensional model (HDM). This system can be solved with high accuracy, and its solution is seen as the high-fidelity solution of the PDEs.

I.2.1 Parametrized partial differential equations

Let the parameter domain $\mathcal{D} \subset \mathbb{R}^p$ be a closed and bounded subset of the Euclidean space $\mathbb{R}^p$ with $p \in \mathbb{N}^*$. Moreover, let $\Omega \subset \mathbb{R}^d$ be a regular open domain, where $d \in \{1, 2, 3\}$ is the space dimension. We consider the parametric, time-dependent, partial-differential equation for $\mathbf{x} \in \Omega$, $t \in \mathbb{R}^*_+$ and $\boldsymbol{\mu} \in \mathcal{D}$:

$$\frac{\partial u}{\partial t} + \mathcal{L}[u] = 0, \tag{I.1}$$

subject to appropriate initial and boundary conditions. Here, $\mathbf{x}$ denotes the space variable, t denotes the time, $\boldsymbol{\mu}$ denotes the input parameters, $u : \Omega \times \mathbb{R}^*_+ \times \mathcal{D} \to \mathbb{R}$ denotes the exact solution, belonging to a suitable functional space $\mathcal{V}(\Omega)$, and $\mathcal{L}[u]$ denotes the spatial differential operator containing, for example, the convective, diffusive and source terms.

I.2.2 Numerical methods

Since the PDE (I.1) does not admit analytical solutions in general, numerical methods are used to transform the continuum problem into its discrete counterpart. The domain Ω is first partitioned into a conforming mesh of non-overlapping elements K_i:

$$\Omega = \bigcup_i K_i \qquad \text{and} \qquad K_i \bigcap_{i \neq j} K_j = \emptyset.$$

This partition depends on the parameter h, defined as the maximum diameter of the mesh elements. In this manuscript, the elements are, for instance, intervals (1D), squares (2D) or triangles (2D). On each element, the exact solution is approximated by polynomial shape functions:

$$u_h \in \mathcal{V}_h(\Omega) \coloneqq \{u \in \mathcal{V}(\Omega), \text{ such that } u_{|K_i} \in \mathcal{P}(K_i)\},$$

where $\mathcal{P}$ denotes the space of polynomial functions and $u_h(\mathbf{x}, t; \boldsymbol{\mu})$ denotes the discrete solution representing the exact solution at point $\mathbf{x}$, time instance t and input parameter $\boldsymbol{\mu}$. Moreover, the discrete solution is encoded as the vector $\mathbf{u}_h(t; \boldsymbol{\mu}) = (u_h(\mathbf{x}_1, t; \boldsymbol{\mu}), \dots, u_h(\mathbf{x}_N, t; \boldsymbol{\mu}))^T \in \mathbb{R}^N$ with N the number of degrees of freedom. The spatial operator is then discretized by, for example, the finite difference (FD), finite element (FE), finite volume (FV) or discontinuous Galerkin (DG) method, leading to the semi-discrete system

$$\frac{\mathrm{d}\mathbf{u}_h}{\mathrm{d}t} = \mathbf{f}_h[\mathbf{u}_h](t; \boldsymbol{\mu}), \tag{I.2}$$

where $\mathbf{f}_h[\mathbf{u}_h]$ denotes the discretization of the spatial differential operator $\mathcal{L}[u_h]$ and is encoded as the vector $\mathbf{f}_h(t; \boldsymbol{\mu}) = (f_h(\mathbf{x}_1, t; \boldsymbol{\mu}), \dots, f_h(\mathbf{x}_N, t; \boldsymbol{\mu}))^T \in \mathbb{R}^N$. The time is finally discretized by a linear multistep scheme or a Runge-Kutta scheme, leading at each time-step t_k to the resolution of the large-scale $N \times N$ system

$$\mathbf{r}_h[\mathbf{u}_h](t_k; \boldsymbol{\mu}) = 0, \tag{I.3}$$

where $\mathbf{r}_h[\mathbf{u}_h]$ denotes the high-dimensional residual and is encoded as the vector $\mathbf{r}_h(t; \boldsymbol{\mu}) = (r_h(\mathbf{x}_1, t; \boldsymbol{\mu}), \dots, r_h(\mathbf{x}_N, t; \boldsymbol{\mu}))^T \in \mathbb{R}^N$. This system is referred to as the high-dimensional model (HDM) in the following. We assume that the HDM can be solved with high accuracy, and its solution is considered to be the high-fidelity solution to the continuum problem (I.1).

I.3 Projection based-reduced order model

The HDM involves a large number of degrees of freedom $N \approx O(10^6, \dots, 10^9)$ to achieve accurate simulations. The computational complexity of the HDM can therefore be problematic due to the resolution of the large-scale $N \times N$ system

(I.3) at each time-step. Notably, many industrial applications require efficient simulations, either due to runtime constraints in the case of extremely large-scale models or due to the large number of simulations to perform for different input parameters in the case of many-query problems. In this context, reduced-order models [109, 44, 57, 18, 4, 5, 38] have been developed in order to reduce the number of unknowns $M \approx O(10^1)$ and thus decrease the computational complexity of the simulations.

I.3.1 Solution approximation

Instead of approximating the solution belonging to the high-dimensional space $\mathcal{V}_h(\Omega)$ without any knowledge about the dynamical system, the ROM uses *a posteriori* information to find a low-dimensional trial subspace $\mathcal{S}_h(\Omega) \subset \mathcal{V}_h(\Omega)$ where the solution is searched. To reduce the number of degrees of freedom ($M \ll N$), the discrete solution is approximated by

$$\widetilde{u}_h(\mathbf{x}, t; \boldsymbol{\mu}) = u_o(\mathbf{x}) + \sum_{n=1}^{M} a_n(t; \boldsymbol{\mu}) \Phi_n(\mathbf{x}). \tag{I.4}$$

Here, the offset u_o and the basis functions Φ_n span the affine trial subspace $\mathcal{S}_h(\Omega)$, and a_n denote the reduced coordinates of the approximate solution $\widetilde{u}_h \in \mathcal{S}_h(\Omega)$ in this subspace. By introducing the vectors $\mathbf{u}_o \in \mathbb{R}^N$ and $\mathbf{a}(t; \boldsymbol{\mu}) \in \mathbb{R}^M$ containing the offset and the reduced coordinates, respectively, and the matrix $\boldsymbol{\Phi} \in \mathbb{R}^{N \times M}$ containing the basis functions, the approximate solution can be written in matrix format as follows

$$\widetilde{\mathbf{u}}_h(t; \boldsymbol{\mu}) = \mathbf{u}_o + \boldsymbol{\Phi} \mathbf{a}(t; \boldsymbol{\mu}),$$

where

$$\mathbf{u}_o = \begin{pmatrix} u_o(\mathbf{x}_1) \\ u_o(\mathbf{x}_2) \\ \vdots \\ u_o(\mathbf{x}_N) \end{pmatrix}, \quad \boldsymbol{\Phi} = \begin{pmatrix} \Phi_1(\mathbf{x}_1) & \Phi_2(\mathbf{x}_1) & \cdots & \Phi_M(\mathbf{x}_1) \\ \Phi_1(\mathbf{x}_2) & \Phi_2(\mathbf{x}_2) & \cdots & \Phi_M(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_1(\mathbf{x}_N) & \Phi_2(\mathbf{x}_N) & \cdots & \Phi_M(\mathbf{x}_N) \end{pmatrix}, \quad \mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_M \end{pmatrix}.$$

The offset and the basis functions are constructed offline during the training stage, while the reduced coordinates are computed online in the prediction stage. To define the offset, there are three popular choices:

1. no offset:
$$u_o(\mathbf{x}) = 0;$$

2. the initial solution (if this one does not depend on the input parameters $\boldsymbol{\mu}$):
$$u_o(\mathbf{x}) = u_0(\mathbf{x}; \boldsymbol{\mu});$$

3. the mean solution over time and parameter space:

$$u_o(\mathbf{x}) = \frac{1}{|\mathcal{D}|(t_{max} - t_0)} \int_{\mathcal{D}} \int_{t_0}^{t_{max}} u_h(\mathbf{x}, t; \boldsymbol{\mu}) \, \mathrm{d}t \, \mathrm{d}\boldsymbol{\mu}.$$

The basis functions are then constructed by Proper Orthogonal Decomposition, which allows to extract, from the high-dimensional solution manifold $\mathcal{V}_h(\Omega)$, the low-dimensional trial subspace $\mathcal{S}_h(\Omega)$ that is optimal in the least-squares sense to approximate the solution. Once the affine trial subspace (i.e. the offset and the basis functions) is defined, the approximate solution depends only on the reduced coordinates during the prediction stage. In this way, the number of degrees of freedom is significantly reduced ($M \ll N$), enabling fast simulations for new input parameters.

I.3.2 Petrov-Galerkin method

In the prediction stage, the reduced coordinates $a_n(t; \boldsymbol{\mu})$ are determined at low cost by the Petrov-Galerkin method. By inserting the approximate solution (I.4) into PDE (I.1), we obtain the residual

$$r[\widetilde{u}_h] = \frac{\partial \widetilde{u}_h}{\partial t} + \mathcal{L}[\widetilde{u}_h]. \tag{I.5}$$

In the Petrov-Galerkin method, this residual (I.5) is enforced to be orthogonal to the test subspace. To this end, the solution manifold $\mathcal{V}_h(\Omega)$ is endowed with the inner product $\langle \cdot, \cdot \rangle_{\boldsymbol{\Theta}}$ associated with the norm $\|\cdot\|_{\boldsymbol{\Theta}} = \sqrt{\langle \cdot, \cdot \rangle_{\boldsymbol{\Theta}}}$. The inner product is induced by the symmetric positive-definite (SPD) matrix $\boldsymbol{\Theta} \in \mathbb{R}^{N \times N}$:

$$\langle v_1(\mathbf{x}), v_2(\mathbf{x}) \rangle_{\boldsymbol{\Theta}} \coloneqq \mathbf{v}_1^T \boldsymbol{\Theta} \mathbf{v}_2,$$

where $\mathbf{v} = (v(\mathbf{x}_1), v(\mathbf{x}_2), \ldots, v(\mathbf{x}_N))^T \in \mathbb{R}^N$. In this manuscript, we will mainly consider the L^2-norm, and $\boldsymbol{\Theta}$ will correspond to the diagonal matrix containing the weights of the quadrature rule on the diagonal. The projection of the residual (I.5) onto the test subspace leads to the system of M equations

$$\forall n \in \{1, \ldots, M\} : \ \langle r[\widetilde{u}_h](\mathbf{x}, t; \boldsymbol{\mu}), \Psi_n(\mathbf{x}) \rangle_{\boldsymbol{\Theta}} = 0, \tag{I.6}$$

where Ψ_n denote the test functions spanning the test subspace. There are two popular choices to define the test subspace:

1. the Galerkin method [81, 94, 96, 11]:

$$\langle r[\widetilde{u}_h](\mathbf{x}, t; \boldsymbol{\mu}), \Phi_n(\mathbf{x}) \rangle_{\boldsymbol{\Theta}} = 0,$$

   wherein the test functions are set to the basis functions, i.e. $\Psi_n = \Phi_n$;

2. the residual minimization method [33, 34, 2, 56]:

$$\underset{\mathbf{a}(t_k, \boldsymbol{\mu}) \in \mathbb{R}^M}{\text{minimize}} \ \| r_h[\widetilde{u}_h](\mathbf{x}, t_k; \boldsymbol{\mu}) \|_{\boldsymbol{\Theta}}^2 ,$$

   wherein the test functions are chosen at each time-step t_k in order to minimize the $\boldsymbol{\Theta}$-norm of the high-dimensional residual $r_h[\widetilde{u}_h]$.

I.3.2.1 Galerkin method

In the Galerkin method, the residual is enforced to be orthogonal to the trial subspace. Inserting the approximate solution into PDE (I.1) and projecting the resulting equation onto the basis functions yield after semi-discretization to the system of ODEs

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = \boldsymbol{\Phi}^T \boldsymbol{\Theta} \mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu}), \tag{I.7}$$

where we assume the basis functions are orthonormal, i.e. $\boldsymbol{\Phi}^T\boldsymbol{\Theta}\boldsymbol{\Phi} = \mathbf{I}_M$. This system is then discretized in time, leading to the small-scale $M \times M$ system

$$\boldsymbol{\Phi}^T \boldsymbol{\Theta} \mathbf{r}_h[\widetilde{\mathbf{u}}_h](t_k;\boldsymbol{\mu}) = 0.$$

The drawback of this approach is that the Galerkin projection may lead to unstable ROM, even if the HDM is stable, according to [95]. Consider, for example, the linear time invariant system

$$\frac{\mathrm{d}\mathbf{u}_h}{\mathrm{d}t} = \mathbf{A}(\boldsymbol{\mu})\mathbf{u}_h(t;\boldsymbol{\mu}), \tag{I.8}$$

where the real part of the eigenvalues of $\mathbf{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N\times N}$ is negative, that is, the HDM is stable. By applying the Galerkin projection, we obtain

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = \boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\mu})\boldsymbol{\Phi}\mathbf{a}(t;\boldsymbol{\mu}).$$

If $\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\mu})$ is a symmetric negative-definite matrix, it follows

$$\frac{\mathrm{d}\left\|\mathbf{a}\right\|_2^2}{\mathrm{d}t} = -2\left\|\boldsymbol{\Phi}\mathbf{a}(t;\boldsymbol{\mu})\right\|^2_{-\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\mu})} \leqslant 0,$$

meaning the ROM is stable. Notably in [95, 11], the inner product is defined so as to obtain a stable ROM formulation. However, if $\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\mu})$ is not symmetric negative-definite, the real part of the eigenvalues of $\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\mu})\boldsymbol{\Phi}$ may be strictly positive, leading to an unstable ROM. To overcome this issue, the residual minimisation method was developed in order to generate stable ROM.

I.3.2.2 Residual minimization method

In the residual minimization method, the test subspace is defined at each time-step t_k so that $\widetilde{u}_h$ minimizes the high-dimensional residual (I.3) in the $\boldsymbol{\Theta}$-norm:

$$\underset{\mathbf{a}(t_k,\boldsymbol{\mu})\in\mathbb{R}^M}{\text{minimize}} \ \left\|r_h[\widetilde{u}_h](\mathbf{x},t_k;\boldsymbol{\mu})\right\|^2_{\boldsymbol{\Theta}}. \tag{I.9}$$

The first-order necessary condition for optimality is

$$\boldsymbol{\Phi}^T \left(\mathbf{J}_h[\widetilde{\mathbf{u}}_h](t_k;\boldsymbol{\mu})\right)^T \boldsymbol{\Theta}\mathbf{r}_h[\widetilde{\mathbf{u}}_h](t_k;\boldsymbol{\mu}) = 0,$$

where $\mathbf{J}_h[\mathbf{u}_h](t_k;\boldsymbol{\mu}) = \dfrac{\partial \mathbf{r}_h}{\partial \mathbf{u}_h}[\mathbf{u}_h](t_k;\boldsymbol{\mu}) \in \mathbb{R}^{N\times N}$ denotes the HDM residual Jacobian. In particular, the residual minimization method is a special case of the Petrov-Galerkin method wherein the test functions are defined by $\boldsymbol{\Psi}_n = \mathbf{J}_h[\widetilde{\mathbf{u}}_h](t_k;\boldsymbol{\mu})\boldsymbol{\Phi}_n$. The minimization problem (I.9) can be solved in practice by the Gauss-Newton method as in the least-squares Petrov-Galerkin (LSPG) method [33, 2, 56].

Under some conditions [32], the Galerkin and residual minimization methods are equivalent. In the case of time explicit discretization, the residual minimization method reduces to the Galerkin method since $\mathbf{J}_h[\widetilde{\mathbf{u}}_h](t_k;\boldsymbol{\mu}) \propto \mathbf{I}_N$. Also when $\boldsymbol{\Theta}\mathbf{J}_h[\widetilde{\mathbf{u}}_h](t_k;\boldsymbol{\mu})$ is SPD, the Galerkin method minimizes the high-dimensional residual in the norm induced by the inner product defined by the matrix $\boldsymbol{\Theta}\mathbf{J}_h[\widetilde{\mathbf{u}}_h](t_k;\boldsymbol{\mu})$.

I.3.3 Error analysis

In the Galerkin method, pre-multipling the semi-discrete system of ODEs (I.7) by the basis functions leads to the equivalent system verified by the approximate solution:

$$\frac{\mathrm{d}\widetilde{\mathbf{u}}_h}{\mathrm{d}t} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu}). \tag{I.10}$$

Notably, an estimate of the error $e(\mathbf{x},t;\boldsymbol{\mu})$ between the solution and its approximation is derived in [89]. By introducing the orthogonal projection $\widehat{u}_h \in \mathcal{S}_h(\Omega)$ of the discrete solution u_h onto the trial subspace

$$\widehat{u}_h(\mathbf{x},t;\boldsymbol{\mu}) = u_o(\mathbf{x}) + \sum_{n=1}^{M} \langle u_h(\mathbf{x},t;\boldsymbol{\mu}) - u_o(\mathbf{x}), \Phi_n(\mathbf{x})\rangle_{\boldsymbol{\Theta}}\, \Phi_n(\mathbf{x}), \tag{I.11}$$

where we assume the basis functions are orthonormal, the error can be decomposed into one component $e_{\mathcal{S}}(\mathbf{x},t;\boldsymbol{\mu}) \in \mathcal{S}_h(\Omega)$ belonging to the trial subspace and one component $e_{\mathcal{S}^\perp}(\mathbf{x},t;\boldsymbol{\mu}) \in \mathcal{S}_h^\perp(\Omega)$ belonging to the orthogonal complement of the trial subspace:

$$\begin{aligned} e(\mathbf{x},t;\boldsymbol{\mu}) &= u_h(\mathbf{x},t;\boldsymbol{\mu}) - \widetilde{u}_h(\mathbf{x},t;\boldsymbol{\mu}) \\ &= \underbrace{u_h(\mathbf{x},t;\boldsymbol{\mu}) - \widehat{u}_h(\mathbf{x},t;\boldsymbol{\mu})}_{e_{\mathcal{S}^\perp}(\mathbf{x},t;\boldsymbol{\mu})} + \underbrace{\widehat{u}_h(\mathbf{x},t;\boldsymbol{\mu}) - \widetilde{u}_h(\mathbf{x},t;\boldsymbol{\mu})}_{e_{\mathcal{S}}(\mathbf{x},t;\boldsymbol{\mu})}. \end{aligned}$$

The first term $e_{\mathcal{S}^\perp}(\mathbf{x},t;\boldsymbol{\mu})$ represents the projection error between the solution and its orthogonal projection onto the trial subspace. It shows the importance of choosing the trial subspace to best represent the solution manifold (e.g. by Proper Orthogonal Decomposition). This term is orthogonal to the trial subspace, and its projection onto the trial subspace verifies $\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{e}_{\mathcal{S}^\perp}(t;\boldsymbol{\mu}) = 0$. The second term $e_{\mathcal{S}}(\mathbf{x},t;\boldsymbol{\mu})$ represents the modeling error between the HDM (I.2) and the equivalent system (I.10). This term is parallel to the trial subspace, and its orthogonal projection onto the trial subspace verifies $\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{e}_{\mathcal{S}}(t;\boldsymbol{\mu}) = \mathbf{e}_{\mathcal{S}}(t;\boldsymbol{\mu})$. The term

$e_{\mathcal{S}^\perp}(\mathbf{x}, t; \boldsymbol{\mu})$ can be estimated without executing the ROM, and then $e_{\mathcal{S}}(\mathbf{x}, t; \boldsymbol{\mu})$ can be estimated from $e_{\mathcal{S}^\perp}(\mathbf{x}, t; \boldsymbol{\mu})$ by solving the initial value problem

$$\frac{\mathrm{d}\mathbf{e}_{\mathcal{S}}}{\mathrm{d}t} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\Theta}\Big(\mathbf{f}_h[\mathbf{u}_h](t; \boldsymbol{\mu}) - \mathbf{f}_h[\mathbf{u}_h + \mathbf{e}_{\mathcal{S}^\perp} + \mathbf{e}_{\mathcal{S}}](t; \boldsymbol{\mu})\Big),$$

where $e_{\mathcal{S}}(\mathbf{x}, t_0; \boldsymbol{\mu}) = 0$ since $\widetilde{u}_h(\mathbf{x}, t_0; \boldsymbol{\mu}) = \widehat{u}_h(\mathbf{x}, t_0; \boldsymbol{\mu})$. In addition, if the initial solution verifies $\widetilde{u}_h(\mathbf{x}, t_0; \boldsymbol{\mu}) = u_h(\mathbf{x}, t_0; \boldsymbol{\mu})$ (e.g. by defining $u_o(\mathbf{x}) = u_h(\mathbf{x}, t_0; \boldsymbol{\mu})$), then $e_{\mathcal{S}^\perp}(\mathbf{x}, t_0; \boldsymbol{\mu}) = 0$ and the initial error is zero. In the case of the linear time invariant system (I.8), the error is reduced in particular to

$$\frac{\mathrm{d}\mathbf{e}_{\mathcal{S}}}{\mathrm{d}t} = \boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\mu})\mathbf{e}_{\mathcal{S}}(t; \boldsymbol{\mu}) + \boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\mu})\mathbf{e}_{\mathcal{S}^\perp}(t; \boldsymbol{\mu}),$$

where $\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{A}(\boldsymbol{\mu})\mathbf{e}_{\mathcal{S}^\perp}(t; \boldsymbol{\mu})$ acts as a forcing term.

I.4 Proper Orthogonal Decomposition

The basis functions spanning the trial subspace $\mathcal{S}_h(\Omega)$ are constructed offline during the training stage. The Proper Orthogonal Decomposition (POD) [87] is a popular dimensionality reduction method used in model reduction [81, 114, 95, 33] to define the trial subspace. It was first introduced in the context of the simulation of turbulent flows [103, 20] to find the coherent structures of the solution. In this approach, snapshots of the high-fidelity solutions are first acquired for some training parameters to learn the solution manifold. Then, the POD allows to extract, from the high-dimensional solution manifold $\mathcal{V}_h(\Omega)$, the low-dimensional affine trial subspace $\mathcal{S}_h(\Omega) \subset \mathcal{V}_h(\Omega)$ that is optimal in the least-squares sense to approximate the solution snapshots. This optimization problem is known as the low-rank approximation problem and is solved by the Schmidt-Eckart-Young-Mirsky theorem [98, 48, 79].

I.4.1 Low-rank approximation

Let $s_l(\mathbf{x}) = u_h(\mathbf{x}, t_{k(l)}; \boldsymbol{\mu}_{j(l)})$ be a snapshot of u_h collected at time instance $t_{k(l)}$ and input parameter $\boldsymbol{\mu}_{j(l)}$. Given a database of K snapshots, the trial subspace is defined as the affine subspace of rank M minimizing, in the least-squares sense, the difference between the snapshots and their orthogonal projections onto this subspace:

$$\begin{cases} \underset{\boldsymbol{\Phi} \in \mathbb{R}^{N \times M}}{\text{minimize}} & \displaystyle\sum_{l=1}^{K} \|s_l(\mathbf{x}) - \widehat{s}_l(\mathbf{x})\|^2_{\boldsymbol{\Theta}} \\ \text{subject to} & \langle \Phi_n(\mathbf{x}), \Phi_m(\mathbf{x}) \rangle_{\boldsymbol{\Theta}} = \delta_{n,m} \quad \forall n, m \in \{1, \ldots, M\}, \end{cases} \tag{I.12}$$

where $\delta_{n,m}$ denotes the Kronecker delta. The orthonormality constrains of this minimization problem (I.12) allow in particular to simplify the projection formulas

(I.7) and (I.11). By introducing the snapshot matrix

$$\mathbf{S} = \begin{pmatrix} \overline{s}_1(\mathbf{x}_1) & \overline{s}_2(\mathbf{x}_1) & \cdots & \overline{s}_K(\mathbf{x}_1) \\ \overline{s}_1(\mathbf{x}_2) & \overline{s}_2(\mathbf{x}_2) & \cdots & \overline{s}_K(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \overline{s}_1(\mathbf{x}_N) & \overline{s}_2(\mathbf{x}_N) & \cdots & \overline{s}_K(\mathbf{x}_N) \end{pmatrix} \in \mathbb{R}^{N \times K},$$

where $\overline{s}_l(\mathbf{x}) = s_l(\mathbf{x}) - u_o(\mathbf{x})$, the minimization problem (I.12) can be cast in matrix format as follows

$$\begin{cases} \underset{\boldsymbol{\Phi} \in \mathbb{R}^{N \times M}}{\text{minimize}} & \left\| \mathbf{S} - \boldsymbol{\Phi}\boldsymbol{\Phi}^T \boldsymbol{\Theta} \mathbf{S} \right\|_{F_{\boldsymbol{\Theta}}}^2 \\ \text{subject to} & \boldsymbol{\Phi}^T \boldsymbol{\Theta} \boldsymbol{\Phi} = \mathbf{I}_M, \end{cases} \tag{I.13}$$

where $\mathbf{I}_M$ denotes the $M \times M$ identity matrix and $\|\mathbf{A}\|_{F_{\boldsymbol{\Theta}}}^2 = \mathrm{Tr}(\mathbf{A}^T \boldsymbol{\Theta} \mathbf{A})$ denotes the Frobenius norm associated with the inner product defined by $\boldsymbol{\Theta}$. Since the matrix $\boldsymbol{\Theta}$ is SPD, the Cholesky decomposition (Definition 1) can be employed to factorize $\boldsymbol{\Theta} = \boldsymbol{\Theta}^{\frac{1}{2}} (\boldsymbol{\Theta}^{\frac{1}{2}})^T$. Moreover, by considering the change of variables $\widetilde{\mathbf{S}} = (\boldsymbol{\Theta}^{\frac{1}{2}})^T \mathbf{S}$ and $\widetilde{\boldsymbol{\Phi}} = (\boldsymbol{\Theta}^{\frac{1}{2}})^T \boldsymbol{\Phi}$ in the minimization problem (I.13), we recover the low-rank approximation problem:

$$\begin{cases} \underset{\widetilde{\boldsymbol{\Phi}} \in \mathbb{R}^{N \times M}}{\text{minimize}} & \left\| \widetilde{\mathbf{S}} - \widetilde{\boldsymbol{\Phi}} \widetilde{\boldsymbol{\Phi}}^T \widetilde{\mathbf{S}} \right\|_F^2 \\ \text{subject to} & \widetilde{\boldsymbol{\Phi}}^T \widetilde{\boldsymbol{\Phi}} = \mathbf{I}_M, \end{cases} \tag{I.14}$$

where $\|\mathbf{A}\|_F^2 = \mathrm{Tr}(\mathbf{A}^T \mathbf{A})$ denotes the Frobenius norm.

Definition 1. ***(Cholesky decomposition)*** *The Cholesky decomposition of a Hermitian positive-definite matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a decomposition of the form*

$$\mathbf{A} = \mathbf{L}\mathbf{L}^T,$$

where $\mathbf{L} \in \mathbb{R}^{n \times n}$ is a lower triangular matrix with real and positive diagonal entries. Every Hermitian positive-definite matrix (and thus also every real-valued symmetric positive-definite matrix) has a unique Cholesky decomposition.

I.4.2 Schmidt-Eckart-Young-Mirsky theorem

In the low-rank approximation problem (I.14), the best approximation to $\widetilde{\mathbf{S}}$ by a matrix $\mathbf{X} = \widetilde{\boldsymbol{\Phi}} \widetilde{\boldsymbol{\Phi}}^T \widetilde{\mathbf{S}}$ of rank M is given by the Schmidt-Eckart-Young-Mirsky theorem 1.

Definition 2. ***(Singular Value Decomposition)*** *Let $\mathbf{M} \in \mathbb{K}^{n \times m}$ where $\mathbb{K}$ is either the field of real numbers or the field of complex numbers. Then, the singular value decomposition of $\mathbf{M}$ exists and is a factorization of the form*

$$\mathbf{M} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T,$$

where $\mathbf{U} \in \mathbb{K}^{n\times n}$ *and* $\mathbf{V} \in \mathbb{K}^{m\times m}$ *are unitary matrices, and* $\mathbf{\Sigma} \in \mathbb{R}_+^{n\times m}$ *is a diagonal matrix with non-negative real numbers on the diagonal. The diagonal entries* σ_n *of* $\mathbf{\Sigma}$ *are known as the singular values of* $\mathbf{M}$*. A common convention is to list the singular values in descending order. In this case,* $\mathbf{\Sigma}$ *is uniquely determined by* $\mathbf{M}$ *(though not the matrices* $\mathbf{U}$ *and* $\mathbf{V}$ *if* $\mathbf{M}$ *is not square).*

Theorem 1. ***(Schmidt-Eckart-Young-Mirsky theorem [98, 48, 79])*** *Let* $\widetilde{\mathbf{S}} \in \mathbb{R}^{n\times m}$ *be a real rectangular matrix. Suppose that the singular value decomposition (Definition 2) of* $\widetilde{\mathbf{S}}$ *is*

$$\widetilde{\mathbf{S}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T,$$

where $\mathbf{U} \in \mathbb{R}^{n\times n}$ *and* $\mathbf{V}^T \in \mathbb{R}^{m\times m}$ *are orthogonal matrices, and* $\mathbf{\Sigma} \in \mathbb{R}_+^{n\times m}$ *is a diagonal matrix with the singular values sorted in descending order. Let* $k \leqslant \min(n, m)$*, the best rank* k *approximation to* $\widetilde{\mathbf{S}}$ *is given by*

$$\min_{\mathrm{rank}(\mathbf{X})\leq k} \left\|\widetilde{\mathbf{S}} - \mathbf{X}\right\|_F^2 = \left\|\widetilde{\mathbf{S}} - \widetilde{\mathbf{S}}^*\right\|_F^2 = \sum_{i=k+1}^{\min(n,m)} \sigma_i^2,$$

where $\widetilde{\mathbf{S}}^*$ *is the trounced singular values decomposition of* $\widetilde{\mathbf{S}}$*:*

$$\widetilde{\mathbf{S}}^* = \begin{pmatrix} U_{1,1} & \cdots & U_{1,k} \\ \vdots & & \vdots \\ U_{n,1} & \cdots & U_{n,k} \end{pmatrix} \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_k \end{pmatrix} \begin{pmatrix} V_{1,1}^T & \cdots & V_{1,m}^T \\ \vdots & & \vdots \\ V_{k,1}^T & \cdots & V_{k,m}^T \end{pmatrix} \in \mathbb{R}^{n\times k}.$$

By considering the change of variables $\mathbf{\Phi} = (\mathbf{\Theta}^{\frac{1}{2}})^{-T}\widetilde{\mathbf{\Phi}}$, the basis functions Φ_n are given by the Schmidt-Eckart-Young-Mirsky theorem 1:

$$\mathbf{\Phi} = (\mathbf{\Theta}^{\frac{1}{2}})^{-T} \begin{pmatrix} U_{1,1} & \cdots & U_{1,M} \\ \vdots & & \vdots \\ U_{N,1} & \cdots & U_{N,M} \end{pmatrix}, \tag{I.15}$$

where $\mathbf{U}$ is obtained from the singular value decomposition (SVD) of $\widetilde{\mathbf{S}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$.

Moreover, according to Theorem 1, the projection error can be evaluated from the singular values corresponding to the neglected basis functions:

$$\left\|\widetilde{\mathbf{S}} - \widetilde{\mathbf{\Phi}}\widetilde{\mathbf{\Phi}}^T\widetilde{\mathbf{S}}\right\|_F = \sqrt{\sum_{n=M+1}^{\min(N,K)} \sigma_n^2}, \tag{I.16}$$

where σ_n are the singular values of $\widetilde{\mathbf{S}}$ sorted in descending order. It follows that the basis functions are ordered in such a way that the first k columns of $\mathbf{\Phi}$ lead to the best rank k approximation to $\mathbf{S}$. The basis functions associated with small singular values can therefore be discarded without significantly changing the accuracy of the projection. This suggests that the number of basis functions M can be chosen so that the projection error is less than a given tolerance.

I.4.3 Trial subspace construction

For the construction of the basis functions, it is not necessary to compute the singular value decomposition of $\widetilde{\mathbf{S}}$ (equation (I.15)), especially for large problem, where the SVD may become computationally prohibitive. In practice, two methods are useful to perform the POD when the number of points N and the number of snapshots K are significantly different from each other:

1. the classical method when $N \ll K$;

2. the method of snapshots [103] when $K \ll N$.

Listing I.1: Matlab style pseudocode to perform the POD.

```
function Phi = POD(S,Theta,tol)

Lt = chol(Theta);
Stilde = Lt*S;
[U,D,~] = svd(Stilde);
ric = cumsum(diag(D).^2)/sum(diag(D).^2);
M = find(ric>1-tol,1);
Phi = Lt\U(:,1:M);

end
```

I.4.3.1 Classical method

In the classical method, we consider the symmetric positive semi-definite correlation matrix

$$\widetilde{\mathbf{S}}\widetilde{\mathbf{S}}^T = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)(\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T.$$

Notably, the matrix $\mathbf{U}$ also corresponds to the left and right eigenvectors of the correlation matrix, and the eigenvalues of $\widetilde{\mathbf{S}}\widetilde{\mathbf{S}}^T$ are equal to the squared singular values of $\widetilde{\mathbf{S}}$. When $N \ll K$, the basis functions can therefore be constructed as follows

$$\mathbf{\Phi} = (\mathbf{\Theta}^{\frac{1}{2}})^{-T} \begin{pmatrix} U_{1,1} & \cdots & U_{1,M} \\ \vdots & & \vdots \\ U_{N,1} & \cdots & U_{N,M} \end{pmatrix},$$

where $\mathbf{U}$ is either obtained from the eigendecomposition or the SVD of $\widetilde{\mathbf{S}}\widetilde{\mathbf{S}}^T \in \mathbb{R}^{N\times N}$. In practice, the SVD of $\widetilde{\mathbf{S}}\widetilde{\mathbf{S}}^T$ is preferable because this decomposition is more accurate for small singular values.

Listing I.2: Matlab style pseudocode to perform the classical method.

```
function Phi = Classical(S,Theta,tol)

Lt = chol(Theta);
Stilde = Lt*S;
[U,D,~] = svd(Stilde*Stilde');
ric = cumsum(diag(D).^2)/sum(diag(D).^2);
M = find(ric>1-tol,1);
Phi = Lt\U(:,1:M);

end
```

I.4.3.2 Method of snapshots

Similarly, the method of snapshots [103] considers the symmetric positive semi-definite correlation matrix

$$\tilde{\mathbf{S}}^T\tilde{\mathbf{S}} = (\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T)^T(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T) = \mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^T.$$

Since $\mathbf{U} = \tilde{\mathbf{S}}\mathbf{V}\boldsymbol{\Sigma}^{-1}$, the basis functions can be constructed when $K \ll N$ as follows

$$\boldsymbol{\Phi} = \mathbf{S}\begin{pmatrix} V_{1,1} & \cdots & V_{1,M} \\ \vdots & & \vdots \\ V_{K,1} & \cdots & V_{K,M} \end{pmatrix}\begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_M \end{pmatrix}^{-1},$$

where $\mathbf{V}$ and $\boldsymbol{\Sigma}$ are obtained from the SVD of $\tilde{\mathbf{S}}^T\tilde{\mathbf{S}} \in \mathbb{R}^{K\times K}$. Note that if $\boldsymbol{\Sigma}$ is singular, the compact SVD of $\tilde{\mathbf{S}}^T\tilde{\mathbf{S}}$, corresponding to non-zero singular values, can be employed since the basis functions associated with zero singular values does not improve the accuracy of the projection.

Listing I.3: Matlab style pseudocode to perform the method of snapshots [103].

```
function Phi = Sirovich(S,Theta,tol)

[V,D,~] = svd(S'*Theta*S);
ric = cumsum(diag(D).^2)/sum(diag(D).^2);
M = find(ric>1-tol,1);
Phi = S*V(:,1:M)/sqrt(D(1:M,1:M));

end
```

I.4.4 Dimensionality reduction analysis

The POD is an effective tool to analyse the reducibility of a problem. According to equation (I.16), the square of the projection error is equal to the sum of the squared singular values corresponding to the neglected basis functions. Based on

this indicator, a criterion to choose the number of basis functions is to find the minimal integer M such that the square of the relative projection error (i.e. the objective function of the low-rank approximation problem (I.14)) is smaller than a given tolerance ϵ. In particular, when the offset is zero (i.e. $u_o(\mathbf{x}) = 0$), this condition reads

$$\frac{\left\|\widetilde{\mathbf{S}} - \widetilde{\mathbf{\Phi}}\widetilde{\mathbf{\Phi}}^T\widetilde{\mathbf{S}}\right\|_F^2}{\left\|\widetilde{\mathbf{S}}\right\|_F^2} = \frac{\sum_{n=M+1}^{\min(N,K)} \sigma_n^2}{\sum_{n=1}^{\min(N,K)} \sigma_n^2} < \epsilon.$$

Equivalently, another popular indicator for choosing the dimension M of the ROM is the Relative Content Information (RIC):

$$RIC(M) = \frac{\sum_{n=1}^{M} \sigma_n^2}{\sum_{n=1}^{\min(N,K)} \sigma_n^2} > 1 - \epsilon,$$

which is often interpreted as the relative energy of the snapshots captured by the basis functions. Notably if the singular values decrease quickly, a small number M of basis functions is suffisant to satisfy small tolerances (e.g. $\epsilon \approx 0.01\%$), and a significant dimensionality reduction can be achieved ($M \ll N$). We illustrate the application of the POD for dimensionality reduction by considering a problem in which a small number of basis functions is sufficient to approximate the solution, and then a problem in which the dimensionality reduction is very limited.

I.4.4.1 Example 1: fast decay of the singular values

First, we consider the one-dimensional heat equation:

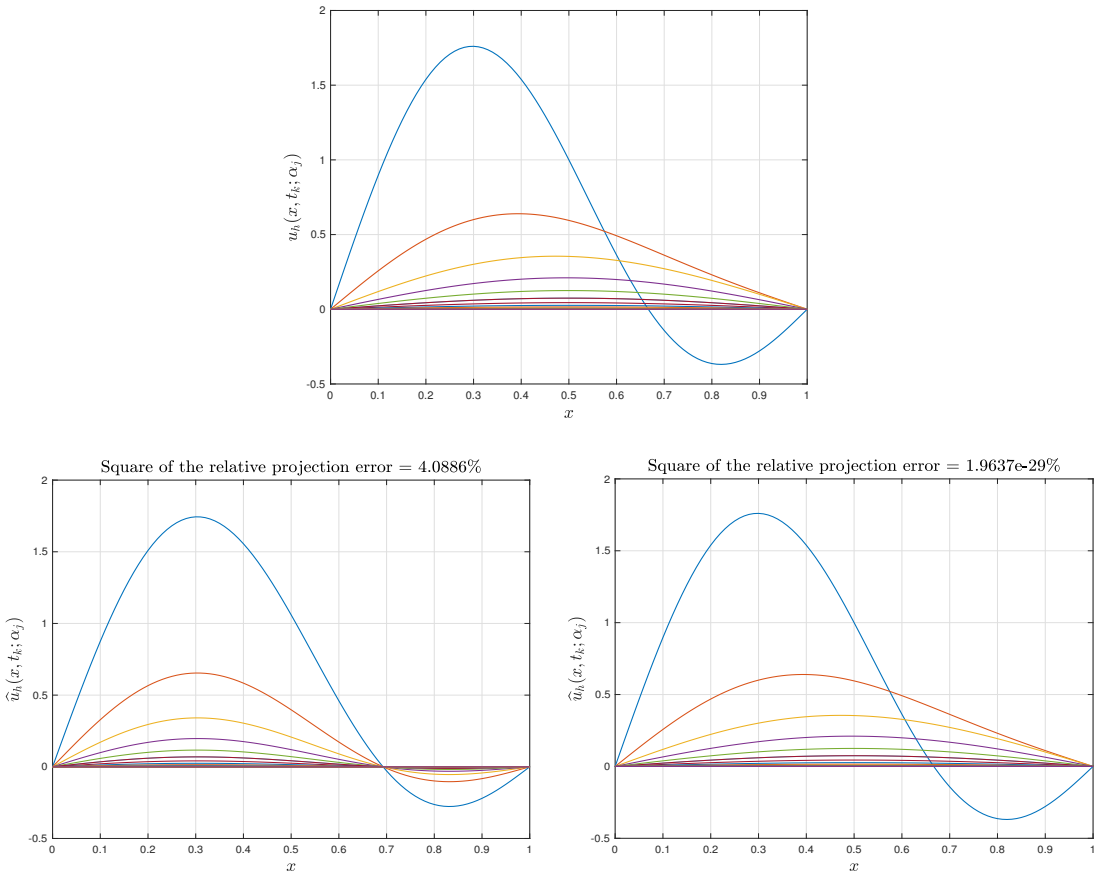
$$\begin{cases} \dfrac{\partial u}{\partial t} - \alpha \dfrac{\partial^2 u}{\partial x^2} = 0 & \text{for } x \in]0,1[, t \in]0,1], \alpha \in [1,5] \\ u(x,0;\alpha) = \sin(\pi x) + \sin(2\pi x) & \text{for } x \in]0,1[, \alpha \in [1,5] \\ u(0,t;\alpha) = u(1,t;\alpha) = 0 & \text{for } t \in]0,1], \alpha \in [1,5] \end{cases}$$

whose exact solution is

$$u(x,t;\alpha) = \sin(\pi x)\exp(-\pi^2 \alpha t) + \sin(2\pi x)\exp(-4\pi^2 \alpha t).$$

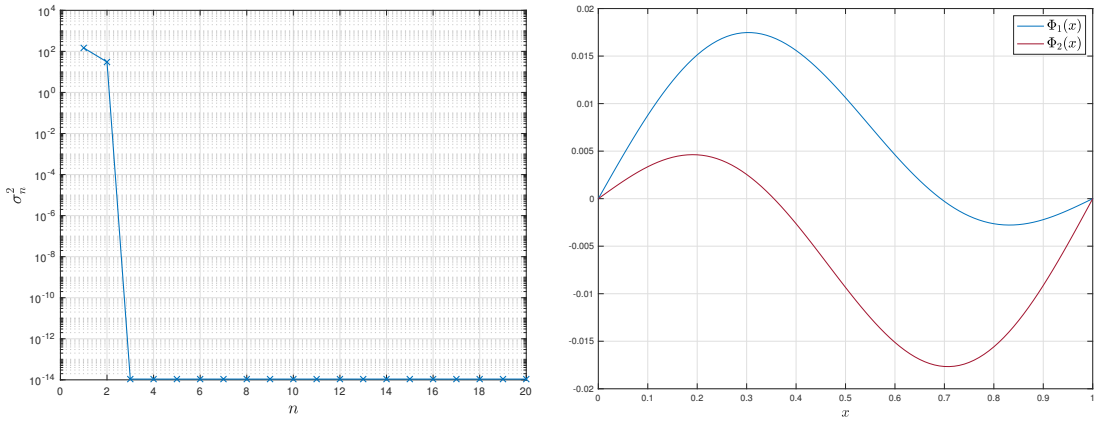
As the solution is a linear combination of two modes $\{\sin(\pi x), \sin(2\pi x)\}$, this one can be exactly represented in the 2-dimensional linear subspace spanned by these two modes.

In Figure I.1, we show snapshots of the solution collected at different time instances t_k and input parameters α_j. In Figure I.2, we plot the squared singular values of the snapshot matrix and the basis functions obtained by POD.



(a) $M = 1$. (b) $M = 2$.

Figure I.1: Top: database containing $K = 40$ snapshots of the solution collected every 0.0526 time units for $\alpha_j \in \{1, 5\}$. Bottom: orthogonal projection of the snapshots onto the M first basis functions.



(a) Squared singular values. (b) Basis functions.

Figure I.2: Results of the POD.

The fast decay of the squared singular values indicates that a significant dimensionality reduction can be achieved. For $n \geqslant 3$, the singular values are almost zero, and the corresponding basis functions does not decrease the projection error. They can therefore be discarded, as expected, since the first two basis functions are sufficient to represent exactly the solution. Note that these two modes are not necessarily equal to $\{\sin(\pi x), \sin(2\pi x)\}$, but they spanned the same subspace.

In Figure I.1, we show the orthogonal projection of the snapshots onto the trial subspace depending the number of basis functions. The projection becomes more accurate when the number of basis functions increases, and with $M = 2$, the projection is exact.

I.4.4.2 Example 2: slow decay of the singular values

Then, we consider the one-dimensional linear transport equation:

$$\begin{cases} \dfrac{\partial u}{\partial t} + c\dfrac{\partial u}{\partial x} = 0 & \text{for } x \in]-5, 25[, t \in]0, 20], c \in [1, 2] \\ u(x, 0; c) = \exp(-x^2) & \text{for } x \in]-5, 25[, c \in [1, 2] \\ u(-5, t; c) = 0 & \text{for } t \in]0, 20], c \in [1, 2] \end{cases}$$

whose exact solution is

$$u(x, t; c) = \exp(-(x - ct)^2).$$

In this case, the exact solution cannot be written as a finite linear combination of modes. To analyse the reducibility of this problem, we first collect $K = 40$ snapshots of the solution taken at different time instances t_k and input parameters c_j as shown in Figure I.3.

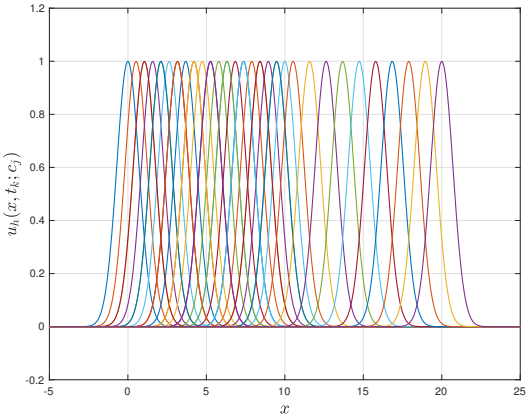
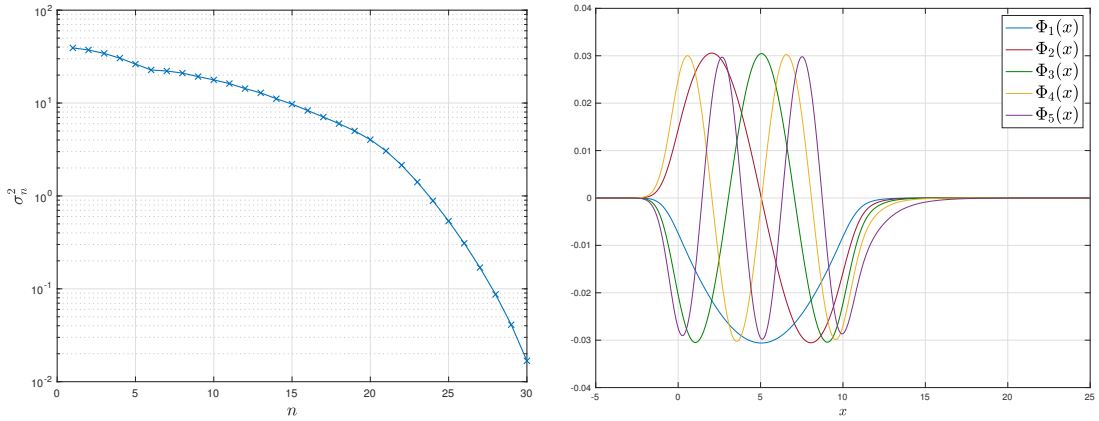


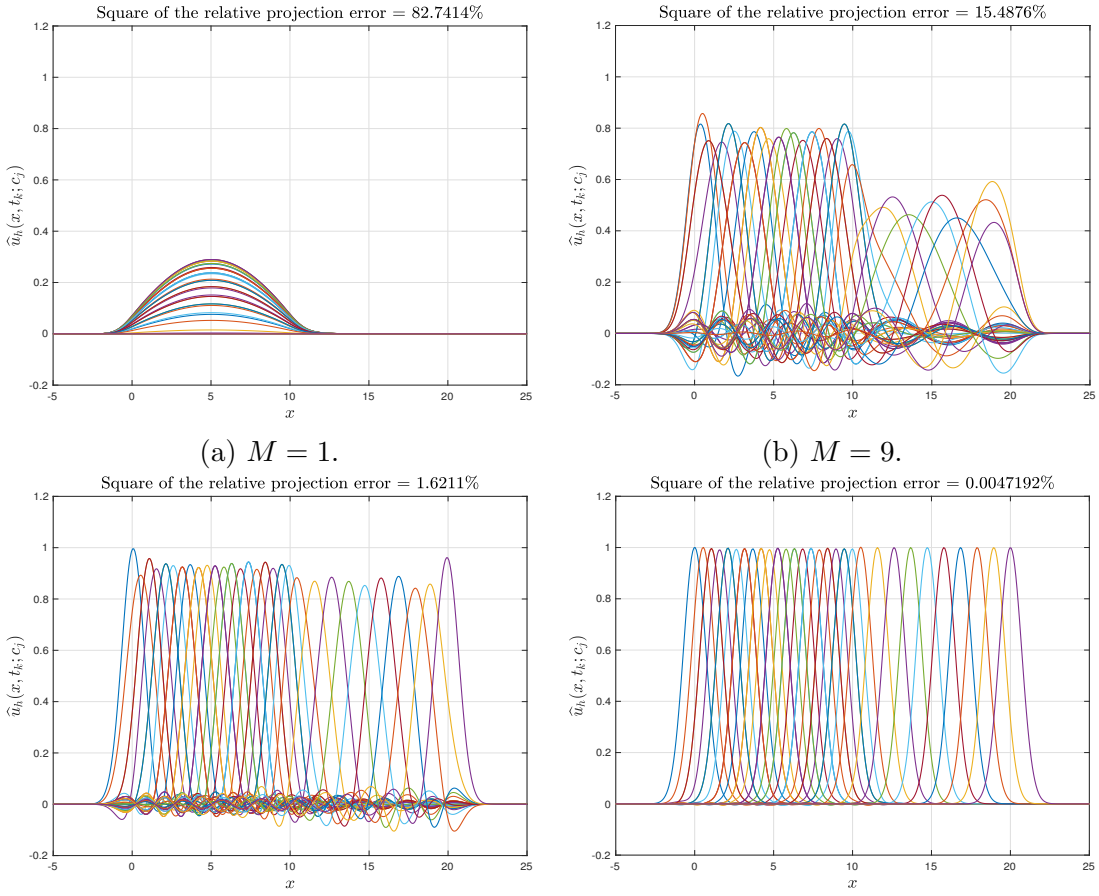
Figure I.3: Database containing $K = 40$ snapshots of the solution collected every 1.0526 time units for $c_j \in \{1, 2\}$.

In Figure I.4, we plot the squared singular values of the snapshot matrix and the basis functions obtained by POD.



(a) Squared singular values. (b) Basis functions.

Figure I.4: Results of the POD.



(a) $M = 1$. (b) $M = 9$.

(c) $M = 16$. (d) $M = 24$.

Figure I.5: Orthogonal projection of the snapshots onto the M first basis functions.

The decay of the squared singular values is very slow, which means that a large number of modes is necessary to accurately approximated the solution. More precisely, at least 24 basis functions are required to obtain a relative squared projection error of less than 0.01% (or equivalently to capture more than 99.99% of the relative energy of the snapshots). Since the rank of the snapshot matrix is at most $\min(N, K) = 40$, the dimensionality reduction is very limited. Figure I.5 shows the orthogonal projection of the snapshots onto the trial subspace depending on the number of basis functions. This example illustrates the limit of the POD to achieve dimensionality reduction for advection-dominated flows.

I.5 Model reduction of nonlinear problems

When the discretization of the spatial differential operator is linear in $\widetilde{\mathbf{u}}_h(t;\boldsymbol{\mu})$ $\big($i.e. $\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu}) = \mathbf{A}\widetilde{\mathbf{u}}_h(t;\boldsymbol{\mu})$ with $\mathbf{A} \in \mathbb{R}^{N\times N}\big)$, the Petrov-Galerkin method leads to a small-scale $M \times M$ system, which can be efficiently solved. For example, consider the semi-discrete system of ODEs (I.7) resulting from the Galerkin projection:

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = \boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu}).$$

Thanks to the linearity of $\mathbf{f}_h[\widetilde{\mathbf{u}}_h]$, this system scales with the dimension M of the ROM:

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = \boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{A}\mathbf{u}_o + \boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{A}\boldsymbol{\Phi}\mathbf{a}(t;\boldsymbol{\mu}),$$

where $\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{A}\mathbf{u}_o \in \mathbb{R}^M$ and $\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{A}\boldsymbol{\Phi} \in \mathbb{R}^{M\times M}$ can be precomputed offline during the training stage. However, in the presence of nonlinear terms, the high-dimensional quantity $\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ must be evaluated and pre-multiplied by $\boldsymbol{\Phi}^T\boldsymbol{\Theta}$. The computational complexity of the resulting ROM scales with the dimension N of the HDM, which is in general computationally prohibitive. To address this computational bottleneck, two methods are commonly used:

1. the precomputation-based approach;

2. the hyper-reduction method [49, 12, 6, 37, 50].

The precomputation-based approach enables the evaluation of $\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ at a cost that scales with the dimension M of the ROM. However, this method can be applied only when $\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ is a polynomial function of $\widetilde{\mathbf{u}}_h(t;\boldsymbol{\mu})$. In the other cases, the ROM is equipped with hyper-reduction techniques, which introduce a second layer of approximation to ensure the computational complexity of the ROM is independent of the dimension N of the HDM.

I.5.1 Precomputation-based approach

When $\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ is polynomial in $\widetilde{\mathbf{u}}_h(t;\boldsymbol{\mu})$, the precomputation-based approach allows to evaluate $\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ without additional approximation at a cost that scales with M. In this approach, $\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ is developed in order to exhibit quantities that can be pre-computed offline in the training stage. The term $\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ can then be evaluated during the prediction stage as a polynomial function of the reduced coordinates $\mathbf{a}(t;\boldsymbol{\mu})$. For example, consider $\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ is a quadratic function of $\widetilde{\mathbf{u}}_h(t;\boldsymbol{\mu})$:

$$\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu}) = \mathbf{b} + \sum_{i=1}^{N} \mathbf{c}_i \widetilde{u}_h(\mathbf{x}_i,t;\boldsymbol{\mu}) + \sum_{i=1}^{N}\sum_{j=1}^{N} \mathbf{d}_{i,j}\widetilde{u}_h(\mathbf{x}_i,t;\boldsymbol{\mu})\widetilde{u}_h(\mathbf{x}_j,t;\boldsymbol{\mu}),$$

where $\mathbf{b},\mathbf{c}_i,\mathbf{d}_{i,j} \in \mathbb{R}^N$. Then, $\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ is given during the prediction stage by

$$\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu}) = \widetilde{\mathbf{b}} + \sum_{n=1}^{M} \widetilde{\mathbf{c}}_n a_n(t;\boldsymbol{\mu}) + \sum_{n=1}^{M}\sum_{m=1}^{M} \widetilde{\mathbf{d}}_{n,m} a_n(t;\boldsymbol{\mu})a_m(t;\boldsymbol{\mu}),$$

where $\widetilde{\mathbf{b}},\widetilde{\mathbf{c}}_n,\widetilde{\mathbf{d}}_{n,m} \in \mathbb{R}^M$ are pre-computed in the training stage as follows

$$\begin{aligned}
\widetilde{\mathbf{b}} &= \boldsymbol{\Phi}^T\boldsymbol{\Theta}\Big(\mathbf{b} + \sum_{i=1}^{N} \mathbf{c}_i u_o(\mathbf{x}_i) + \sum_{i=1}^{N}\sum_{j=1}^{N} \mathbf{d}_{i,j}u_o(\mathbf{x}_i)u_o(\mathbf{x}_j)\Big),\\
\widetilde{\mathbf{c}}_n &= \boldsymbol{\Phi}^T\boldsymbol{\Theta}\Big(\sum_{i=1}^{N} \mathbf{c}_i \Phi_n(\mathbf{x}_i) + \sum_{i=1}^{N}\sum_{j=1}^{N} \mathbf{d}_{i,j}(u_o(\mathbf{x}_i)\Phi_n(\mathbf{x}_j) + u_o(\mathbf{x}_j)\Phi_n(\mathbf{x}_i))\Big),\\
\widetilde{\mathbf{d}}_{n,m} &= \sum_{i=1}^{N}\sum_{j=1}^{N} \boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{d}_{i,j}\Phi_n(\mathbf{x}_i)\Phi_m(\mathbf{x}_j).
\end{aligned}$$

Note that while detailed here for the quadratic case for the sake of clarity, the method is easily generalizable to higher-order polynomials. Given $\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ a polynomial function of degree D, the computational complexity of evaluating $\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ is therefore $O(M^{D+1})$. It follows that if $\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ is a high-order polynomial, then the precomputation-based approach will quickly become computationally prohibitive due to the large number of pre-computed quantities.

I.5.2 Hyper-reduction

When $\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ exhibits non-polynomial nonlinearities or the complexity of the precomputation-based approach is computationally prohibitive, the ROM is equipped with hyper-reduction techniques. Theses methods can be divided into two classes:

1. the approximate-then-project approach [49, 12, 37, 34];
2. the project-then-approximate approach [6, 50, 60, 117].

In theses approaches, $f_h[\widetilde{u}_h](\mathbf{x}, t; \boldsymbol{\mu})$ is evaluated at a few points $\{\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2, \ldots, \widetilde{\mathbf{x}}_L\} \in \Omega$, and a second layer of approximation is introduced in order to ensure the cost of the ROM scales with L instead of N ($L \ll N$).

I.5.2.1 Approximate-then-project approach

In the approximate-then-project methods, $f_h[\widetilde{u}_h](\mathbf{x}, t; \boldsymbol{\mu})$ is approximated by

$$\widetilde{f_h}[\widetilde{u}_h](\mathbf{x}, t; \boldsymbol{\mu}) = \sum_{n=1}^{M_\theta} b_n(t; \boldsymbol{\mu}) \theta_n(\mathbf{x}),$$

where the empirical modes θ_n are built by POD from snapshots of $f_h[s_l](\mathbf{x}, t_{k(l)}; \boldsymbol{\mu}_{j(l)})$ during the training stage and are stored in the matrix

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_1(\mathbf{x}_1) & \theta_2(\mathbf{x}_1) & \cdots & \theta_{M_\theta}(\mathbf{x}_1) \\ \theta_1(\mathbf{x}_2) & \theta_2(\mathbf{x}_2) & \cdots & \theta_{M_\theta}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \theta_1(\mathbf{x}_N) & \theta_2(\mathbf{x}_N) & \cdots & \theta_{M_\theta}(\mathbf{x}_N) \end{pmatrix} \in \mathbb{R}^{N \times M_\theta}.$$

During the prediction stage, the nonlinear function $f_h[\widetilde{u}_h](\mathbf{x}, t; \boldsymbol{\mu})$ is then interpolated at points $\widetilde{\mathbf{x}}_i$ by least-squares regression:

$$\min_{\mathbf{b}(t;\boldsymbol{\mu}) \in \mathbb{R}^{M_\theta}} \sum_{i=1}^{L} \left(f_h[\widetilde{u}_h](\widetilde{\mathbf{x}}_i, t; \boldsymbol{\mu}) - \widetilde{f_h}[\widetilde{u}_h](\widetilde{\mathbf{x}}_i, t; \boldsymbol{\mu}) \right)^2$$

whose solution is given by

$$\mathbf{b}(t; \boldsymbol{\mu}) = (\mathbf{P}\boldsymbol{\theta})^+ \begin{pmatrix} f_h[\widetilde{u}_h](\widetilde{\mathbf{x}}_1, t; \boldsymbol{\mu}) \\ f_h[\widetilde{u}_h](\widetilde{\mathbf{x}}_2, t; \boldsymbol{\mu}) \\ \vdots \\ f_h[\widetilde{u}_h](\widetilde{\mathbf{x}}_L, t; \boldsymbol{\mu}) \end{pmatrix},$$

where $(\mathbf{P}\boldsymbol{\theta})^+$ denotes the Moore-Penrose inverse of $\mathbf{P}\boldsymbol{\theta}$ and $\mathbf{P} \in \mathbb{R}^{L \times N}$ denotes the index matrix

$$P_{i,j} \coloneqq \begin{cases} 1 & \text{if } \widetilde{\mathbf{x}}_i = \mathbf{x}_j \\ 0 & \text{otherwise.} \end{cases} \tag{I.17}$$

By substituting $\widetilde{\mathbf{f}}_h[\widetilde{\mathbf{u}}_h](t; \boldsymbol{\mu})$ to $\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t; \boldsymbol{\mu})$, we finally obtain

$$\boldsymbol{\Phi}^T \boldsymbol{\Theta} \mathbf{f}_h[\widetilde{\mathbf{u}}_h](t; \boldsymbol{\mu}) \approx \boldsymbol{\Phi}^T \boldsymbol{\Theta} \boldsymbol{\theta} (\mathbf{P}\boldsymbol{\theta})^+ \begin{pmatrix} f_h[\widetilde{u}_h](\widetilde{\mathbf{x}}_1, t; \boldsymbol{\mu}) \\ f_h[\widetilde{u}_h](\widetilde{\mathbf{x}}_2, t; \boldsymbol{\mu}) \\ \vdots \\ f_h[\widetilde{u}_h](\widetilde{\mathbf{x}}_L, t; \boldsymbol{\mu}) \end{pmatrix},$$

where $\boldsymbol{\Phi}^T \boldsymbol{\Theta} \boldsymbol{\theta} (\mathbf{P}\boldsymbol{\theta})^+ \in \mathbb{R}^{M \times L}$ is pre-computed offline in the training stage. To select the interpolation points $\widetilde{\mathbf{x}}_i$, many strategies have been proposed in the literature, such as for example, the empirical interpolation method (EIM) [12], the discrete EIM (DEIM) [37] and the Gauss-Newton with approximation tensor (GNAT) [34].

I.5.2.2 Project-then-approximate approach

Instead of approximating and projecting the nonlinear function $f_h[\widetilde{u}_h](\mathbf{x}, t; \boldsymbol{\mu})$, the project-then-approximate methods estimate directly $\langle f_h[\widetilde{u}_h](\mathbf{x}, t; \boldsymbol{\mu}), \Phi_n(\mathbf{x})\rangle_{\boldsymbol{\Theta}}$. In this approach, the inner product is approximated by

$$\langle f_h[\widetilde{u}_h](\mathbf{x}, t; \boldsymbol{\mu}), \Phi_n(\mathbf{x})\rangle_{\boldsymbol{\Theta}} \approx \sum_{i=1}^{L} \widetilde{\omega}_i f_h[\widetilde{u}_h](\widetilde{\mathbf{x}}_i, t; \boldsymbol{\mu})\Phi_n(\widetilde{\mathbf{x}}_i),$$

where $\widetilde{\mathbf{x}}_i$ and $\widetilde{\omega}_l > 0$ denotes the quadrature points and weights, respectively, and we assume here that $\boldsymbol{\Theta}$ is a diagonal matrix for simplicity. The great advantage of this approach is that the quadrature points and weights are computed simultaneously during the training stage in order to best approximate the exact inner product:

$$\underbrace{\begin{pmatrix} F_{1,1}[\widetilde{u}_h](t;\boldsymbol{\mu}) & \cdots & F_{N,1}[\widetilde{u}_h](t;\boldsymbol{\mu}) \\ \vdots & & \vdots \\ F_{1,M}[\widetilde{u}_h](t;\boldsymbol{\mu}) & \cdots & F_{N,M}[\widetilde{u}_h](t;\boldsymbol{\mu}) \end{pmatrix}}_{\mathbf{F}[\widetilde{u}_h](t;\boldsymbol{\mu})} \underbrace{\begin{pmatrix} \omega_1 \\ \vdots \\ \omega_N \end{pmatrix}}_{\boldsymbol{\omega}} \approx \underbrace{\begin{pmatrix} \langle f_h[\widetilde{u}_h](\mathbf{x}, t; \boldsymbol{\mu}), \Phi_1(\mathbf{x})\rangle_{\boldsymbol{\Theta}} \\ \vdots \\ \langle f_h[\widetilde{u}_h](\mathbf{x}, t; \boldsymbol{\mu}), \Phi_M(\mathbf{x})\rangle_{\boldsymbol{\Theta}} \end{pmatrix}}_{\mathbf{c}[\widetilde{u}_h](t;\boldsymbol{\mu})},$$

where $F_{i,n}[\widetilde{u}_h](t;\boldsymbol{\mu}) = f_h[\widetilde{u}_h](\mathbf{x}_i, t; \boldsymbol{\mu})\Phi_n(\mathbf{x}_i)$. As the training is based on the entire snapshot database, we obtain the approximation problem

$$\underbrace{\begin{pmatrix} \mathbf{F}[s_1](t_{k(1)};\boldsymbol{\mu}_{j(1)}) \\ \mathbf{F}[s_2](t_{k(2)};\boldsymbol{\mu}_{j(2)}) \\ \vdots \\ \mathbf{F}[s_K](t_{k(K)};\boldsymbol{\mu}_{j(K)}) \end{pmatrix}}_{\mathbf{G}} \underbrace{\begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \end{pmatrix}}_{\boldsymbol{\omega}} \approx \underbrace{\begin{pmatrix} \mathbf{c}[s_1](t_{k(1)};\boldsymbol{\mu}_{j(1)}) \\ \mathbf{c}[s_2](t_{k(2)};\boldsymbol{\mu}_{j(2)}) \\ \vdots \\ \mathbf{c}[s_K](t_{k(K)};\boldsymbol{\mu}_{j(K)}) \end{pmatrix}}_{\mathbf{d}},$$

where the snapshots s_l can also be replaced by their orthogonal projections $\widehat{s}_l$ in order to further reduce the dimension of the problem and the number of points $\widetilde{\mathbf{x}}_i$ required to achieve accurate approximations. The weights ω_i are then solution of the sparse minimisation problem:

$$\begin{cases} \underset{\boldsymbol{\omega}\in\mathbb{R}_+^N}{\text{minimize}} & \|\boldsymbol{\omega}\|_0 \\ \text{subject to} & \|\mathbf{G}\boldsymbol{\omega} - \mathbf{d}\|_2 \leqslant \epsilon \|\mathbf{d}\|_2 , \end{cases} \tag{I.18}$$

where $\|\cdot\|_0$ denotes the ℓ_0 pseudo-norm. Unfortunately, this problem (I.18) is NP-hard, and in practice, it is replaced by simpler problems such as the non-negative least-squares problem in the energy-conserving sampling and weighting method (ECSW) [50, 51, 55], or the ℓ_1-norm regularization problem in the empirical quadrature procedure (EQP) [117, 116]. The weights $\widetilde{\omega}_i$ are finally obtained

by keeping only the nonzero components of the solution to problem (I.18), and the points $\widetilde{\mathbf{x}}_i$ are the points associated with these weights $\widetilde{\omega}_i$. In the prediction stage, $\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu})$ is therefore approximated by

$$\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{f}_h[\widetilde{\mathbf{u}}_h](t;\boldsymbol{\mu}) \approx (\mathbf{P}\boldsymbol{\Phi})^T \begin{pmatrix} \widetilde{\omega}_1 & & & \\ & \widetilde{\omega}_2 & & \\ & & \ddots & \\ & & & \widetilde{\omega}_L \end{pmatrix} \begin{pmatrix} f_h[\widetilde{u}_h](\widetilde{\mathbf{x}}_1, t; \boldsymbol{\mu}) \\ f_h[\widetilde{u}_h](\widetilde{\mathbf{x}}_2, t; \boldsymbol{\mu}) \\ \vdots \\ f_h[\widetilde{u}_h](\widetilde{\mathbf{x}}_L, t; \boldsymbol{\mu}) \end{pmatrix},$$

where $\mathbf{P}$ denotes the index matrix (I.17) and $(\mathbf{P}\boldsymbol{\Phi})^T\mathrm{diag}(\widetilde{\omega}_1, \widetilde{\omega}_2, \ldots, \widetilde{\omega}_L) \in \mathbb{R}^{M\times L}$ is precomputed offline during the training stage.

Chapter II

A reduced-order model for rarified flows

II.1 Introduction

In fluid dynamics, the regime of a gas flow is characterized by the Knudsen number

$$Kn \coloneqq \frac{\lambda}{L},$$

defined as the ratio between the mean free path of the particles λ and the characteristic length of the problem L. When the Knudsen number is low ($Kn \ll 1$), the gas particles are close to each other with respect to the characteristic length of the problem. The behaviour of the particles is similar to the macroscopic flow, and the regime is said hydrodynamic. Conversely in the rarefied regime ($Kn \gtrsim 1$), the behaviour of each particle can significantly differ from the macroscopic flow due to the large distance between the particles.

For the simulation of hydrodynamic flows, it is generally sufficient to consider the macroscopic flow as in the Euler or Navier-Stokes equations. However in the rarified regime, this approach can fail to properly describe the dynamic of the fluid. In this work, we consider the Boltzmann equation [35]:

$$\frac{\partial f}{\partial t}(\mathbf{x}, \boldsymbol{\xi}, t) + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\xi}, t) = Q(f, f), \tag{II.1}$$

which is valid to model gas flows in both hydrodynamic and rarefied regimes. This equation describes the microscopic behaviour of the gas particles, instead of considering only the macroscopic state such as the density, velocity and pressure of the gas. The non-negative function f represents the temporal evolution of the distribution of the gas particles at point $\mathbf{x}$ and moving with microscopic velocity $\boldsymbol{\xi}$. Two approaches are mainly used to solve the Boltzmann equation:

1. the probabilistic approach, such as the direct simulation Monte Carlo method;

2. the deterministic approach, relying on the discretization of the Boltzmann equation (II.1).

The deterministic approach is computationally very expensive due to the quadratic cost of the velocity discretization of the collision operator Q. As a consequence, the probabilistic approach is extensively used in engineering applications due to its lower computational cost. However, this approach leads to noisy results compared to the deterministic approach.

In this work, we consider a deterministic model developed during the PhD thesis of F. Bernard [21] to simulate gas flows in both hydrodynamic and rarefied regimes. In this model, the Boltzmann equation is replaced by simplified models such as the BGK equation [23], which is known to be sufficient for moderate and small Knudsen numbers ($Kn < 1$). This equation is then discretized in velocity space by a discrete velocity method [31, 78], leading to a system of transport equations. This system is finally solved by the finite volume method [110] in space and an implicit-explicit Runge-Kutta scheme [10, 67, 86] in time. The resulting model is referred to as the high-dimensional model (HDM) in the following and allows to efficiently simulate gas flows in both hydrodynamic and rarefied regimes. However, the large number of dimensions (i.e. 3 in space + 3 in velocity + 1 in time) still leads to a computationally expensive model, whose simulations require weeks on supercomputers.

For this reason, we develop in this thesis [22] a stable, accurate and efficient reduced-order model (ROM) to compute approximations of the density distribution function f at low cost with respect to the HDM. This ROM employs a new reduced-order approximation of the BGK equation where the gas density distribution function is represented in velocity space by a small number of basis functions:

$$\widetilde{f}(\mathbf{x}, \boldsymbol{\xi}, t) = \sum_{n=1}^{N_{pod}} a_n(\mathbf{x}, t)\Phi_n(\boldsymbol{\xi}).$$

The construction of the ROM adopts an approach based on Proper Orthogonal Decomposition [87, 48, 103, 20] in the training stage and on the Galerkin method in the prediction stage. This approach is then adapted to the case of the BGK equation, and the ROM is modified in order to preserve important properties of the HDM. Furthermore, we derive the CFL condition of the numerical schemes to ensure a stable ROM in 1D. We investigate the reproduction and prediction of unsteady flows in both hydrodynamic and rarefied regimes. The results demonstrate the accuracy of the ROM and the significant reduction of the computational cost with respect to the HDM.

This work is organized as follows. In Section II.2, we briefly introduce the HDM modeling gas flows in both hydrodynamic and rarefied regimes. Then, Section II.3 presents in detail the training and prediction stages of the ROM approximating the HDM. Finally, the last Section II.4 demonstrates the performance of the ROM with respect to the HDM.

II.2 High-dimensional model

The high-dimensional model (HDM) was developed during the PhD thesis of F. Bernard [21] to simulate gas flows in both hydrodynamic and rarefied regimes. The dynamic of the gas flow is described by the Bathnagar-Gross-Krook (BGK) equation [23], which is an approximation of the Boltzmann equation, known to be sufficient for moderate and small Knudsen numbers ($Kn < 1$). This equation is then discretized in velocity space by a discrete velocity method [31, 78] to ensure the conservation of mass, momentum and energy of the gas at the discrete level. This discretization step leads to a large-scale system of transport equations, which is solved by a finite volume scheme [110] in space and an implicit-explicit Runge-Kutta scheme [10, 67, 86] in time.

II.2.1 BGK model

The dynamic of the gas flow is described by the BGK equation, wherein the collision term Q is approximated by a relaxation of the density distribution function towards the Maxwellian distribution function. Moreover, the Chu reduction [40] is used in 1D and 2D to reduce the number of dimension in velocity space and thus speed up the computations. For simplicity, we consider a monoatomic gas, and the specific gas constant R is taken as $R = 1$ in the following.

II.2.1.1 BGK equation

Let the parameter domain $\mathcal{D} \subset \mathbb{R}^p$ be a closed and bounded subset of the Euclidean space $\mathbb{R}^p$ with $p \in \mathbb{N}^*$. Moreover, let $\Omega_{\mathbf{x}} \subset \mathbb{R}^d$ be a regular open physical domain with boundary $\partial\Omega_{\mathbf{x}}$, where $d \in \{1, 2, 3\}$ is the space dimension. In the HDM, the dynamic of the gas flow is governed by the parametrized BGK equation for $\mathbf{x} \in \Omega_{\mathbf{x}}$, $\boldsymbol{\xi} = (\xi_u, \xi_v, \xi_w)^T \in \mathbb{R}^3$, $t \in \mathbb{R}_+^*$ and $\boldsymbol{\mu} \in \mathcal{D}$:

$$\frac{\partial f}{\partial t}(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) + \boldsymbol{\xi} \cdot \nabla_{\mathbf{x}} f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \frac{M_f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) - f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu})}{\tau(\mathbf{x}, t; \boldsymbol{\mu})}. \tag{II.2}$$

For each input parameter $\boldsymbol{\mu}$, the density distribution function $f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu})$ represents the temporal evolution of the distribution of the gas particles at point $\mathbf{x}$ and moving with microscopic velocity $\boldsymbol{\xi}$. The relaxation time τ is given in dimensionless form by

$$\tau^{-1}(\mathbf{x}, t; \boldsymbol{\mu}) = \frac{\rho(\mathbf{x}, t; \boldsymbol{\mu}) T^{1-\nu}(\mathbf{x}, t; \boldsymbol{\mu})}{Kn}$$

with ν, the exponent of the viscosity law of the gas, taken as $\nu = 1$ in the following. In the BGK equation, the collision term is linearized around the Maxwellian equilibrium distribution function

$$M_f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \frac{\rho(\mathbf{x}, t; \boldsymbol{\mu})}{(2\pi T(\mathbf{x}, t; \boldsymbol{\mu}))^{\frac{3}{2}}} \exp\left(-\frac{\|\boldsymbol{\xi} - \mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu})\|_2^2}{2T(\mathbf{x}, t; \boldsymbol{\mu})}\right),$$

where ρ is the density, $\mathbf{u} \in \mathbb{R}^d$ is the macroscopic velocity and T is the temperature of the gas. These macroscopic quantities of interest are recovered from the density distribution function. The density, momentum and energy E are given by

$$\int_{\mathbb{R}^3} f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) \begin{pmatrix} 1 \\ \boldsymbol{\xi} \\ \frac{\|\boldsymbol{\xi}\|_2^2}{2} \end{pmatrix} \mathrm{d}\boldsymbol{\xi} = \begin{pmatrix} \rho(\mathbf{x}, t; \boldsymbol{\mu}) \\ \rho(\mathbf{x}, t; \boldsymbol{\mu}) \mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu}) \\ E(\mathbf{x}, t; \boldsymbol{\mu}) \end{pmatrix}, \tag{II.3}$$

and the temperature T and pressure p of the gas are then deduced from

$$T(\mathbf{x}, t; \boldsymbol{\mu}) = \frac{2E(\mathbf{x}, t; \boldsymbol{\mu})}{3\rho(\mathbf{x}, t; \boldsymbol{\mu})} - \frac{\|\mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu})\|_2^2}{3} \quad \text{and} \quad p(\mathbf{x}, t; \boldsymbol{\mu}) = \rho(\mathbf{x}, t; \boldsymbol{\mu}) T(\mathbf{x}, t; \boldsymbol{\mu}).$$

This equation (II.3) connects the microscopic behaviour of the particles with the macroscopic state of the gas. It is verified by every distribution function (i.e. f and M_f) and ensures the conservation of mass, momentum and energy. In the hydrodynamic limit ($Kn \to 0$), the density distribution function tends to the Maxwellian distribution function ($f \to M_f$), and the compressible Euler equations can be derived from the BGK equation by the Chapman-Enskog expansion [36].

II.2.1.2 Chu reduction

To ensure equation (II.3), the velocity space has always 3 dimensions even if the physical space has less dimensions. In 1D and 2D, the Chu reduction allows to speed up computations by reducing the number of dimension in velocity space. The density distribution functions f, defined on $\mathbb{R}^3$ in velocity space, is replaced by two density distribution functions ϕ and ψ, defined on $\mathbb{R}^d$ in velocity space. The macroscopic quantities of interest are then deduced from these new density distribution functions.

1D case. We consider the one-dimensional BGK equation ($d = 1$):

$$\frac{\partial f}{\partial t}(x, \boldsymbol{\xi}, t; \boldsymbol{\mu}) + \xi_u \frac{\partial f}{\partial x}(x, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \frac{M_f(x, \boldsymbol{\xi}, t; \boldsymbol{\mu}) - f(x, \boldsymbol{\xi}, t; \boldsymbol{\mu})}{\tau(x, t; \boldsymbol{\mu})}. \tag{II.4}$$

In 1D, the density distribution function $f(x, \boldsymbol{\xi}, t; \boldsymbol{\mu})$ is replaced by

$$\begin{aligned} \phi(x, \xi_u, t; \boldsymbol{\mu}) &= \int_{\mathbb{R}^2} f(x, \boldsymbol{\xi}, t; \boldsymbol{\mu}) \, \mathrm{d}\xi_v \, \mathrm{d}\xi_w, \\ \psi(x, \xi_u, t; \boldsymbol{\mu}) &= \int_{\mathbb{R}^2} \frac{\xi_v^2 + \xi_w^2}{2} f(x, \boldsymbol{\xi}, t; \boldsymbol{\mu}) \, \mathrm{d}\xi_v \, \mathrm{d}\xi_w. \end{aligned} \tag{II.5}$$

By integrating equation (II.4) in velocity space, the new density distribution functions verify

$$\begin{cases} \dfrac{\partial \phi}{\partial t}(x, \xi_u, t; \boldsymbol{\mu}) + \xi_u \dfrac{\partial \phi}{\partial x}(x, \xi_u, t; \boldsymbol{\mu}) = \dfrac{M_\phi(x, \xi_u, t; \boldsymbol{\mu}) - \phi(x, \xi_u, t; \boldsymbol{\mu})}{\tau(x, t; \boldsymbol{\mu})} \\ \dfrac{\partial \psi}{\partial t}(x, \xi_u, t; \boldsymbol{\mu}) + \xi_u \dfrac{\partial \psi}{\partial x}(x, \xi_u, t; \boldsymbol{\mu}) = \dfrac{M_\psi(x, \xi_u, t; \boldsymbol{\mu}) - \psi(x, \xi_u, t; \boldsymbol{\mu})}{\tau(x, t; \boldsymbol{\mu})}, \end{cases}$$

where the new equilibrium distribution functions are defined by

$$M_\phi(x,\xi_u,t;\boldsymbol{\mu}) = \frac{\rho(x,t;\boldsymbol{\mu})}{\sqrt{2\pi T(x,t;\boldsymbol{\mu})}}\exp\left(-\frac{(\xi_u-u(x,t;\boldsymbol{\mu}))^2}{2T(x,t;\boldsymbol{\mu})}\right),$$
$$M_\psi(x,\xi_u,t;\boldsymbol{\mu}) = T(x,t;\boldsymbol{\mu})M_\phi(x,\xi_u,t;\boldsymbol{\mu}).$$

By inserting (II.5) into (II.3), the macroscopic quantities are then deduced from

$$\int_{\mathbb{R}}\phi(x,\xi_u,t;\boldsymbol{\mu})\begin{pmatrix}1\\ \xi_u\\ \frac{\xi_u^2}{2}\end{pmatrix}\mathrm{d}\xi_u + \int_{\mathbb{R}}\psi(x,\xi_u,t;\boldsymbol{\mu})\begin{pmatrix}0\\0\\1\end{pmatrix}\mathrm{d}\xi_u = \begin{pmatrix}\rho(x,t;\boldsymbol{\mu})\\ \rho(x,t;\boldsymbol{\mu})u(x,t;\boldsymbol{\mu})\\ E(x,t;\boldsymbol{\mu})\end{pmatrix}. \tag{II.6}$$

2D case. Similarly, we consider the two-dimensional BGK equation ($d=2$):

$$\frac{\partial f}{\partial t}(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}) + \boldsymbol{\xi}_2\cdot\nabla_{\mathbf{x}}f(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}) = \frac{M_f(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu})-f(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu})}{\tau(\mathbf{x},t;\boldsymbol{\mu})},$$

where $\mathbf{x}=(x,y)^T$ and $\boldsymbol{\xi}_2=(\xi_u,\xi_v)^T$. The new density distribution functions are defined by

$$\phi(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu}) = \int_{\mathbb{R}} f(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu})\,\mathrm{d}\xi_w \quad\text{and}\quad \psi(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu}) = \int_{\mathbb{R}}\frac{\xi_w^2}{2}f(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu})\,\mathrm{d}\xi_w.$$

These new density distribution functions verify

$$\begin{cases}\dfrac{\partial\phi}{\partial t}(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu}) + \boldsymbol{\xi}_2\cdot\nabla_{\mathbf{x}}\phi(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu}) = \dfrac{M_\phi(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu})-\phi(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu})}{\tau(\mathbf{x},t;\boldsymbol{\mu})}\\ \dfrac{\partial\psi}{\partial t}(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu}) + \boldsymbol{\xi}_2\cdot\nabla_{\mathbf{x}}\psi(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu}) = \dfrac{M_\psi(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu})-\psi(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu})}{\tau(\mathbf{x},t;\boldsymbol{\mu})},\end{cases}$$

where the new equilibrium distribution functions are

$$M_\phi(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu}) = \frac{\rho(\mathbf{x},t;\boldsymbol{\mu})}{2\pi T(\mathbf{x},t;\boldsymbol{\mu})}\exp\left(-\frac{\|\boldsymbol{\xi}_2-\mathbf{u}(\mathbf{x},t;\boldsymbol{\mu})\|_2^2}{2T(\mathbf{x},t;\boldsymbol{\mu})}\right),$$
$$M_\psi(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu}) = \frac{T(\mathbf{x},t;\boldsymbol{\mu})}{2}M_\phi(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu}),$$

with $\mathbf{u}=(u,v)^T$. The macroscopic state of the gas is finally recovered from

$$\int_{\mathbb{R}^2}\phi(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu})\begin{pmatrix}1\\ \xi_u\\ \xi_v\\ \frac{\|\boldsymbol{\xi}_2\|_2^2}{2}\end{pmatrix}\mathrm{d}\boldsymbol{\xi}_2 + \int_{\mathbb{R}^2}\psi(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu})\begin{pmatrix}0\\0\\0\\1\end{pmatrix}\mathrm{d}\boldsymbol{\xi}_2 = \begin{pmatrix}\rho(\mathbf{x},t;\boldsymbol{\mu})\\ \rho(\mathbf{x},t;\boldsymbol{\mu})u(\mathbf{x},t;\boldsymbol{\mu})\\ \rho(\mathbf{x},t;\boldsymbol{\mu})v(\mathbf{x},t;\boldsymbol{\mu})\\ E(\mathbf{x},t;\boldsymbol{\mu})\end{pmatrix}. \tag{II.7}$$

II.2.2 Numerical methods

The BGK model is discretized in velocity space by a discrete velocity method (DVM) [31, 78] to ensure the conservation of mass, momentum and energy of the gas at the discrete level. This discretization step leads to a large-scale system of transport equations solved by the finite volume method [110] in space and an implicit-explicit (IMEX) Runge-Kutta scheme [10, 67, 86] in time. The resulting HDM is a first-order scheme, and the discrete density distribution function $f_h(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu})$ represents the exact density distribution function f at point $\mathbf{x}$, microscopic velocity $\boldsymbol{\xi}$, time instance t and input parameter $\boldsymbol{\mu}$.

II.2.2.1 Velocity space discretization

In equation (II.3), the density distribution function is integrated in velocity space over $\mathbb{R}^3$. For this reason, the velocity space $\Omega_{\boldsymbol{\xi}}$ is chosen so that f is negligible outside $\Omega_{\boldsymbol{\xi}}$. More precisely, the lengths L_{ξ_u}, L_{ξ_u} and L_{ξ_w} of the velocity domain $\Omega_{\boldsymbol{\xi}} =]-L_{\xi_u}, L_{\xi_u}[\times]-L_{\xi_u}, L_{\xi_u}[\times]-L_{\xi_w}, L_{\xi_w}[$ are defined to capture at least 99.99% of the integral of the distribution functions. The velocity space is then discretized by a uniform cartesian grid containing $N_{\boldsymbol{\xi}}$ points $\boldsymbol{\xi}_{i,j,k} = (\xi_u^i, \xi_v^j, \xi_w^k)$, where $\xi_u^i = -L_{\xi_u} + (i - \frac{1}{2})\Delta\xi_u$ and $\Delta\xi_u = \frac{2L_{\xi_u}}{N_{\xi_u}}$. In the following, the points are indexed by a multi-index $\boldsymbol{\xi}_l = \boldsymbol{\xi}_{i(l),j(l),k(l)}$ to simplify notation.

Quadrature rule. The integrals are approximated by the midpoint rule:

$$\int_{\Omega_{\boldsymbol{\xi}}} g(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi} \approx \Delta\boldsymbol{\xi} \sum_{l=1}^{N_{\boldsymbol{\xi}}} g(\boldsymbol{\xi}_l),$$

where $\Delta\boldsymbol{\xi} = \Delta\xi_u \Delta\xi_u \Delta\xi_w$. The discrete inner product associated with the L^2-norm is therefore defined by

$$\langle g_1(\boldsymbol{\xi}), g_2(\boldsymbol{\xi}) \rangle_{\boldsymbol{\Theta}} = \Delta\boldsymbol{\xi} \sum_{l=1}^{N_{\boldsymbol{\xi}}} g_1(\boldsymbol{\xi}_l) g_2(\boldsymbol{\xi}_l).$$

This discrete inner product is induced by the diagonal matrix

$$\boldsymbol{\Theta} = \begin{pmatrix} \Delta\boldsymbol{\xi} & & & \\ & \Delta\boldsymbol{\xi} & & \\ & & \ddots & \\ & & & \Delta\boldsymbol{\xi} \end{pmatrix} \in \mathbb{R}^{N_{\boldsymbol{\xi}} \times N_{\boldsymbol{\xi}}}$$

and corresponds in matrix form to

$$\langle g_1(\boldsymbol{\xi}), g_2(\boldsymbol{\xi}) \rangle_{\boldsymbol{\Theta}} = \mathbf{g}_1^T \boldsymbol{\Theta} \mathbf{g}_2,$$

where the scalar function $g(\boldsymbol{\xi})$ is encoded as the vector $\mathbf{g} = (g(\boldsymbol{\xi}_1), g(\boldsymbol{\xi}_2), \ldots, g(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}))^T \in \mathbb{R}^{N_{\boldsymbol{\xi}}}$.

Discrete velocity method. If the discrete Maxwellian distribution function M_{f_h} is defined by

$$M_{f_h}(\mathbf{x}, \boldsymbol{\xi}_l, t; \boldsymbol{\mu}) = M_f(\mathbf{x}, \boldsymbol{\xi}_l, t; \boldsymbol{\mu}), \tag{II.8}$$

then M_{f_h} will not necessarily verify equation (II.3) at the discrete level, and the conservation of mass, momentum and energy will not hold:

$$\left\langle M_{f_h}(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}), \begin{pmatrix} 1 \\ \boldsymbol{\xi} \\ \frac{\|\boldsymbol{\xi}\|_2^2}{2} \end{pmatrix} \right\rangle_{\boldsymbol{\Theta}} \neq \begin{pmatrix} \rho(\mathbf{x}, t; \boldsymbol{\mu}) \\ \rho(\mathbf{x}, t; \boldsymbol{\mu}) \mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu}) \\ E(\mathbf{x}, t; \boldsymbol{\mu}) \end{pmatrix}.$$

For this reason, the discrete Maxwellian distribution function is not computed from equation (II.8). In the DVM [31, 78], the discrete Maxwellian distribution function is defined by

$$M_{f_h}(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \exp(\boldsymbol{\omega}(\mathbf{x}, t; \boldsymbol{\mu}) \cdot \mathbf{m}(\boldsymbol{\xi})),$$

where the vector $\boldsymbol{\omega}(\mathbf{x}, t; \boldsymbol{\mu}) \in \mathbb{R}^5$ is computed in order to verify

$$\left\langle M_{f_h}(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}), \begin{pmatrix} 1 \\ \boldsymbol{\xi} \\ \frac{\|\boldsymbol{\xi}\|_2^2}{2} \end{pmatrix} \right\rangle_{\boldsymbol{\Theta}} = \begin{pmatrix} \rho(\mathbf{x}, t; \boldsymbol{\mu}) \\ \rho(\mathbf{x}, t; \boldsymbol{\mu}) \mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu}) \\ E(\mathbf{x}, t; \boldsymbol{\mu}) \end{pmatrix} \tag{II.9}$$

with $\mathbf{m}(\boldsymbol{\xi}) = (1, \boldsymbol{\xi}, \frac{\|\boldsymbol{\xi}\|_2^2}{2})^T \in \mathbb{R}^5$. This nonlinear system (II.9) is solved by the Newton-Raphson method at each point $\mathbf{x}$, time instance t and input parameter $\boldsymbol{\mu}$, as explained in [21]. In the same way, the discrete equilibrium distribution functions M_{ϕ_h} and M_{ψ_h} are computed to verify, at the discrete level, equation (II.6) (resp. (II.7)) in 1D (resp. 2D), see [21]. The BGK equation (II.2) becomes after velocity space discretization

$$\frac{\partial f_h}{\partial t}(\mathbf{x}, \boldsymbol{\xi}_l, t; \boldsymbol{\mu}) + \boldsymbol{\xi}_l \cdot \nabla_{\mathbf{x}} f_h(\mathbf{x}, \boldsymbol{\xi}_l, t; \boldsymbol{\mu}) = \frac{M_{f_h}(\mathbf{x}, \boldsymbol{\xi}_l, t; \boldsymbol{\mu}) - f_h(\mathbf{x}, \boldsymbol{\xi}_l, t; \boldsymbol{\mu})}{\tau(\mathbf{x}, t; \boldsymbol{\mu})} \tag{II.10}$$

for $l \in \{1, \ldots, N_{\boldsymbol{\xi}}\}$. Notably, this system (II.10) consists of $N_{\boldsymbol{\xi}}$ transport equations with a collision term coupling all the equations.

II.2.2.2 Physical space discretization

The physical domain $\Omega_{\mathbf{x}} =]x_{min}, x_{max}[\times]y_{min}, y_{max}[\times]z_{min}, z_{max}[$ is discretized by a uniform cartesian mesh containing $N_{\mathbf{x}}$ cells $K_{i,j,k}$ with center $\mathbf{x}_{i,j,k} = (x_i, y_j, z_k)$ and size $\Delta x \Delta y \Delta z$, where $x_i = x_{min} + (i - \frac{1}{2})\Delta x$ and $\Delta x = \frac{x_{max} - x_{min}}{N_x}$. On each cell, the convective term is approximated by the finite volume method, while the collision term is discretized by a centered approximation. On cartesian grid, the first-order finite volume scheme reads

$$\xi_{u_l} \frac{\partial f_h}{\partial x}(\mathbf{x}_{i,j,k}, \boldsymbol{\xi}_l, t; \boldsymbol{\mu}) = \frac{F^l_{i+\frac{1}{2},j,k} - F^l_{i-\frac{1}{2},j,k}}{\Delta x},$$

where the flux $F^l_{i+\frac{1}{2},j,k}$ between the cells $K_{i,j,k}$ and $K_{i+1,j,k}$ is

$$F^l_{i+\frac{1}{2},j,k} = \max(\xi_{u_l},0) f_h(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu}) + \min(\xi_{u_l},0) f_h(\mathbf{x}_{i+1,j,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu}).$$

After physical space discretization, the system (II.10) becomes

$$\begin{aligned}
\frac{\partial f_h}{\partial t}(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu}) = &- \max(\xi_{u_l},0)\frac{f_h(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu}) - f_h(\mathbf{x}_{i-1,j,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu})}{\Delta x} \\
&- \min(\xi_{u_l},0)\frac{f_h(\mathbf{x}_{i+1,j,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu}) - f_h(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu})}{\Delta x} \\
&- \max(\xi_{v_l},0)\frac{f_h(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu}) - f_h(\mathbf{x}_{i,j-1,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu})}{\Delta y} \\
&- \min(\xi_{v_l},0)\frac{f_h(\mathbf{x}_{i,j+1,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu}) - f_h(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu})}{\Delta y} \\
&- \max(\xi_{w_l},0)\frac{f_h(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu}) - f_h(\mathbf{x}_{i,j,k-1},\boldsymbol{\xi}_l,t;\boldsymbol{\mu})}{\Delta z} \\
&- \min(\xi_{w_l},0)\frac{f_h(\mathbf{x}_{i,j,k+1},\boldsymbol{\xi}_l,t;\boldsymbol{\mu}) - f_h(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu})}{\Delta z} \\
&+ \frac{M_{f_h}(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu}) - f_h(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l,t;\boldsymbol{\mu})}{\tau(\mathbf{x}_{i,j,k},t;\boldsymbol{\mu})},
\end{aligned} \tag{II.11}$$

which corresponds to a system of $N_{\mathbf{x}}N_{\boldsymbol{\xi}}$ ODEs. The boundary conditions completing this system are presented in Section II.2.2.4.

II.2.2.3 Time discretization

The system (II.11) is solved by an IMEX Runge-Kutta scheme [10, 67, 86]. In this method, the convective term is treated explicitly, while the collision term is treated implicitly. In this way, the CFL condition does not depend on the collision term, which tends to zero in the hydrodynamic limit ($Kn \to 0$). In the first-order IMEX Runge-Kutta scheme, there is one intermediate time-step given by the implicit formula

$$f_h^{(1)}(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l;\boldsymbol{\mu}) = f_h(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l,t_p;\boldsymbol{\mu}) + \Delta t \frac{M_{f_h}^{(1)}(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l;\boldsymbol{\mu}) - f_h^{(1)}(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l;\boldsymbol{\mu})}{\tau^{(1)}(\mathbf{x}_{i,j,k};\boldsymbol{\mu})}.$$

By integrating this formula in velocity space, it follows that f_h and $f_h^{(1)}$ have the same moments because $f_h^{(1)}$ and $M_{f_h}^{(1)}$ have the same moments. In addition, since f_h and M_{f_h} also have the same moments, this implies that $M_{f_h}^{(1)}$ (resp. $\tau^{(1)}$) is equal to M_{f_h} (resp. τ). The intermediate time-step can therefore be re-written in explicit format as follows

$$f_h^{(1)}(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l;\boldsymbol{\mu}) = \frac{\tau(\mathbf{x}_{i,j,k},t_p;\boldsymbol{\mu})}{\Delta t + \tau(\mathbf{x}_{i,j,k},t_p;\boldsymbol{\mu})}\left(f_h(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l,t_p;\boldsymbol{\mu}) + \Delta t \frac{M_{f_h}(\mathbf{x}_{i,j,k},\boldsymbol{\xi}_l,t_p;\boldsymbol{\mu})}{\tau(\mathbf{x}_{i,j,k},t_p;\boldsymbol{\mu})}\right). \tag{II.12}$$

The next time-step is then given by

$$\begin{aligned} f_h(\mathbf{x}_{i,j,k}, \boldsymbol{\xi}_l, t_{p+1}; \boldsymbol{\mu}) = f_h(\mathbf{x}_{i,j,k}, \boldsymbol{\xi}_l, t_p; \boldsymbol{\mu}) \\ - \Delta t \max(\xi_{u_l}, 0) \frac{f_h^{(1)}(\mathbf{x}_{i,j,k}, \boldsymbol{\xi}_l; \boldsymbol{\mu}) - f_h^{(1)}(\mathbf{x}_{i-1,j,k}, \boldsymbol{\xi}_l; \boldsymbol{\mu})}{\Delta x} \\ - \Delta t \min(\xi_{u_l}, 0) \frac{f_h^{(1)}(\mathbf{x}_{i+1,j,k}, \boldsymbol{\xi}_l; \boldsymbol{\mu}) - f_h^{(1)}(\mathbf{x}_{i,j,k}, \boldsymbol{\xi}_l; \boldsymbol{\mu})}{\Delta x} \\ - \Delta t \max(\xi_{v_l}, 0) \frac{f_h^{(1)}(\mathbf{x}_{i,j,k}, \boldsymbol{\xi}_l; \boldsymbol{\mu}) - f_h^{(1)}(\mathbf{x}_{i,j-1,k}, \boldsymbol{\xi}_l; \boldsymbol{\mu})}{\Delta y} \\ - \Delta t \min(\xi_{v_l}, 0) \frac{f_h^{(1)}(\mathbf{x}_{i,j+1,k}, \boldsymbol{\xi}_l; \boldsymbol{\mu}) - f_h^{(1)}(\mathbf{x}_{i,j,k}, \boldsymbol{\xi}_l; \boldsymbol{\mu})}{\Delta y} \\ - \Delta t \max(\xi_{w_l}, 0) \frac{f_h^{(1)}(\mathbf{x}_{i,j,k}, \boldsymbol{\xi}_l; \boldsymbol{\mu}) - f_h^{(1)}(\mathbf{x}_{i,j,k-1}, \boldsymbol{\xi}_l; \boldsymbol{\mu})}{\Delta z} \\ - \Delta t \min(\xi_{w_l}, 0) \frac{f_h^{(1)}(\mathbf{x}_{i,j,k+1}, \boldsymbol{\xi}_l; \boldsymbol{\mu}) - f_h^{(1)}(\mathbf{x}_{i,j,k}, \boldsymbol{\xi}_l; \boldsymbol{\mu})}{\Delta z} \\ + \Delta t \frac{M_{f_h}(\mathbf{x}_{i,j,k}, \boldsymbol{\xi}_l, t_p; \boldsymbol{\mu}) - f_h^{(1)}(\mathbf{x}_{i,j,k}, \boldsymbol{\xi}_l; \boldsymbol{\mu})}{\tau(\mathbf{x}_{i,j,k}, t_p; \boldsymbol{\mu})}. \end{aligned} \tag{II.13}$$

The initial solution of this system

$$f_h(\mathbf{x}_{i,j,k}, \boldsymbol{\xi}_l, t_0; \boldsymbol{\mu}) = M_{f_h}(\mathbf{x}_{i,j,k}, \boldsymbol{\xi}_l, t_0; \boldsymbol{\mu})$$

corresponds to the initial discrete Maxwellian distribution function computed from the initial state $(\rho_0, \mathbf{u}_0, T_0)$ of the gas. Furthermore, the time-step size is chosen in the HDM according to the CFL condition

$$\Delta t < \min_{1 \leqslant l \leqslant N_{\boldsymbol{\xi}}} \left(\frac{\Delta x}{|\xi_{u_l}|}, \frac{\Delta y}{|\xi_{v_l}|}, \frac{\Delta z}{|\xi_{w_l}|} \right).$$

II.2.2.4 Boundary conditions

To impose the boundary conditions, ghost cells are employed at boundary. These ones contain the density distribution function f_{bc} determined by the boundary conditions. Let $\boldsymbol{\sigma} \in \partial\Omega_{\mathbf{x}}$ be a boundary point of $\Omega_{\mathbf{x}}$. In the following, $f_{bc}(\boldsymbol{\sigma}^+, \boldsymbol{\xi}, t; \boldsymbol{\mu})$ denotes the density distribution function contained in the ghost cell that shares an interface, at point $\boldsymbol{\sigma}$, with the interior cell containing the density distribution function $f_h(\boldsymbol{\sigma}^-, \boldsymbol{\xi}, t; \boldsymbol{\mu})$.

Free flow. For a free flow boundary condition, the density distribution functions in the ghost cell and in the cell centered at point $\boldsymbol{\sigma}^- \in \Omega_{\mathbf{x}}$ are the same

$$f_{bc}(\boldsymbol{\sigma}^+, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = f_h(\boldsymbol{\sigma}^-, \boldsymbol{\xi}, t; \boldsymbol{\mu}).$$

Inflow/outflow. The inflow (or outflow) is represented by a fluid state in the ghost cell. This fluid state depends on the density ρ_{bc}, the macroscopic velocity $\mathbf{u}_{bc}$ and the temperature T_{bc}, and it corresponds to

$$f_{bc}(\boldsymbol{\sigma}^+, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = M_{f_h}[\rho_{bc}, \mathbf{u}_{bc}, T_{bc}](\boldsymbol{\sigma}, \boldsymbol{\xi}, t; \boldsymbol{\mu}), \tag{II.14}$$

where $M_{f_h}[\rho_{bc}, \mathbf{u}_{bc}, T_{bc}]$ denotes the discrete Maxwellian distribution function determined by $(\rho_{bc}, \mathbf{u}_{bc}, T_{bc})$.

Specular reflection. The wall specular reflection is a wall reflecting the particles in opposite normal direction. The wall is moving with macroscopic velocity $\mathbf{u}_{bc}$, and there is no mass and energy fluxes through the wall. The microscopic velocity of the particles becomes after collision

$$\boldsymbol{\xi}_{refl} = \boldsymbol{\xi} - 2((\boldsymbol{\xi} - \mathbf{u}_{bc}(\boldsymbol{\sigma}, t; \boldsymbol{\mu})) \cdot \mathbf{n}_w)\ \mathbf{n}_w,$$

where $\mathbf{n}_w$ denotes the outward unit normal at the wall. The particles hitting the wall verify $0 < (\boldsymbol{\xi} - \mathbf{u}_{bc}) \cdot \mathbf{n}_w$, and the boundary condition reads

$$f_{bc}(\boldsymbol{\sigma}^+, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \begin{cases} f_h(\boldsymbol{\sigma}^-, \boldsymbol{\xi}_{refl}, t; \boldsymbol{\mu}) & \text{if } 0 < (\boldsymbol{\xi} - \mathbf{u}_{bc}(\boldsymbol{\sigma}, t; \boldsymbol{\mu})) \cdot \mathbf{n}_w \\ f_h(\boldsymbol{\sigma}^-, \boldsymbol{\xi}, t; \boldsymbol{\mu}) & \text{otherwise.} \end{cases}$$

Diffuse reflection. The wall diffuse reflection is a wall reflecting the particles as a Maxwellian distribution function. The macroscopic velocity and temperature of the wall are $\mathbf{u}_{bc}$ and T_{bc}, respectively, and there is no mass flux through the wall. In the ghost cell, the boundary condition is represented by the Maxwellian distribution function determined by $(\rho_{bc}, \mathbf{u}_{bc}, T_{bc})$ with the density ρ_{bc} computed to guarantee zero mass flux:

$$\int_{(\boldsymbol{\xi}-\mathbf{u}_{bc})\cdot\mathbf{n}_w<0} (\boldsymbol{\xi} - \mathbf{u}_{bc}) f_h \,\mathrm{d}\boldsymbol{\xi} + \int_{(\boldsymbol{\xi}-\mathbf{u}_{bc})\cdot\mathbf{n}_w>0} (\boldsymbol{\xi} - \mathbf{u}_{bc}) M_f[\rho_{bc}, \mathbf{u}_{bc}, T_{bc}] \,\mathrm{d}\boldsymbol{\xi} = 0,$$

$$\int_{(\boldsymbol{\xi}-\mathbf{u}_{bc})\cdot\mathbf{n}_w<0} (\boldsymbol{\xi} - \mathbf{u}_{bc}) f_h \,\mathrm{d}\boldsymbol{\xi} + \int_{(\boldsymbol{\xi}-\mathbf{u}_{bc})\cdot\mathbf{n}_w>0} (\boldsymbol{\xi} - \mathbf{u}_{bc}) \rho_{bc} M_f[1, \mathbf{u}_{bc}, T_{bc}] \,\mathrm{d}\boldsymbol{\xi} = 0,$$

$$\rho_{bc} = -\frac{\displaystyle\int_{(\boldsymbol{\xi}-\mathbf{u}_{bc})\cdot\mathbf{n}_w<0} (\boldsymbol{\xi} - \mathbf{u}_{bc}) f_h \,\mathrm{d}\boldsymbol{\xi}}{\displaystyle\int_{(\boldsymbol{\xi}-\mathbf{u}_{bc})\cdot\mathbf{n}_w>0} (\boldsymbol{\xi} - \mathbf{u}_{bc}) M_{f_h}[1, \mathbf{u}_{bc}, T_{bc}] \,\mathrm{d}\boldsymbol{\xi}},$$

where we have substituted M_{f_h} to M_f. Moreover, only the particles hitting the wall are reflected, yielding to

$$f_{bc}(\boldsymbol{\sigma}^+, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \begin{cases} M_{f_h}[\rho_{bc}, \mathbf{u}_{bc}, T_{bc}](\boldsymbol{\sigma}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) & \text{if } 0 < (\boldsymbol{\xi} - \mathbf{u}_{bc}(\boldsymbol{\sigma}, t; \boldsymbol{\mu})) \cdot \mathbf{n}_w \\ f_h(\boldsymbol{\sigma}^-, \boldsymbol{\xi}, t; \boldsymbol{\mu}) & \text{otherwise.} \end{cases} \tag{II.15}$$

II.3 Reduced-order model

The HDM simulations are computationally expensive due to the large number of dimensions, i.e. d in space + d in velocity + 1 in time. For this reason, we develop in this thesis [22] a stable, accurate and efficient reduced-order model to compute approximations of the density distribution function at low cost with respect to the HDM. This ROM presents a new reduced-order approximation of the BGK equation where the gas density distribution function is represented in velocity space by a small number of basis functions. The construction of the ROM adopts an approach based on Proper Orthogonal Decomposition [87, 48, 103, 20] in the training stage and on the Galerkin method in the prediction stage. This approach is then adapted to the case of the BGK equation, and the ROM is modified in order to conserve the mass, momentum and energy of the gas. Moreover, we derive the CFL condition ensuring a stable ROM in 1D.

II.3.1 Solution approximation

In the ROM, the discrete density distribution function f_h is approximated in velocity space by a small number of basis functions Φ_n^f:

$$\widetilde{f_h}(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \sum_{n=1}^{N_{pod}} a_n^f(\mathbf{x}, t; \boldsymbol{\mu}) \Phi_n^f(\boldsymbol{\xi})$$

in order to reduce the computational complexity of the model. The basis functions Φ_n^f are constructed in the training stage by Proper Orthogonal Decomposition (POD), and the reduced coordinates a_n^f are computed at low cost by the Galerkin method during the prediction stage.

In 1D and 2D, the discrete density distribution functions ϕ_h and ψ_h are approximated in the same way by

$$\widetilde{\phi}_h(x, \xi_u, t; \boldsymbol{\mu}) = \sum_{n=1}^{N_{pod}^{\phi}} a_n^{\phi}(x, t; \boldsymbol{\mu}) \Phi_n^{\phi}(\xi_u), \ \widetilde{\psi}_h(x, \xi_u, t; \boldsymbol{\mu}) = \sum_{n=1}^{N_{pod}^{\psi}} a_n^{\psi}(x, t; \boldsymbol{\mu}) \Phi_n^{\psi}(\xi_u) \ \text{ in 1D,}$$

$$\widetilde{\phi}_h(\mathbf{x}, \boldsymbol{\xi}_2, t; \boldsymbol{\mu}) = \sum_{n=1}^{N_{pod}^{\phi}} a_n^{\phi}(\mathbf{x}, t; \boldsymbol{\mu}) \Phi_n^{\phi}(\boldsymbol{\xi}_2), \ \widetilde{\psi}_h(\mathbf{x}, \boldsymbol{\xi}_2, t; \boldsymbol{\mu}) = \sum_{n=1}^{N_{pod}^{\psi}} a_n^{\psi}(\mathbf{x}, t; \boldsymbol{\mu}) \Phi_n^{\psi}(\boldsymbol{\xi}_2) \ \text{ in 2D.}$$

We will detail the model for $\widetilde{f_h}$ in the following. For $\widetilde{\phi}_h$ and $\widetilde{\psi}_h$, the formulation is similar.

II.3.2 Training stage

The basis functions are constructed in the training stage by POD (Section I.4). In this approach, the HDM provides snapshots of the density distribution function

to learn the solution manifold. This sampling is enriched with snapshots of the Maxwellian distribution function since this one is also represented by the basis functions. Then, the POD is used to extract the low-dimensional trial subspace spanned by the basis functions, which is optimal in the least-squares sense to approximate these snapshots. In this work, the POD is performed by the classical method (Section I.4.3.1) due to the large number of snapshots collected during the sampling of the solution manifold.

II.3.2.1 Snapshot database

Let $s_l^f(\boldsymbol{\xi}) = f_h(\mathbf{x}_{i(l),j(l),k(l)}, \boldsymbol{\xi}, t_{p(l)}; \boldsymbol{\mu}_{q(l)})$ be a snapshot of the discrete density distribution function collected at point $\mathbf{x}_{i(l),j(l),k(l)}$, time instance $t_{p(l)}$ and input parameter $\boldsymbol{\mu}_{q(l)}$. The snapshots are provided by the HDM and are taken at every point of the physical space and uniformly in time. In this way, the snapshots are uniformly distributed to represent the discrete density distribution function for each input parameter. The N_s snapshots provided by this sampling are stored in the snapshot matrix

$$\mathbf{S}_f = \begin{pmatrix} s_1^f(\boldsymbol{\xi}_1) & s_2^f(\boldsymbol{\xi}_1) & \cdots & s_{N_s}^f(\boldsymbol{\xi}_1) \\ s_1^f(\boldsymbol{\xi}_2) & s_2^f(\boldsymbol{\xi}_2) & \cdots & s_{N_s}^f(\boldsymbol{\xi}_2) \\ \vdots & \vdots & \ddots & \vdots \\ s_1^f(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) & s_2^f(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) & \cdots & s_{N_s}^f(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) \end{pmatrix} \in \mathbb{R}^{N_{\boldsymbol{\xi}} \times N_s}.$$

In the prediction stage (Section II.3.3), the basis functions also represent the discrete Maxwellian distribution function. For this reason, the basis functions are constructed in order to represent accurately the density and Maxwellian distribution functions. For this purpose, the snapshot database also contains snapshots of the discrete Maxwellian distribution function $s_l^{M_f}(\boldsymbol{\xi}) = M_{f_h}(\mathbf{x}_{i(l),j(l),k(l)}, \boldsymbol{\xi}, t_{p(l)}; \boldsymbol{\mu}_{q(l)})$, collected in the same way as the discrete density distribution function. These snapshots are stored in the matrix

$$\mathbf{S}_{M_f} = \begin{pmatrix} s_1^{M_f}(\boldsymbol{\xi}_1) & s_2^{M_f}(\boldsymbol{\xi}_1) & \cdots & s_{N_s}^{M_f}(\boldsymbol{\xi}_1) \\ s_1^{M_f}(\boldsymbol{\xi}_2) & s_2^{M_f}(\boldsymbol{\xi}_2) & \cdots & s_{N_s}^{M_f}(\boldsymbol{\xi}_2) \\ \vdots & \vdots & \ddots & \vdots \\ s_1^{M_f}(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) & s_2^{M_f}(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) & \cdots & s_{N_s}^{M_f}(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) \end{pmatrix} \in \mathbb{R}^{N_{\boldsymbol{\xi}} \times N_s},$$

and the complete snapshot database is

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_f & \mathbf{S}_{M_f} \end{pmatrix} \in \mathbb{R}^{N_{\boldsymbol{\xi}} \times (2N_s)}. \tag{II.16}$$

The results of this modification are presented in Section II.4.2. In Figure II.1, we show 1D and 2D examples of snapshots collected at a low Knudsen number ($Kn = 10^{-5}$). In this case, the regime of the gas flow is hydrodynamic, and the snapshots are close to the Maxwellian distribution function.

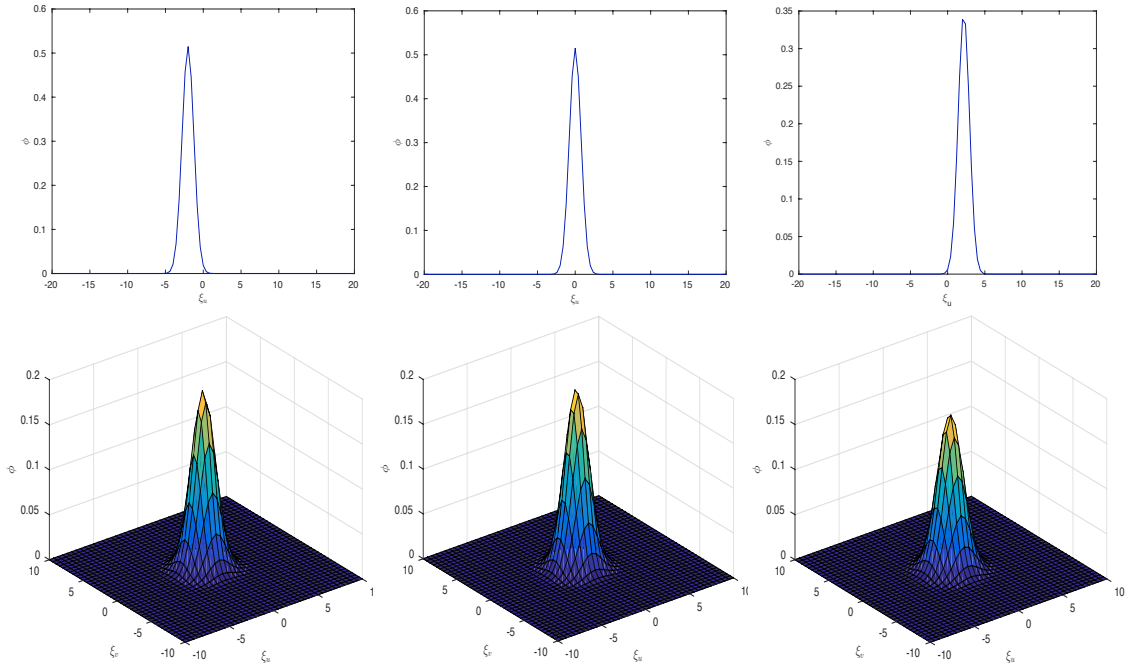


Figure II.1: Examples of snapshots of ϕ_h in 1D (top) and in 2D (bottom) randomly choosen.

II.3.2.2 Proper Orthogonal Decomposition

Given the snapshot database $\mathbf{S}$, the trial subspace is defined as the linear subspace of rank N_{pod} minimizing, in the least-squares sense, the difference between the snapshots and their projections onto this subspace:

$$\begin{cases} \underset{\boldsymbol{\Phi}\in\mathbb{R}^{N_{\boldsymbol{\xi}}\times N_{pod}}}{\text{minimize}} & \left\|\mathbf{S}-\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{S}\right\|_{F_{\boldsymbol{\Theta}}}^2 \\ \text{subject to} & \boldsymbol{\Phi}^T\boldsymbol{\Theta}\boldsymbol{\Phi}=\mathbf{I}_{N_{pod}}. \end{cases}$$

Here, the basis functions are stored in the matrix

$$\boldsymbol{\Phi}=\begin{pmatrix} \Phi_1^f(\boldsymbol{\xi}_1) & \Phi_2^f(\boldsymbol{\xi}_1) & \cdots & \Phi_{N_{pod}}^f(\boldsymbol{\xi}_1) \\ \Phi_1^f(\boldsymbol{\xi}_2) & \Phi_2^f(\boldsymbol{\xi}_2) & \cdots & \Phi_{N_{pod}}^f(\boldsymbol{\xi}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_1^f(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) & \Phi_2^f(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) & \cdots & \Phi_{N_{pod}}^f(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) \end{pmatrix}\in\mathbb{R}^{N_{\boldsymbol{\xi}}\times N_{pod}},$$

and the matrix $\boldsymbol{\Theta}\in\mathbb{R}^{N_{\boldsymbol{\xi}}\times N_{\boldsymbol{\xi}}}$ is defined in Section II.2.2.1. According to the Schmidt-Eckart-Young-Mirsky theorem 1, the basis functions are given by

$$\boldsymbol{\Phi}=(\boldsymbol{\Theta}^{\frac{1}{2}})^{-T}\begin{pmatrix} U_{1,1} & \cdots & U_{1,N_{pod}} \\ \vdots & & \vdots \\ U_{N_{\boldsymbol{\xi}},1} & \cdots & U_{N_{\boldsymbol{\xi}},N_{pod}} \end{pmatrix},$$

where $\widetilde{\mathbf{S}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ is the singular value decomposition (SVD) of $\widetilde{\mathbf{S}} = \mathbf{\Theta}^{\frac{1}{2}}\mathbf{S}$ and $\mathbf{\Theta} = \mathbf{\Theta}^{\frac{1}{2}}(\mathbf{\Theta}^{\frac{1}{2}})^T$ is the Cholesky decomposition of $\mathbf{\Theta}$. Since the sampling is performed over physical space, time and parameter domain, the number of snapshots is in practice too large ($N_s \approx O(10^7)$) to find the SVD of $\widetilde{\mathbf{S}}$. For this reason, the basis functions are computed by the classical method, where $\mathbf{U}$ is obtained from the SVD of $\widetilde{\mathbf{S}}\widetilde{\mathbf{S}}^T \in \mathbb{R}^{N_{\boldsymbol{\xi}} \times N_{\boldsymbol{\xi}}}$ instead of $\widetilde{\mathbf{S}} \in \mathbb{R}^{N_{\boldsymbol{\xi}} \times (2N_s)}$.

Figure II.2 shows examples of basis functions obtained with this method. In this case, the snapshots (see Figure II.1) are close to the Maxwellian distribution functions, and the first basis function Φ_1^ϕ (on the left) is close to the mean of the snapshots.

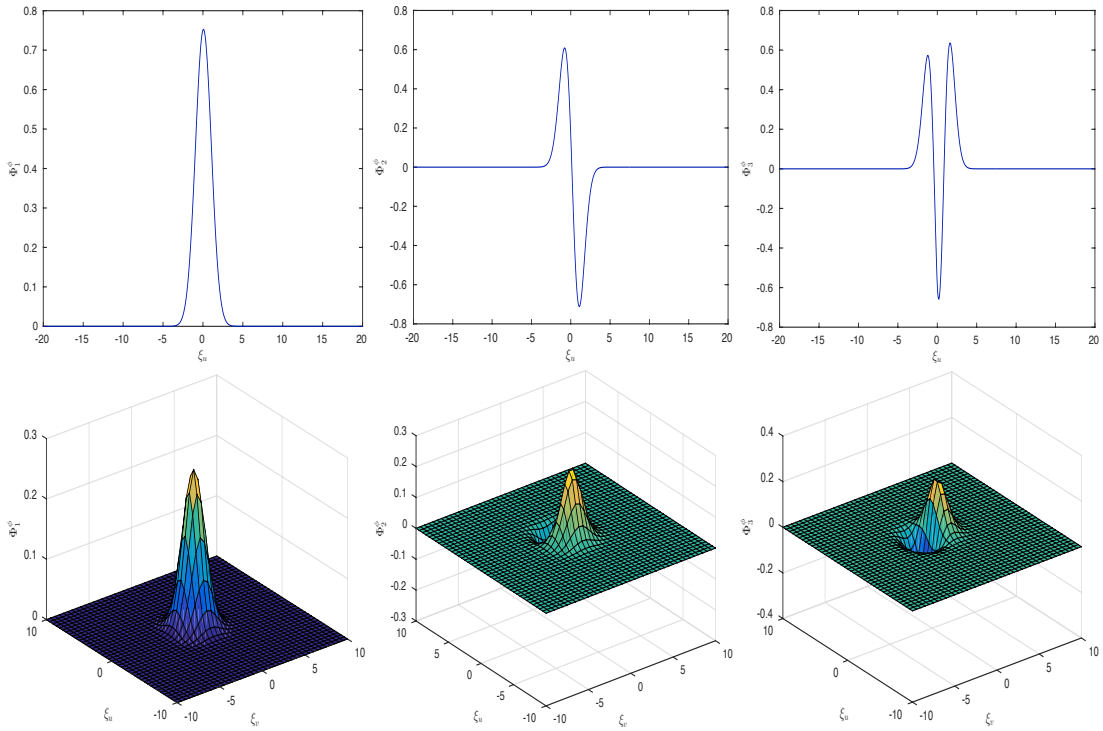


Figure II.2: Examples of basis functions for ϕ_h in 1D (top) and in 2D (bottom).

II.3.3 Prediction stage

Once the basis functions are constructed, the approximate density distribution function depends only on the reduced coordinates. These ones are determined at low cost during the prediction stage by the Galerkin method (Section I.3.2.1). In this approach, the residual is enforced to be orthogonal to the trial subspace, leading to the resolution of a small-scale system which is hyperbolic by construction. This system is then modified in order to preserve important properties (II.3) of the HDM. Finally, the ROM is discretized by the same numerical methods used in the HDM, and we derive the CFL condition ensuring a stable ROM in 1D.

II.3.3.1 Galerkin method

In the Galerkin method, the approximate density distribution function is inserted into the system of transport equations (II.10), yielding to the residual

$$r(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}) = \frac{\partial \widetilde{f_h}}{\partial t}(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}) + \boldsymbol{\xi}\cdot\nabla_{\mathbf{x}}\widetilde{f_h}(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}) - \frac{M_{f_h}(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}) - \widetilde{f_h}(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu})}{\tau(\mathbf{x},t;\boldsymbol{\mu})}.$$

This residual is then projected onto the basis functions

$$\forall n \in \{1,\dots,N_{pod}\} : \ \left\langle r(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}), \Phi_n^f(\boldsymbol{\xi})\right\rangle_{\boldsymbol{\Theta}} = 0,$$

leading to the system verified by the reduced coordinates:

$$\frac{\partial \mathbf{a}^f}{\partial t} + \mathbf{A}\frac{\partial \mathbf{a}^f}{\partial x} + \mathring{\mathbf{A}}\frac{\partial \mathbf{a}^f}{\partial y} + \overset{\star}{\mathbf{A}}\frac{\partial \mathbf{a}^f}{\partial z} = \frac{\mathbf{a}^{M_f} - \mathbf{a}^f}{\tau}, \tag{II.17}$$

where $A_{n,m}=\left\langle \xi_u \Phi_m^f(\boldsymbol{\xi}), \Phi_n^f(\boldsymbol{\xi})\right\rangle_{\boldsymbol{\Theta}}$, $\mathring{A}_{n,m}=\left\langle \xi_v \Phi_m^f(\boldsymbol{\xi}), \Phi_n^f(\boldsymbol{\xi})\right\rangle_{\boldsymbol{\Theta}}$, $\overset{\star}{A}_{n,m}=\left\langle \xi_w \Phi_m^f(\boldsymbol{\xi}), \Phi_n^f(\boldsymbol{\xi})\right\rangle_{\boldsymbol{\Theta}}$, $a_n^{M_f}(\mathbf{x},t,\boldsymbol{\mu}) = \left\langle M_{f_h}(\mathbf{x},\boldsymbol{\xi},t,\boldsymbol{\mu}), \Phi_n^f(\boldsymbol{\xi})\right\rangle_{\boldsymbol{\Theta}}$, $\mathbf{a}^f(\mathbf{x},t,\boldsymbol{\mu}) = (a_1^f(\mathbf{x},t,\boldsymbol{\mu}),\dots,a_{N_{pod}}^f(\mathbf{x},t,\boldsymbol{\mu}))^T$ and $\mathbf{a}^{M_f}(\mathbf{x},t,\boldsymbol{\mu}) = (a_1^{M_f}(\mathbf{x},t,\boldsymbol{\mu}),\dots,a_{N_{pod}}^{M_f}(\mathbf{x},t,\boldsymbol{\mu}))^T$. The matrices $\mathbf{A},\mathring{\mathbf{A}},\overset{\star}{\mathbf{A}} \in \mathbb{R}^{N_{pod}\times N_{pod}}$ are symmetric and thus diagonalizable by a real orthogonal similarity. This system is therefore hyperbolic, and the equations can be decoupled direction by direction with a linear change of variables. Let the eigendecompositions be $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^T$, where $\mathbf{D} \in \mathbb{R}^{N_{pod}\times N_{pod}}$ is a diagonal matrix and $\mathbf{P} \in \mathbb{R}^{N_{pod}\times N_{pod}}$ is an orthogonal matrix. The hyperbolic system (II.17) becomes

$$\frac{\partial \mathbf{a}^f}{\partial t} + \mathbf{P}\mathbf{D}\frac{\partial \mathbf{b}^f}{\partial x} + \mathring{\mathbf{P}}\mathring{\mathbf{D}}\frac{\partial \mathbf{c}^f}{\partial y} + \overset{\star}{\mathbf{P}}\overset{\star}{\mathbf{D}}\frac{\partial \mathbf{d}^f}{\partial z} = \frac{\mathbf{a}^{M_f} - \mathbf{a}^f}{\tau}, \tag{II.18}$$

where the changes of variables are $\mathbf{b}^f = \mathbf{P}^T\mathbf{a}^f$, $\mathbf{c}^f = \mathring{\mathbf{P}}^T\mathbf{a}^f$ and $\mathbf{d}^f = \overset{\star}{\mathbf{P}}^T\mathbf{a}^f$.

II.3.3.2 Preservation of properties of the HDM

In system (II.18), the discrete Maxwellian distribution function is projected onto the basis functions:

$$a_n^{M_f}(\mathbf{x},t;\boldsymbol{\mu}) = \left\langle M_{f_h}(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}), \Phi_n^f(\boldsymbol{\xi})\right\rangle_{\boldsymbol{\Theta}}, \tag{II.19}$$

and the approximate Maxwellian distribution function $\widetilde{M_{f_h}}$ is given by

$$\widetilde{M_{f_h}}(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}) = \sum_{n=1}^{N_{pod}} a_n^{M_f}(\mathbf{x},t;\boldsymbol{\mu})\Phi_n^f(\boldsymbol{\xi}).$$

Due to projection error, the conservation of mass, momentum and energy is not necessarily preserved:

$$\left\langle \widetilde{M_{f_h}}(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}), \begin{pmatrix} 1 \\ \boldsymbol{\xi} \\ \frac{\|\boldsymbol{\xi}\|_2^2}{2} \end{pmatrix} \right\rangle_{\boldsymbol{\Theta}} \neq \begin{pmatrix} \rho(\mathbf{x},t;\boldsymbol{\mu}) \\ \rho(\mathbf{x},t;\boldsymbol{\mu})\mathbf{u}(\mathbf{x},t;\boldsymbol{\mu}) \\ E(\mathbf{x},t;\boldsymbol{\mu}) \end{pmatrix}.$$

For this reason, the reduced coordinates $a_n^{M_f}$ are not computed from equation (II.19). The approximate Maxwellian distribution function is determined to conserve the mass, momentum and energy of the gas and to be as close as possible to the Maxwellian distribution function M_f:

$$\begin{cases} \underset{\mathbf{a}^{M_f}(\mathbf{x},t;\boldsymbol{\mu})\in\mathbb{R}^{N_{pod}}}{\text{minimize}} & \left\|\widetilde{M}_{f_h}(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}) - M_f(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu})\right\|_{\boldsymbol{\Theta}}^2 \\ \text{subject to} & \left\langle \widetilde{M}_{f_h}(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}), \begin{pmatrix} 1 \\ \boldsymbol{\xi} \\ \frac{\|\boldsymbol{\xi}\|^2}{2} \end{pmatrix} \right\rangle_{\boldsymbol{\Theta}} = \begin{pmatrix} \rho(\mathbf{x},t;\boldsymbol{\mu}) \\ \rho(\mathbf{x},t;\boldsymbol{\mu})\mathbf{u}(\mathbf{x},t;\boldsymbol{\mu}) \\ E(\mathbf{x},t;\boldsymbol{\mu}) \end{pmatrix}. \end{cases} \tag{II.20}$$

The objective function of this minimization problem (II.20) can be cast in matrix format using the $N_{\boldsymbol{\xi}} \times N_{pod}$ system

$$\underbrace{\begin{pmatrix} \Phi_1^f(\boldsymbol{\xi}_1) & \Phi_2^f(\boldsymbol{\xi}_1) & \cdots & \Phi_{N_{pod}}^f(\boldsymbol{\xi}_1) \\ \Phi_1^f(\boldsymbol{\xi}_2) & \Phi_2^f(\boldsymbol{\xi}_2) & \cdots & \Phi_{N_{pod}}^f(\boldsymbol{\xi}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_1^f(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) & \Phi_2^f(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) & \cdots & \Phi_{N_{pod}}^f(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) \end{pmatrix}}_{\boldsymbol{\Phi}} \underbrace{\begin{pmatrix} a_1^{M_f}(\mathbf{x},t;\boldsymbol{\mu}) \\ a_2^{M_f}(\mathbf{x},t;\boldsymbol{\mu}) \\ \vdots \\ a_{N_{pod}}^{M_f}(\mathbf{x},t;\boldsymbol{\mu}) \end{pmatrix}}_{\mathbf{a}^{M_f}(\mathbf{x},t;\boldsymbol{\mu})} \approx \underbrace{\begin{pmatrix} M_f(\mathbf{x},\boldsymbol{\xi}_1,t;\boldsymbol{\mu}) \\ M_f(\mathbf{x},\boldsymbol{\xi}_2,t;\boldsymbol{\mu}) \\ \vdots \\ M_f(\mathbf{x},\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}},t;\boldsymbol{\mu}) \end{pmatrix}}_{\mathbf{M}_f(\mathbf{x},t;\boldsymbol{\mu})}.$$

Likewise, the constraints in problem (II.20) lead to the $5 \times N_{pod}$ system

$$\underbrace{\begin{pmatrix} \left\langle \Phi_1^f(\boldsymbol{\xi}),1\right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}}^f(\boldsymbol{\xi}),1\right\rangle_{\boldsymbol{\Theta}} \\ \left\langle \Phi_1^f(\boldsymbol{\xi}),\xi_u\right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}}^f(\boldsymbol{\xi}),\xi_u\right\rangle_{\boldsymbol{\Theta}} \\ \left\langle \Phi_1^f(\boldsymbol{\xi}),\xi_v\right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}}^f(\boldsymbol{\xi}),\xi_v\right\rangle_{\boldsymbol{\Theta}} \\ \left\langle \Phi_1^f(\boldsymbol{\xi}),\xi_w\right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}}^f(\boldsymbol{\xi}),\xi_w\right\rangle_{\boldsymbol{\Theta}} \\ \left\langle \Phi_1^f(\boldsymbol{\xi}),\frac{\|\boldsymbol{\xi}\|_2^2}{2}\right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}}^f(\boldsymbol{\xi}),\frac{\|\boldsymbol{\xi}\|_2^2}{2}\right\rangle_{\boldsymbol{\Theta}} \end{pmatrix}}_{\boldsymbol{\Psi}} \underbrace{\begin{pmatrix} a_1^{M_f}(\mathbf{x},t;\boldsymbol{\mu}) \\ a_2^{M_f}(\mathbf{x},t;\boldsymbol{\mu}) \\ \vdots \\ a_{N_{pod}}^{M_f}(\mathbf{x},t;\boldsymbol{\mu}) \end{pmatrix}}_{\mathbf{a}^{M_f}(\mathbf{x},t;\boldsymbol{\mu})} = \underbrace{\begin{pmatrix} \rho(\mathbf{x},t;\boldsymbol{\mu}) \\ \rho(\mathbf{x},t;\boldsymbol{\mu})u(\mathbf{x},t;\boldsymbol{\mu}) \\ \rho(\mathbf{x},t;\boldsymbol{\mu})v(\mathbf{x},t;\boldsymbol{\mu}) \\ \rho(\mathbf{x},t;\boldsymbol{\mu})w(\mathbf{x},t;\boldsymbol{\mu}) \\ E(\mathbf{x},t;\boldsymbol{\mu}) \end{pmatrix}}_{\boldsymbol{\rho}(\mathbf{x},t;\boldsymbol{\mu})}.$$

The solution to (II.20) is then given by the method of Lagrange multipliers:

$$\mathbf{a}^{M_f} = \boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{M}_f + \boldsymbol{\Psi}^T(\boldsymbol{\Psi}\boldsymbol{\Psi}^T)^{-1}\left(\boldsymbol{\rho} - \boldsymbol{\Psi}\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{M}_f\right). \tag{II.21}$$

If $\boldsymbol{\Psi}\boldsymbol{\Psi}^T$ is singular, there is no solution satisfying the constraints. In this case, we search the best approximation in the least-squares sense of the constraints that minimizes the objective function. The corresponding approximate Maxwellian distribution function is given by

$$\mathbf{a}^{M_f}(\mathbf{x},t;\boldsymbol{\mu}) = \boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{M}_f(\mathbf{x},t;\boldsymbol{\mu}) + \boldsymbol{\Psi}^+\left(\boldsymbol{\rho}(\mathbf{x},t;\boldsymbol{\mu}) - \boldsymbol{\Psi}\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{M}_f(\mathbf{x},t;\boldsymbol{\mu})\right), \tag{II.22}$$

where $\boldsymbol{\Psi}^+$ denotes the Moore-Penrose inverse of $\boldsymbol{\Psi}$. When $\boldsymbol{\Psi}\boldsymbol{\Psi}^T$ is invertible, these two formulations (II.21) and (II.22) are equivalent since $\boldsymbol{\Psi}^+ = \boldsymbol{\Psi}^T(\boldsymbol{\Psi}\boldsymbol{\Psi}^T)^{-1}$.

Moreover, the matrices $\mathbf{\Phi}^T\mathbf{\Theta}$, $\mathbf{\Psi}^+$ and $\mathbf{\Psi}\mathbf{\Phi}^T\mathbf{\Theta}$ are pre-computed offline to save computing time. This modification is used in the Galerkin projection (II.18), in the initial condition (II.23) and in the boundary conditions (II.14) and (II.15), see [22] for more details. The results of this modification are presented in Section II.4.1. In 1D and 2D, the approximate equilibrium distribution functions $\widetilde{M_{\phi_h}}$ and $\widetilde{M_{\psi_h}}$ are determined in the same way, see Appendix A.

II.3.3.3 Numerical methods

The hyperbolic system (II.18) is solved by the finite volume method in space and an IMEX Runge-Kutta scheme in time, as in the HDM. The resulting ROM is a first-order scheme, and we derive the CFL condition ensuring a stable ROM in 1D. This CFL condition leads in particular to larger time-step size Δt than those used in the HDM, allowing to further reduce the computational cost of the ROM.

Physical space discretization. For simplicity in this work, the HDM and ROM both use the same mesh to discretize the physical space. On each cell, the convective term is discretized by the finite volume method, and the collision term is approximated by a centered approximation. Since the reduced coordinates b_n^f are transported in the x-direction at constant speed $D_{n,n}$ in the hyperbolic system (II.18), the first-order finite volume scheme reads on cartesian grid

$$D_{n,n}\frac{\partial b_n^f}{\partial x}(\mathbf{x},t;\boldsymbol{\mu}) = \frac{F^n_{i+\frac{1}{2},j,k} - F^n_{i-\frac{1}{2},j,k}}{\Delta x},$$

where the flux $F^n_{i+\frac{1}{2},j,k}$ between the cells $K_{i,j,k}$ and $K_{i+1,j,k}$ is

$$F^n_{i+\frac{1}{2},j,k} = \max(D_{n,n},0)b_n^f(\mathbf{x}_{i,j,k},t;\boldsymbol{\mu}) + \min(D_{n,n},0)b_n^f(\mathbf{x}_{i+1,j,k},t;\boldsymbol{\mu}).$$

By using the change of variables $\mathbf{b}^f = \mathbf{P}^T\mathbf{a}^f$, the hyperbolic system (II.18) becomes

$$\begin{aligned}
\frac{\partial \mathbf{a}^f}{\partial t}(\mathbf{x}_{i,j,k},t;\boldsymbol{\mu}) = &- \mathbf{P}\max(\mathbf{D},\mathbf{0})\mathbf{P}^T\frac{\mathbf{a}^f(\mathbf{x}_{i,j,k},t;\boldsymbol{\mu}) - \mathbf{a}^f(\mathbf{x}_{i-1,j,k},t;\boldsymbol{\mu})}{\Delta x}\\
&- \mathbf{P}\min(\mathbf{D},\mathbf{0})\mathbf{P}^T\frac{\mathbf{a}^f(\mathbf{x}_{i+1,j,k},t;\boldsymbol{\mu}) - \mathbf{a}^f(\mathbf{x}_{i,j,k},t;\boldsymbol{\mu})}{\Delta x}\\
&- \mathring{\mathbf{P}}\max(\mathring{\mathbf{D}},\mathbf{0})\mathring{\mathbf{P}}^T\frac{\mathbf{a}^f(\mathbf{x}_{i,j,k},t;\boldsymbol{\mu}) - \mathbf{a}^f(\mathbf{x}_{i,j-1,k},t;\boldsymbol{\mu})}{\Delta y}\\
&- \mathring{\mathbf{P}}\min(\mathring{\mathbf{D}},\mathbf{0})\mathring{\mathbf{P}}^T\frac{\mathbf{a}^f(\mathbf{x}_{i,j+1,k},t;\boldsymbol{\mu}) - \mathbf{a}^f(\mathbf{x}_{i,j,k},t;\boldsymbol{\mu})}{\Delta y}\\
&- \overset{\star}{\mathbf{P}}\max(\overset{\star}{\mathbf{D}},\mathbf{0})\overset{\star}{\mathbf{P}}^T\frac{\mathbf{a}^f(\mathbf{x}_{i,j,k},t;\boldsymbol{\mu}) - \mathbf{a}^f(\mathbf{x}_{i,j,k-1},t;\boldsymbol{\mu})}{\Delta z}\\
&- \overset{\star}{\mathbf{P}}\min(\overset{\star}{\mathbf{D}},\mathbf{0})\overset{\star}{\mathbf{P}}^T\frac{\mathbf{a}^f(\mathbf{x}_{i,j,k+1},t;\boldsymbol{\mu}) - \mathbf{a}^f(\mathbf{x}_{i,j,k},t;\boldsymbol{\mu})}{\Delta z}\\
&+ \frac{\mathbf{a}^{M_f}(\mathbf{x}_{i,j,k},t;\boldsymbol{\mu}) - \mathbf{a}^f(\mathbf{x}_{i,j,k},t;\boldsymbol{\mu})}{\tau(\mathbf{x}_{i,j,k},t;\boldsymbol{\mu})},
\end{aligned}$$

where $\max(\mathbf{D},\mathbf{0})$ denotes the diagonal matrix $\mathbf{D}$ with the negative elements replaced by 0. The matrices $\mathbf{P}\max(\mathbf{D},\mathbf{0})\mathbf{P}^T$ and $\mathbf{P}\min(\mathbf{D},\mathbf{0})\mathbf{P}^T$ can be pre-computed offline to save computing time. Moreover, the boundary conditions are given by projecting the boundary density distribution function f_{bc} (Section II.2.2.4) onto the basis functions, see [22] for more details.

Time discretization. The intermediate time-step of the IMEX Runge-Kutta scheme [10, 67, 86] is in explicit form

$$\mathbf{a}^{(1)}(\mathbf{x}_{i,j,k};\boldsymbol{\mu}) = \frac{\tau(\mathbf{x}_{i,j,k},t_p;\boldsymbol{\mu})}{\Delta t+\tau(\mathbf{x}_{i,j,k},t_p;\boldsymbol{\mu})}\left(\mathbf{a}^f(\mathbf{x}_{i,j,k},t_p;\boldsymbol{\mu})+\Delta t\,\frac{\mathbf{a}^{M_f}(\mathbf{x}_{i,j,k},t_p;\boldsymbol{\mu})}{\tau(\mathbf{x}_{i,j,k},t_p;\boldsymbol{\mu})}\right)$$

by following the same arguments used in Section II.2.2.3. The next time-step is then given by

$$\begin{aligned}
\mathbf{a}^f(\mathbf{x}_{i,j,k},t_{p+1};\boldsymbol{\mu}) = \mathbf{a}^f(\mathbf{x}_{i,j,k},t_p;\boldsymbol{\mu})&\\
-\Delta t\mathbf{P}\max(\mathbf{D},\mathbf{0})\mathbf{P}^T&\frac{\mathbf{a}^{(1)}(\mathbf{x}_{i,j,k};\boldsymbol{\mu})-\mathbf{a}^{(1)}(\mathbf{x}_{i-1,j,k};\boldsymbol{\mu})}{\Delta x}\\
-\Delta t\mathbf{P}\min(\mathbf{D},\mathbf{0})\mathbf{P}^T&\frac{\mathbf{a}^{(1)}(\mathbf{x}_{i+1,j,k};\boldsymbol{\mu})-\mathbf{a}^{(1)}(\mathbf{x}_{i,j,k};\boldsymbol{\mu})}{\Delta x}\\
-\Delta t\mathring{\mathbf{P}}\max(\mathring{\mathbf{D}},\mathbf{0})\mathring{\mathbf{P}}^T&\frac{\mathbf{a}^{(1)}(\mathbf{x}_{i,j,k};\boldsymbol{\mu})-\mathbf{a}^{(1)}(\mathbf{x}_{i,j-1,k};\boldsymbol{\mu})}{\Delta y}\\
-\Delta t\mathring{\mathbf{P}}\min(\mathring{\mathbf{D}},\mathbf{0})\mathring{\mathbf{P}}^T&\frac{\mathbf{a}^{(1)}(\mathbf{x}_{i,j+1,k};\boldsymbol{\mu})-\mathbf{a}^{(1)}(\mathbf{x}_{i,j,k};\boldsymbol{\mu})}{\Delta y}\\
-\Delta t\overset{\star}{\mathbf{P}}\max(\overset{\star}{\mathbf{D}},\mathbf{0})\overset{\star}{\mathbf{P}}^T&\frac{\mathbf{a}^{(1)}(\mathbf{x}_{i,j,k};\boldsymbol{\mu})-\mathbf{a}^{(1)}(\mathbf{x}_{i,j,k-1};\boldsymbol{\mu})}{\Delta z}\\
-\Delta t\overset{\star}{\mathbf{P}}\min(\overset{\star}{\mathbf{D}},\mathbf{0})\overset{\star}{\mathbf{P}}^T&\frac{\mathbf{a}^{(1)}(\mathbf{x}_{i,j,k+1};\boldsymbol{\mu})-\mathbf{a}^{(1)}(\mathbf{x}_{i,j,k};\boldsymbol{\mu})}{\Delta z}\\
+\Delta t&\frac{\mathbf{a}^{M_f}(\mathbf{x}_{i,j,k},t_p;\boldsymbol{\mu})-\mathbf{a}^{(1)}(\mathbf{x}_{i,j,k};\boldsymbol{\mu})}{\tau(\mathbf{x}_{i,j,k},t_p;\boldsymbol{\mu})}.
\end{aligned}$$

The approximate density distribution function is initialized from

$$a_n^f(\mathbf{x}_{i,j,k},t_0;\boldsymbol{\mu}) = a_n^{M_f}(\mathbf{x}_{i,j,k},t_0;\boldsymbol{\mu}), \tag{II.23}$$

where the reduced coordinates of the initial approximate Maxwellian distribution function are computed from the initial state of the flow $(\rho_0,\mathbf{u}_0,T_0)$. Moreover, the CFL condition used in this ROM reads

$$\Delta t < \min_{1\leqslant n\leqslant N_{pod}}\left(\frac{\Delta x}{|D_{n,n}|},\frac{\Delta y}{|\mathring{D}_{n,n}|},\frac{\Delta z}{|\overset{\star}{D}_{n,n}|}\right). \tag{II.24}$$

Notably, this CFL condition ensures a stable ROM in 1D and leads to larger time-step sizes than those used in the HDM, allowing to further reduce the computational cost of the ROM by decreasing the number of time-steps.

II.4 Applications

We analyze the performance of the ROM with respect to the HDM for four applications in 1D and 2D. The ROM accuracy is evaluated using the relative approximation error in the predicted density distribution functions at final time t_{max}:

$$\text{Error} = \left(\frac{1}{2} \frac{\left\| \phi_h - \widetilde{\phi}_h \right\|_{L^2(\Omega_{\mathbf{x}} \times \Omega_{\boldsymbol{\xi}} \times \{t_{max}\})}}{\left\| \phi_h \right\|_{L^2(\Omega_{\mathbf{x}} \times \Omega_{\boldsymbol{\xi}} \times \{t_{max}\})}} + \frac{1}{2} \frac{\left\| \psi_h - \widetilde{\psi}_h \right\|_{L^2(\Omega_{\mathbf{x}} \times \Omega_{\boldsymbol{\xi}} \times \{t_{max}\})}}{\left\| \psi_h \right\|_{L^2(\Omega_{\mathbf{x}} \times \Omega_{\boldsymbol{\xi}} \times \{t_{max}\})}} \right) \times 100\%.$$

Furthermore, the computational speedup of the ROM with respect to the HDM is evaluated using the relative run time in order to quantify the reduction in computational cost provided by the ROM:

$$\text{Run time} = \frac{\text{ROM run time}}{\text{HDM run time}} \times 100\%.$$

II.4.1 Reproduction of a shock wave

The first application evaluates different definitions of the approximate Maxwellian distribution function. In the Galerkin method, $\widetilde{M_{f_h}}$ is defined as the projection of the discrete Maxwellian distribution function onto the basis functions, while in Section II.3.3.2, we propose to determine the approximate Maxwellian distribution function by constrained projection in order to conserve the mass, momentum and energy of the gas. We compare these two approaches to compute $\widetilde{M_{f_h}}$.

We consider the Sod shock tube problem [104] at $Kn = 10^{-5}$. The physical space $\Omega_{\mathbf{x}} =]0, 1[$ is discretized using $N_{\mathbf{x}} = 200$ cells, and the velocity space $\Omega_{\boldsymbol{\xi}} =]-10, 10[$ is discretized using $N_{\boldsymbol{\xi}} = 41$ points. The final time is $t_{max} = 0.12$ and the CFL number is 0.1. The initial condition is

$$\begin{cases} \rho_0(x) = 1,\ u_0(x) = 0,\ T_0(x) = 1 & \text{if } x \in]0, 0.5[\\ \rho_0(x) = 0.125,\ u_0(x) = 0,\ T_0(x) = 0.8 & \text{otherwise,} \end{cases}$$

and we consider free flow boundary conditions.

For the construction of the basis functions, the database $\mathbf{S}_\phi$ (resp. $\mathbf{S}_\psi$) contains snapshots of ϕ_h (resp. ψ_h) taken at each point in space and every 0.005 time units. The Figure II.3 shows the squared singular values of $\widetilde{\mathbf{S}}_\phi$ and $\widetilde{\mathbf{S}}_\psi$. The decay of the squared singular values is fast, and 3 basis functions (i.e. 7.3% of the complete basis) are sufficient to obtain a relative squared projection error lower than 0.01%.

In Figure II.4, we plot the macroscopic quantities of interest of the gas at final time obtained by the ROM using the constrained projection-based approach to determine the approximate Maxwellian distribution function.

In Figure II.5, we compare the performance of the two approaches to compute $\widetilde{M_{f_h}}$ as a function of the number of basis functions $N_{pod} = N_{pod}^{\phi} = N_{pod}^{\psi}$.

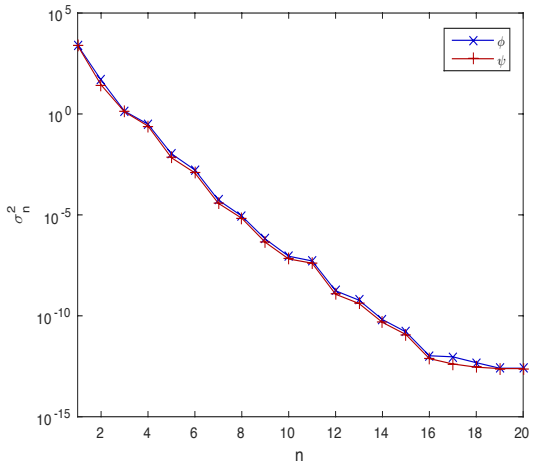


Figure II.3: Squared singular values of $\widetilde{\mathbf{S}}_\phi$ and $\widetilde{\mathbf{S}}_\psi$ for the shock wave reproduction.

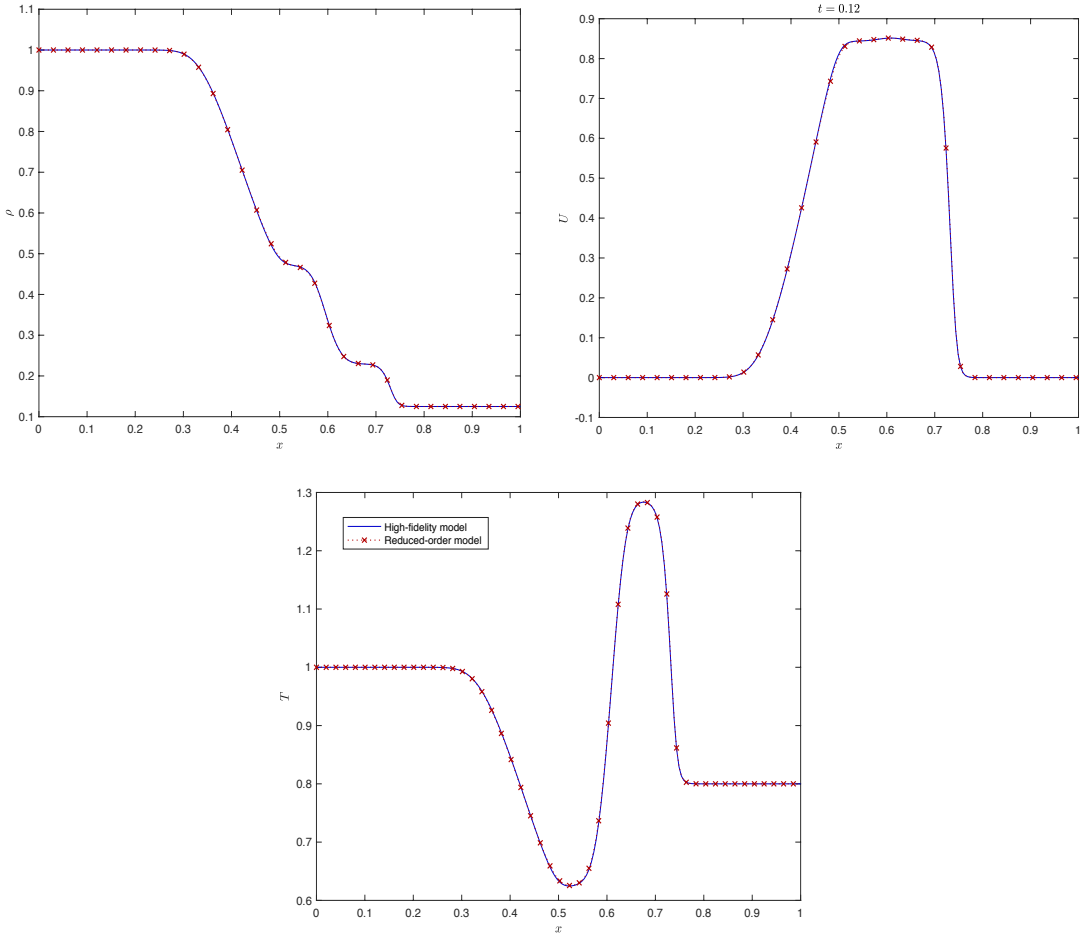


Figure II.4: Density, macroscopic velocity and temperature of the gas at final time for the reproduction of a shock wave with $N_{pod} = 9$.

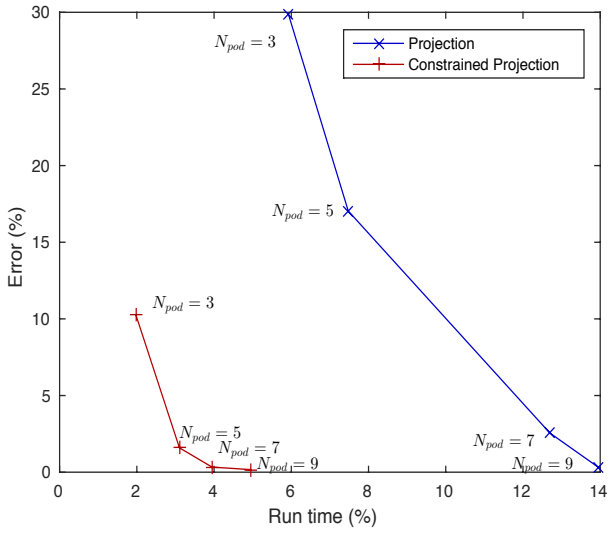


Figure II.5: Performance of the ROM depending on the definition of the approximate Maxwellian distribution function.

The constrained projection-based approach is more accurate than the projection-based approach because the approximate Maxwellian distribution function conserves the mass, momentum and energy of the gas. Moreover, the constrained projection-based approach is also more efficient in terms of computational cost since $\widetilde{M_{f_h}}$ is given by the explicit formula (II.22), whereas in the projection-based approach, we have to solve a nonlinear system (II.9). More specifically, in the projection method-based approach, the Maxwellian distribution function M_f is evaluated to initialize the Newton-Raphson method (Section II.2.2.1). Then, this nonlinear system (II.9) is solved to obtain the discrete Maxwellian distribution function, which is finally projected onto the basis functions. In the constrained projection-based approach, the Maxwellian distribution function M_f is evaluated and projected directly from equation (II.22) to obtain $\widetilde{M_{f_h}}$. For these reasons, the approximate Maxwellian distribution function will be determined by the constrained projection-based approach in the following.

II.4.2 Reproduction of two boundary layers

The second application concerns the choice of the snapshots. Originally, the database $\mathbf{S}$ contains snapshots of the discrete density distribution function because we want the basis functions to be the best representation of f_h. In Section II.3.3, the Maxwellian distribution function is also represented by the basis functions. For this reason, we evaluate the benefit of enriching the database with snapshots of the discrete Maxwellian distribution function.

We consider the reproduction of a flow between two walls (diffuse reflection) placed at $x = 0$ and $x = 1$ with different temperatures at $Kn = 10^{-2}$. The physical space $\Omega_{\mathbf{x}} =]0, 1[$ is discretized using $N_{\mathbf{x}} = 100$ cells, and the velocity

space $\Omega_{\boldsymbol{\xi}} =]-20, 20[$ is discretized using $N_{\boldsymbol{\xi}} = 100$ points. The final time is $t_{max} = 13.03$ and the CFL number is 0.1. The initial and boundary conditions are

$$\begin{cases} \rho_0(x) = 1,\ u_0(x) = 0,\ T_0(x) = 1 & \text{for } x \in \Omega_{\mathbf{x}} \\ u_{bc}(0,t) = 0,\ T_{bc}(0,t) = 0.5,\ u_{bc}(1,t) = 0,\ T_{bc}(1,t) = 1.5 & \text{for } t \in]0, t_{max}]. \end{cases}$$

For the construction of the basis functions, we consider two methods. In the first one, the database $\mathbf{S}_\phi$ (resp. $\mathbf{S}_\psi$) contains snapshots of ϕ_h (resp. ψ_h) taken at each point in space and every 0.4 time units. In the second one, the database $\mathbf{S}_\phi$ (resp. $\mathbf{S}_\psi$) contains snapshots of ϕ_h and M_{ϕ_h} (resp. ψ_h and M_{ψ_h}) taken at each point in space and every 0.4 time units. Figure II.6 shows the squared singular values of $\widetilde{\mathbf{S}}_\phi$ and $\widetilde{\mathbf{S}}_\psi$ for the two methods.

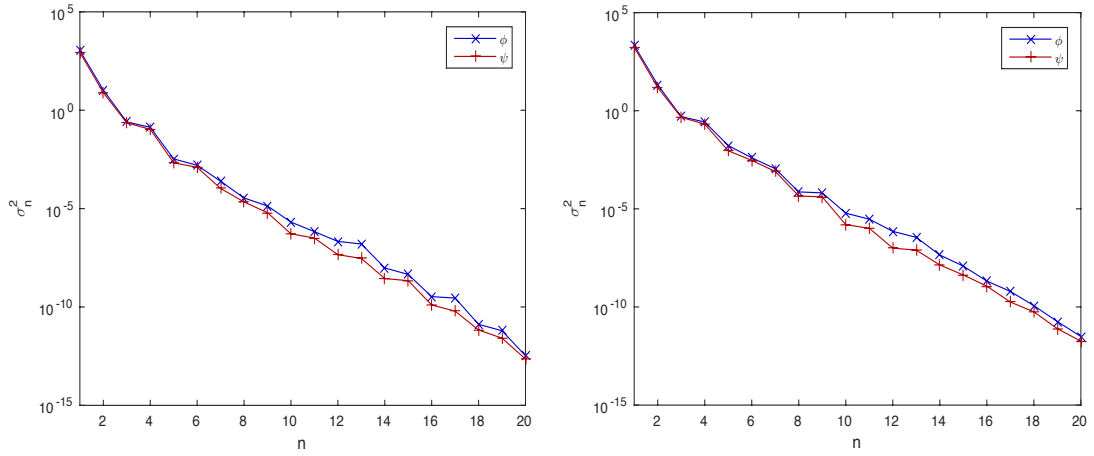


Figure II.6: Squared singular values of $\widetilde{\mathbf{S}}_\phi$ and $\widetilde{\mathbf{S}}_\psi$ without (on the left) and with (on the right) Maxwellian snapshots for the boundary layers reproduction.

The decay of the squared singular values is fast, and 4 basis functions (i.e. 4% of the complete basis) are sufficient to obtain a relative squared projection error of less than 0.01%. In Figure II.7, we plot the macroscopic quantities of interest of the gas at final time obtained by the ROM containing snapshots of the density and Maxwellian distribution functions in the database.

In Figure II.8, we compare the performance of the two approaches to construct the ROM depending on the number of basis functions $N_{pod} = N_{pod}^{\phi} = N_{pod}^{\psi}$.

The enrichment of the snapshot database with the discrete Maxwellian distribution function reduces the approximation error because the Maxwellian distribution function is better represented. Moreover, the run time is almost the same for the two methods. The run time is not exactly the same because the time-step sizes Δt are determined by the CFL condition (II.24), which is slightly different since the basis function are not the same. In the following, the database will contain snapshots of the density and discrete Maxwellian distribution functions in order to improve ROM accuracy.

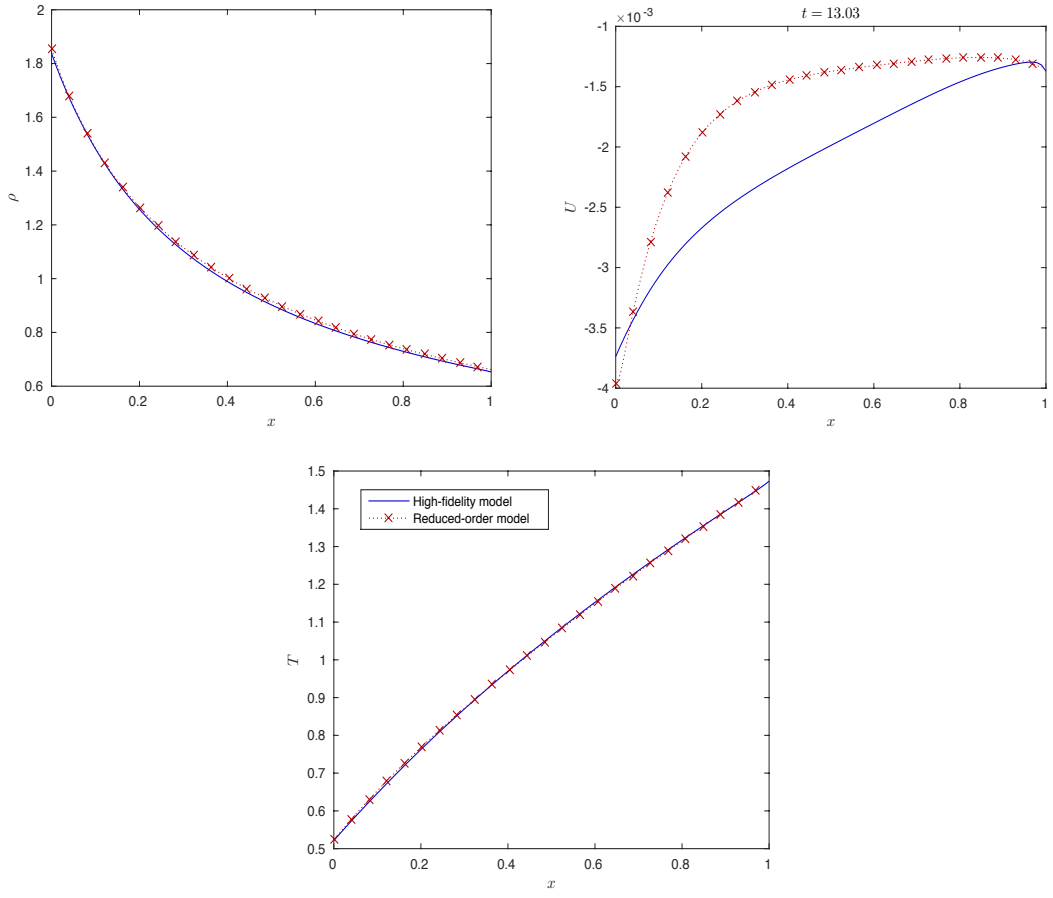


Figure II.7: Density, macroscopic velocity and temperature of the gas at final time for the reproduction of two boundary layers with $N_{pod} = 12$.

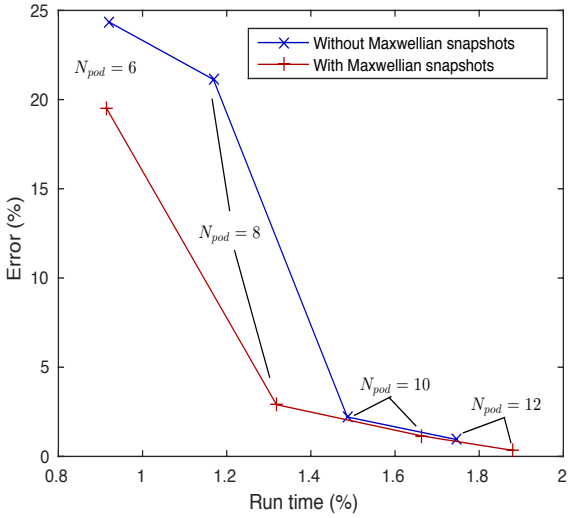


Figure II.8: Performance of the ROM depending on the choice of the database **S**.

II.4.3 Reproduction of a vortex

The third application evaluates the ROM in 2D. We consider the reproduction of a flow past a vertical plate [24] at $Kn = \{0.0345, 0.0689, 0.115, 0.23\}$. The physical space $\Omega_{\mathbf{x}} =]-1.33, 2[\times]0, 3.33[$ is discretized using $N_{\mathbf{x}} = 64^2$ cells, and the velocity space $\Omega_{\boldsymbol{\xi}} =]-10, 10[^2$ is discretized using $N_{\boldsymbol{\xi}} = 41^2$ points. The final time is $t_{max} = 5.3332$ and the CFL number is 0.5. The initial condition is a uniform flow at Mach 0.68

$$\forall \mathbf{x} \in \Omega_{\mathbf{x}} : \ \rho_0(\mathbf{x}) = 1, \ u_0(\mathbf{x}) = 0.68, \ v_0(\mathbf{x}) = 0, \ T_0(\mathbf{x}) = 1.$$

An inflow is imposed at the boundary ($x = -1.33$, $x = 2$ and $y = 3.33$) and is set to be a uniform flow at Mach 0.68. Moreover, a specular reflection is applied at the wall $\mathbf{x} = \{0\} \times]0, 1[$ and at the boundary $y = 0$. The basis functions Φ_n^ϕ (resp. Φ_n^ψ) are constructed from the database $\mathbf{S}_\phi$ (resp. $\mathbf{S}_\psi$) containing snapshots of ϕ_h and M_{ϕ_h} (resp. ψ_h and M_{ψ_h}) taken at each point in space and every 0.2665 time units. The Figure II.9 shows the squared singular values of $\widetilde{\mathbf{S}}_\phi$ and $\widetilde{\mathbf{S}}_\psi$ at $Kn = 0.0345$.

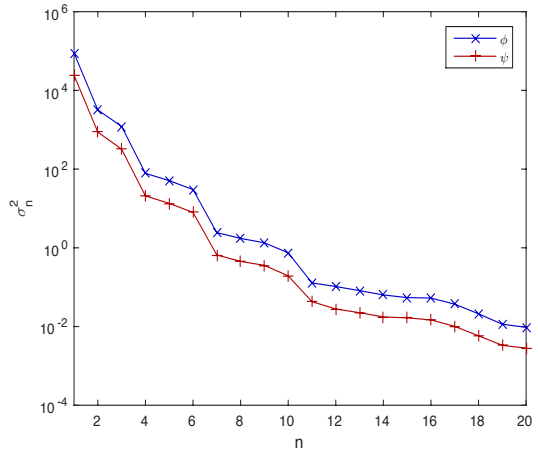
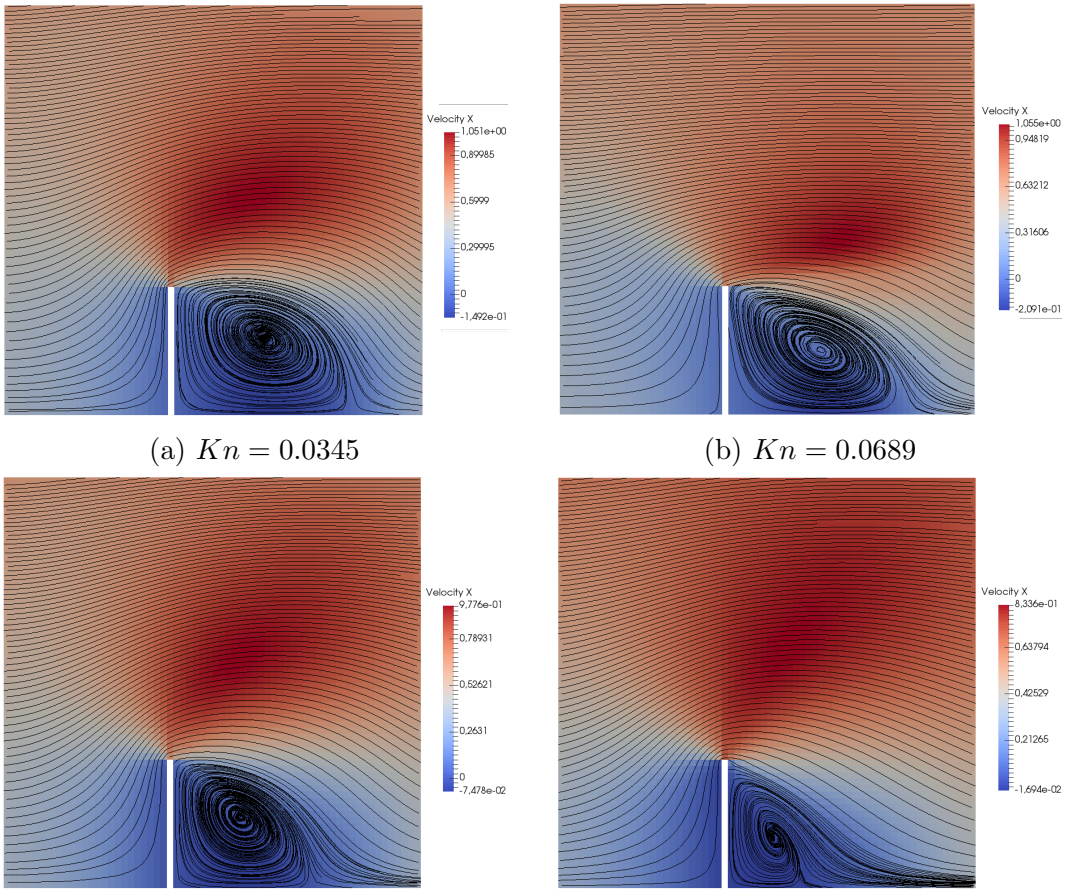


Figure II.9: Squared singular values of $\widetilde{\mathbf{S}}_\phi$ (blue) and $\widetilde{\mathbf{S}}_\psi$ (red) for the vortex reproduction.

The decay of the squared singular values is fast, and 6 basis functions (i.e. 0.4% of the complete basis) are sufficient to obtain a relative squared projection error below 0.01%. In Figure II.10, we plot the streamlines of the macroscopic velocity of the gas at final time obtained by the ROM for different Knudsen numbers. According to the high-fidelity simulations, a vortex is formed at the back of the wall, and the vortex becomes stronger when the Knudsen number decreases. In Figure II.11, we evaluate the performance of the ROM as a function of the number of basis functions $N_{pod} = N_{pod}^\phi = N_{pod}^\psi$.



(a) $Kn = 0.0345$ (b) $Kn = 0.0689$

(c) $Kn = 0.115$ (d) $Kn = 0.23$

Figure II.10: Streamlines of $\mathbf{u}$ at final time for the vortex reproduction with $N_{pod} = 20$.

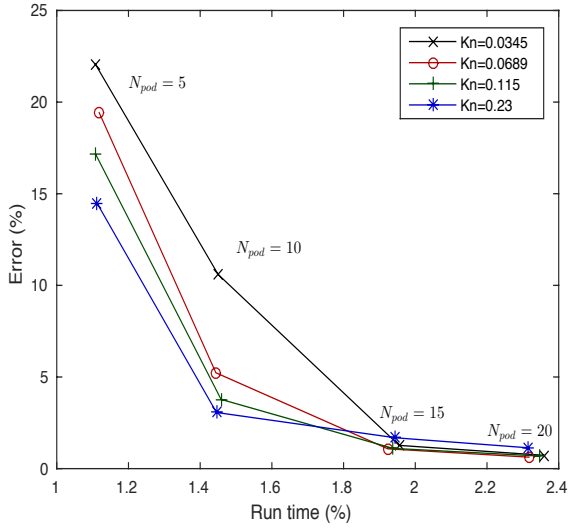


Figure II.11: Performance of the ROM for the reproduction of a vortex.

When the number of basis functions N_{pod} increases, the approximation error decreases and the run time increases. The approximate density distribution function becomes more accurate because the trial subspace spanned by the basis functions converges to the solution manifold. The computational cost increases because we solve more equations in system (II.18) and because the time-step size Δt decreases. The number N_{pod} of basis functions therefore represents a trade-off between accuracy and computational cost. With $N_{pod} = 20$ basis functions, the approximation error is less than 1% and the run time is divided on average by about 45 with respect to the HDM.

II.4.4 Prediction of a vortex

The previous applications were reproduction tests, i.e. the density distribution function that we approximated was included in the snapshot database used to construct the basis functions. We now consider the prediction of a 2D flow past a vertical plate at $Kn = 0.0345$. The input parameter is the free-stream Mach number $\mu \in [0.23, 0.63]$. As in the previous application (Section II.4.3), the physical space $\Omega_{\mathbf{x}} =]-1.33, 2[\times]0, 3.33[$ is discretized using $N_{\mathbf{x}} = 64^2$ cells, and the velocity space $\Omega_{\boldsymbol{\xi}} =]-10, 10[^2$ is discretized using $N_{\boldsymbol{\xi}} = 41^2$ points. The final time is $t_{max} = 5.3332$ and the CFL number is 0.5. The initial condition is a uniform flow at Mach μ

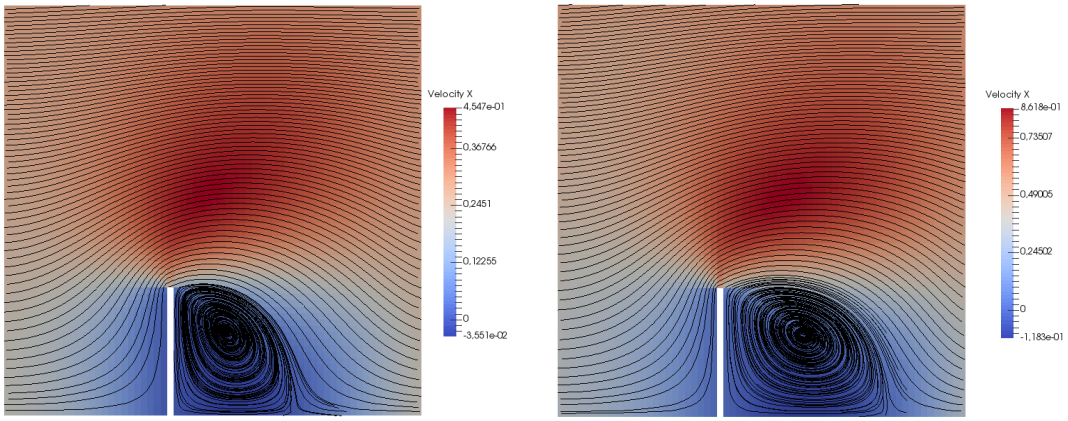
$$\forall \mathbf{x} \in \Omega_{\mathbf{x}} : \ \rho_0(\mathbf{x}; \mu) = 1, \ u_0(\mathbf{x}; \mu) = \mu, \ v_0(\mathbf{x}; \mu) = 0, \ T_0(\mathbf{x}; \mu) = 1.$$

An inflow is imposed at the boundary ($x = -1.33$, $x = 2$ and $y = 3.33$) and is set to be a uniform flow at Mach μ. Moreover, a specular reflection is applied at the wall $\mathbf{x} = \{0\} \times]0, 1[$ and at the boundary $y = 0$.

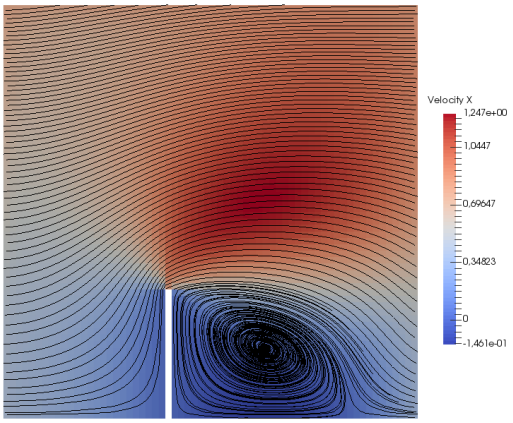
The snapshot database $\mathbf{S}_\phi$ (resp. $\mathbf{S}_\psi$) contains snapshots of ϕ_h and M_{ϕ_h} (resp. ψ_h and M_{ψ_h}) taken at each point in space, every 0.2665 time units and at training input parameter $\mu = 0.63$. In this way, the database contains all the information required to predict flows corresponding to $\mu \in [0.23, 0.63]$.

In Figure II.12, we plot the streamlines of the macroscopic velocity of the gas at final time obtained by the ROM for different free-stream Mach numbers $\mu \in \{0.23, 0.43, 0.63\}$.

In Figure II.13, we evaluate the performance of the ROM for different predictive input parameters $\mu \in \{0.23, 0.33, 0.43, 0.53, 0.63\}$. For $\mu \in [0.23, 0.63]$, the ROM is able to represent the density distribution function even if this one is not in the snapshot database used to construct the basis functions. Moreover, when the number of basis functions $N_{pod} = N_{pod}^{\phi} = N_{pod}^{\psi}$ increases, the ROM becomes more accurate, and with $N_{pod} = 20$ basis functions, the error is less than 1% for all prediction tests.



(a) $\mu = 0.23$ (b) $\mu = 0.43$



(c) $\mu = 0.63$

Figure II.12: Streamlines of $\mathbf{u}$ at final time for the vortex prediction with $N_{pod} = 20$.

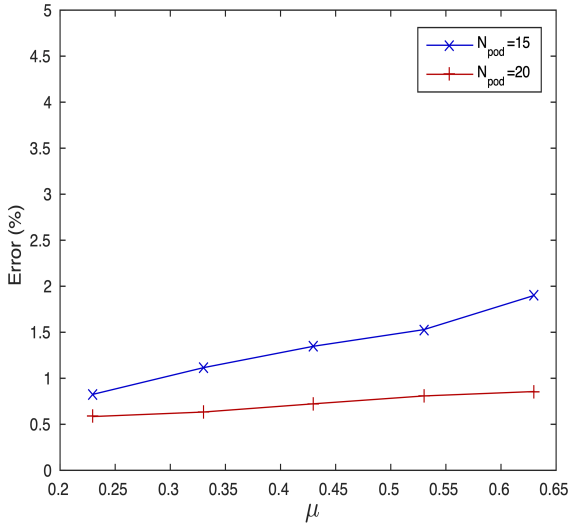


Figure II.13: Accuracy of the ROM for the prediction of a vortex

II.5 Conclusion

In this work, we have presented a new reduced-order approximation of the BGK equation for the simulation of gas flows in both hydrodynamic and rarefied regimes. In this approach, the density distribution function f is represented in velocity space by a small number of basis functions in order to considerably reduce the computational cost associated with these simulations. The basis functions are constructed in the training stage by POD, and the approximate density distribution function is determined at low cost during the prediction stage by the Galerkin method.

In the training stage, we have proposed to collect snapshots of f_h and M_{f_h} since the discrete Maxwellian distribution function is also approximated by the basis functions. The POD is then performed by the classical method due to the large number of snapshots provided by the sampling of the solution manifold.

In the prediction stage, we have shown that the system obtained by the Galerkin method is hyperbolic by construction. Moreover, this system has been modified to preserve the conservation of mass, momentum and energy of the gas. The ROM is finally solved by the finite volume method in space and an IMEX Runge-Kutta scheme in time. Notably, the CFL condition derived from the numerical schemes ensures a stable ROM in 1D and leads to larger time-step sizes than those used in the HDM, allowing to further reduce the computational cost of the ROM by decreasing the number of time-steps.

The performance of the resulting ROM has been evaluated for the simulation of gas flows in both hydrodynamic and rarefied regimes. We have validated the proposed modifications to improve the ROM on the reproduction of a shock wave and boundary layers in 1D. Then, we have investigated the reproduction and prediction of unsteady flows containing vortices in 2D. The results demonstrate the accuracy of the ROM (with less than 1% error) over a range of predictive input parameters and the significant computational speedup factor (approximately 45) delivered by the ROM over the HDM simulation.

Chapter III

Optimal transportation for model order reduction

III.1 Introduction

In Chapter II, we have developed a ROM for the simulation of gas flows in both hydrodynamic and rarefied regimes. In this model, the solution is approximated in velocity space by a small number of basis functions in order to reduce the computational complexity of the simulations. These basis functions are constructed by Proper Orthogonal Decomposition from previously collected solution snapshots. However, the number of high-fidelity simulations to explore the parameter space $\mathcal{D}$ is limited due to the expensive computational cost of the HDM. Notably if the sampling fails to correctly learn the solution manifold, then the training snapshots may be too different from the new predicted solution, and the ROM may lead to unreliable predictions. In addition, the accuracy of the ROM also depends on the snapshot database resulting from the sampling of the solution manifold. Since the snapshots of the high-fidelity solution are collected at different physical points $\mathbf{x}$, time instances t and input parameters $\boldsymbol{\mu}$, the sampling provides a large number of snapshots characterized by different physical regimes and moving features. In particular, due to advection-dominated phenomena, the dimensionality reduction of the resulting snapshot database may be limited, and the number of basis functions required to accurately approximate all these snapshots could be large, as illustrated in Section I.4.4.2.

For these reasons, we propose to modify the snapshot database resulting from the sampling of the solution manifold in order to improve the accuracy and reliability of the ROM developed in Chapter II. These improvements are based on the optimal transport problem, which provides powerful tools to analyze and manipulate the snapshots of the distribution functions (f_h and M_{f_h}). To illustrate this problem, consider a pile of sand that must be displaced to fill up a hole and a cost of transporting one unit of mass from one place to another. The optimal transport problem [80, 66, 27, 53, 3, 112, 97] is to find the optimal way to transport the

pile of sand to fill up the hole while minimizing the total transport cost. When the transport cost is associated with the L^2-ground cost, the square root of the minimal total transport cost corresponds to the L^2-Wasserstein distance, which defines a robust metric to quantify the notion of proximity between two distribution functions. Compared to the classical L^2-norm which corresponds to the pointwise difference of the two distributions, the Wasserstein distance measures the minimal effort to push forward one distribution onto the other. In addition, this distance can also be used to define geodesic paths between distribution functions. In particular, Wasserstein barycenters on these geodesics give rise to realistic interpolations that preserve the features of the interpolated distribution functions.

In this work, we propose two applications of the optimal transport problem for model order reduction [64, 22]. In the first application, the sampling of the solution manifold is completed with additional snapshots [82, 113, 22] generated by optimal transport in order to improve the reliability of the ROM. To this end, the new artificial snapshots are defined as the Wasserstein barycenters of the high-fidelity snapshots, enabling a fast enrichment of the snapshot database without employing the computationally expensive HDM. In the second application, the Wasserstein distance is combined with a cluster analysis method to partition the large snapshot database [30, 5, 65]. Instead of approximating the solution with the same reduced basis in all the physical domain $\Omega_{\mathbf{x}}$, different local reduced bases are used to improve the ROM accuracy. The objectif of this clustering is to identify regions where the behaviour of the solution is similar to decompose the physical domain. The solution is then approximated in each subdomain by a local reduced basis, which is more accurate than the global reduced basis to represent the corresponding snapshot cluster.

This work is organized as follows. In Section III.2, we introduce the optimal transport problem and its numerical resolution. Then, two applications of the optimal transport problem for model reduction are presented. In Section III.3, the high-fidelity snapshots are interpolated in velocity space by optimal transport to enrich the snapshot database with new artificial snapshots. In the second application, a clustering analysis algorithm combined with the Wasserstein distance is employed to partition automatically the physical domain from the snapshot database, as described in Section III.4.

III.2 Optimal transport

The optimal transport problem was introduced by Monge [80] and then developed by Kantorovich [66]. Given two non-negative functions f_1, f_2 and a cost $c(\mathbf{x}, \mathbf{y})$ of transporting one unit of mass from $\mathbf{x}$ to $\mathbf{y}$, the optimal transport problem [27, 53, 3, 112, 97] is to find the optimal way to transport f_1 to f_2 while minimizing the total transport cost. Even though this problem is difficult to solve, special cases have simple characterizations of the solution. In particular, the one-

dimensional case and the optimal transport problem for normal distributions have useful applications. In the general case, many approaches [16, 84, 17, 85] have been developed to solve the optimal transport problem. However, these methods are computationally prohibitive for large-scale problems. For this reason, we consider an approach based on an entropic-regularization of the optimal transport problem [105], which enables fast computations of the solution.

III.2.1 Optimal transport problem

The optimal transport problem was first introduced by Monge [80]. In this formulation, the problem is to find the transport map $\mathbf{M}$ minimizing the total transport cost. However, the mass is mapped and cannot be split, leading to difficulties concerning the existence of valid transport maps. For this reason, Kantorovich developed a natural relaxation of the optimal transport problem [66] allowing mass to be split. In this formulation, the problem consists in finding the transport plan π minimizing the total transport cost. In particular, when the cost function $c(\mathbf{x}, \mathbf{y})$ is the L^p-ground cost, the minimum of the total transport cost corresponds to the L^p-Wasserstein distance to the p-th power. Notably, this metric offers a relevant way to compare density distribution functions by measuring the cost of transporting their features. In the same way, the optimal transport framework also provides natural interpolations of distribution functions, as illustrated in Figures III.2 and III.3.

III.2.1.1 Monge-Kantorovich formulation

Let $f_1, f_2 : \mathbb{R}^d \to \mathbb{R}_+$ be two non-negative functions with bounded supports in $\mathbb{R}^d$ ($d \in \mathbb{N}^*$). Since f_1 and f_2 are only transported, they must have the same total mass and in the following, we assume without loss of generality that f_1 and f_2 are probability density functions with total mass one:

$$\int_{\mathbb{R}^d} f_1(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{\mathbb{R}^d} f_2(\mathbf{y})\,\mathrm{d}\mathbf{y} = 1.$$

Moreover, the cost function $c(\mathbf{x}, \mathbf{y}) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ represents the cost of transporting one unit of mass from $\mathbf{x}$ to $\mathbf{y}$ in the following.

Monge formulation. The original problem is to find the optimal transport map $\mathbf{M} : \mathbb{R}^d \to \mathbb{R}^d$ minimizing the total transport cost:

$$\min_{\mathbf{M} \in \Gamma(f_1, f_2)} \int_{\mathbb{R}^d} c(\mathbf{x}, \mathbf{M}(\mathbf{x}))\, f_1(\mathbf{x})\,\mathrm{d}\mathbf{x},$$

where $\mathbf{M}$ denotes the map transporting $f_1(\mathbf{x})$ to $f_2(\mathbf{M}(\mathbf{x}))$. A valid transport map $\mathbf{M}$ that pushes forward f_1 onto f_2 satisfies for all bounded subset $\Omega \subset \mathbb{R}^d$:

$$\int_{\mathbf{M}^{-1}(\Omega)} f_1(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_{\Omega} f_2(\mathbf{y})\,\mathrm{d}\mathbf{y}, \tag{III.1}$$

and $\Gamma(f_1, f_2)$ denotes the set of valid transport map verifying (III.1). Moreover, if $\mathbf{M}$ is a smooth one-to-one map, then this condition (III.1) is equivalent to

$$f_1(\mathbf{x}) = f_2(\mathbf{M}(\mathbf{x}))|\det(\nabla \mathbf{M}(\mathbf{x}))|$$

by using the change of variables $\mathbf{M}(\mathbf{x}) = \mathbf{y}$.

Kantorovitch formulation. In the Monge formulation, the mass is mapped and cannot be split. In particular, this causes difficulties concerning the existence of valid transport maps (e.g. consider $f_1 = \delta_0$ and $f_2 = \frac{1}{2}\delta_{-1} + \frac{1}{2}\delta_1$). For this reason, Kantorovich formulated a natural relaxation of the optimal transport problem allowing mass to be split. This problem is to find the optimal transport plan $\pi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ minimizing the total transport cost:

$$\min_{\pi \in \Pi(f_1, f_2)} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} c(\mathbf{x}, \mathbf{y}) \pi(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{y}, \tag{III.2}$$

where $\pi(\mathbf{x}, \mathbf{y})$ denotes the among of mass transferred from $\mathbf{x}$ to $\mathbf{y}$. A valid transport plan π conserves the mass moved from $f_1(\mathbf{x})$ and to $f_2(\mathbf{y})$:

$$f_1(\mathbf{x}) = \int_{\mathbb{R}^d} \pi(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{y} \quad \text{and} \quad f_2(\mathbf{y}) = \int_{\mathbb{R}^d} \pi(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{x}, \tag{III.3}$$

and $\Pi(f_1, f_2)$ denotes the set of valid transport plan verifying (III.3). This formulation leads to a linear programming problem since the objective function (III.2) and the constraints (III.3) are linear with respect to the transport plan π. However, the computational complexity of this approach is prohibitive for large-scale problem due to the quadratic number of unknowns.

III.2.1.2 Wasserstein distance

The optimal transport framework offers a relevant way to measure distances between pairs of probability density functions. Let the Euclidian space $\mathbb{R}^d$ endowed with the L^p-norm $\|\cdot\|_p$. The L^p-Wasserstein distance $\mathcal{W}_p(f_1, f_2)$ between two probability density functions $f_1, f_2 : \mathbb{R}^d \to [0, 1]$ with bounded p-th moment is defined by

$$\mathcal{W}_p(f_1, f_2) \coloneqq \inf_{\pi \in \Pi(f_1, f_2)} \left(\int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \|\mathbf{y} - \mathbf{x}\|_p^p \; \pi(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{y} \right)^{1/p}. \tag{III.4}$$

In particular, $\mathcal{W}_p(f_1, f_2)$ corresponds to the p-th root of the minimal total transport cost (III.2) associated with the cost function $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - \mathbf{x}\|_p^p$. In the following, we will focus on the L^2-Wasserstein distance since there exists a unique solution [28, 111] to the optimal transport problem associated with the quadratic cost function $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - \mathbf{x}\|_2^2$.

In addition, the Wasserstein distance also provides natural interpolation of probability density functions. Given K probability density functions f_k, the L^2-Wasserstein barycenter of $(f_1, \dots, f_K)$ at barycentric coordinates $(\lambda_1, \dots, \lambda_K)$ is defined as the Féchet mean associated with the L^2-Wasserstein distance:

$$\begin{cases} \underset{f}{\text{minimize}} & \displaystyle\sum_{k=1}^{K} \lambda_k \mathcal{W}_2(f_k, f)^2 \\ \text{subject to} & \displaystyle\sum_{k=1}^{K} \lambda_k = 1 \\ & \lambda_k \geqslant 0 \qquad \forall k \in \{1, \dots, K\}. \end{cases}$$

Since the Wasserstein distance measures the cost of transporting the features of the probability density functions, an interpolation based on this distance also takes into account the transport of these features. Compared to the L^2-norm which leads to a pointwise interpolation, the Wasserstein barycenter interpolates the position of the features, as illustrated in Figures III.2 and III.3.

III.2.2 Special cases

Even though the optimal transport problem is difficult to solve in general, the one-dimensional case and the optimal transport problem for normal distribution have simple characterizations of the solution. In particular, this last special case has useful applications since the Maxwellian distribution function $M_f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu})$ follows a normal distribution with mean $\mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu})$ and variance $T(\mathbf{x}, t; \boldsymbol{\mu})$ in velocity space after normalization, i.e. $\rho(\mathbf{x}, t; \boldsymbol{\mu}) = 1$. Furthermore, the density distribution functions $f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu})$ also tends to a Maxwellian distribution ($f \to M_f$) in the hydrodynamic limit ($Kn \to 0$) according to Section II.2.1.1.

III.2.2.1 Optimal transport in 1D

Let $F_1, F_2 : \mathbb{R} \to [0, 1]$ be the cumulative distribution functions of f_1 and f_2, respectively,

$$F(x) = \int_{-\infty}^{x} f(s)\, \mathrm{d}s, \tag{III.5}$$

where F is right-continuous, non-decreasing, $F(-\infty) = 0$ and $F(+\infty) = 1$. Moreover, let F^{-1} be the generalized inverse of F defined by

$$F^{-1}(s) = \inf\{x \in \mathbb{R}, \text{ such that } F(x) > s\}. \tag{III.6}$$

Note that this definition is not unique since we could also consider $F^{-1}(s) = \sup\{x \in \mathbb{R}, \text{ such that } F(x) < s\}$ for example. In 1D, the L^2-Wasserstein distance between f_1 and f_2 corresponds to the L^1-distance between the cumulative distribution functions:

$$\mathcal{W}_2(f_1, f_2)^2 = \int_0^1 |F_1^{-1}(s) - F_2^{-1}(s)|\, \mathrm{d}s = \int_{\mathbb{R}} |F_1(x) - F_2(x)|\, \mathrm{d}x,$$

as illustrated in Figure III.1. Moreover, the L^2-Wasserstein barycenter f_λ^* of $\{f_k\}_{k=1}^K$ at barycentric coordinates $\lambda = (\lambda_1, \dots, \lambda_K)$, defined by

$$f_\lambda^* = \arg\min_f \sum_{k=1}^K \lambda_k \mathcal{W}_2(f_k, f)^2$$

where $\sum_{k=1}^K \lambda_k = 1$ and $\lambda_k \geqslant 0$ for $k \in \{1, \dots, K\}$, verifies

$$(F_\lambda^*)^{-1}(s) = \sum_{l=1}^K \lambda_l F_l^{-1}(s).$$

By inverting equations (III.6) and (III.5), the Wasserstein barycenter f_λ^* is then recovered from $(F_\lambda^*)^{-1}$, as illustrated in Figure III.1.

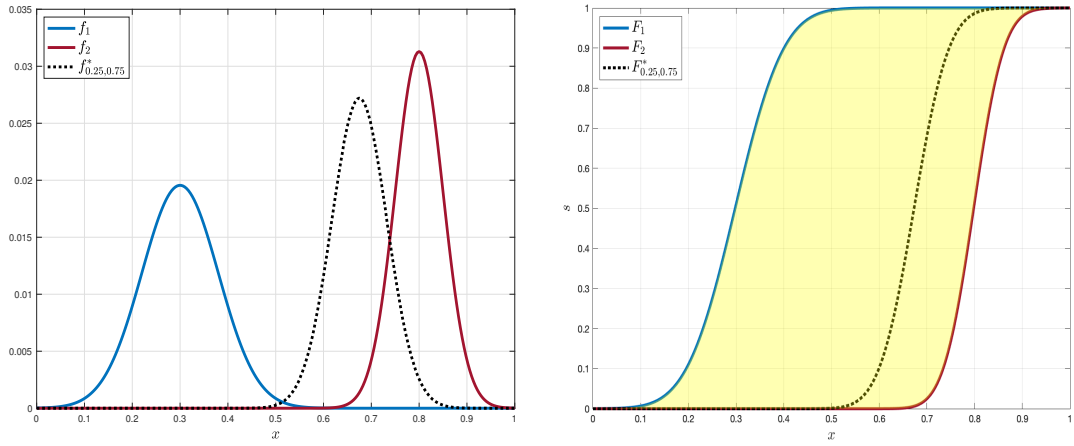


Figure III.1: Illustration of the one-dimensional optimal transport problem. The yellow surface represents the L^2-Wasserstein distance, and the dotted black lines illustrate the construction of L^2-Wasserstein barycenters.

III.2.2.2 Optimal transport for normal distributions

Let f_1, f_2 be two density probability functions following a normal distribution with respective means $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{R}^d$ and symmetric positive semi-definite covariance matrices $\mathbf{\Sigma}_1, \mathbf{\Sigma}_2 \in \mathbb{R}^{d \times d}$. The L^2-Wasserstein distance between f_1 and f_2 is

$$\mathcal{W}_2(f_1, f_2)^2 = \|\mathbf{u}_1 - \mathbf{u}_2\|_2^2 + \mathrm{Tr}(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2 - 2(\mathbf{\Sigma}_1^{\frac{1}{2}} \mathbf{\Sigma}_2 \mathbf{\Sigma}_1^{\frac{1}{2}})^{\frac{1}{2}}),$$

where $\mathbf{\Sigma} = \mathbf{\Sigma}^{\frac{1}{2}} (\mathbf{\Sigma}^{\frac{1}{2}})^T$ denotes the Cholesky decomposition (Definition 1) of $\mathbf{\Sigma}$. Moreover, the L^2-Wasserstein barycenter f_λ^* of $\{(f_k, \lambda_k)\}_{k=1}^K$, defined by

$$f_\lambda^* = \arg\min_f \sum_{k=1}^K \lambda_k \mathcal{W}_2(f_k, f)^2$$

where $\sum_{k=1}^{K} \lambda_k = 1$ and $\lambda_k \geqslant 0$ for $k \in \{1, \dots, K\}$, is the normal distribution with mean $\sum_{k=1}^{K} \lambda_k \mathbf{u}_k$ and covariance matrix $\mathbf{\Sigma}$ verifying

$$\mathbf{\Sigma} = \sum_{k=1}^{K} \lambda_k (\mathbf{\Sigma}^{\frac{1}{2}} \mathbf{\Sigma}_k \mathbf{\Sigma}^{\frac{1}{2}})^{\frac{1}{2}}. \tag{III.7}$$

This equation (III.7) is solved in practice by the fixed point iteration

$$\mathbf{\Sigma}_{\text{next}} = \mathbf{\Sigma}_{\text{old}}^{-\frac{1}{2}} \left(\sum_{k=1}^{K} \lambda_k (\mathbf{\Sigma}_{\text{old}}^{\frac{1}{2}} \mathbf{\Sigma}_k \mathbf{\Sigma}_{\text{old}}^{\frac{1}{2}})^{\frac{1}{2}} \right)^2 \mathbf{\Sigma}_{\text{old}}^{-\frac{1}{2}}.$$

However, for $K = 2$, the solution to equation (III.7) is simply given by

$$\mathbf{\Sigma} = \mathbf{\Sigma}_1^{-\frac{1}{2}} (\lambda_1 \mathbf{\Sigma}_1 + \lambda_2 (\mathbf{\Sigma}_1^{\frac{1}{2}} \mathbf{\Sigma}_2 \mathbf{\Sigma}_1^{\frac{1}{2}})^{\frac{1}{2}})^2 \mathbf{\Sigma}_1^{-\frac{1}{2}}.$$

Moreover, if the covariance matrix is a diagonal matrix, then

$$\sqrt{\mathbf{\Sigma}_{n,n}} = \sum_{k=1}^{K} \lambda_k \sqrt{(\mathbf{\Sigma}_k)_{n,n}}.$$

Notably in the case of Maxwellian distribution functions, the covariance matrix is the identity matrix times the temperature $T \in \mathbb{R}_+$, which leads to

$$\sqrt{T} = \sum_{k=1}^{K} \lambda_k \sqrt{T_k}.$$

A comparaison of the resulting barycenters computed from the L^2-norm and the L^2-Wasserstein distance is presented in Figure III.2.

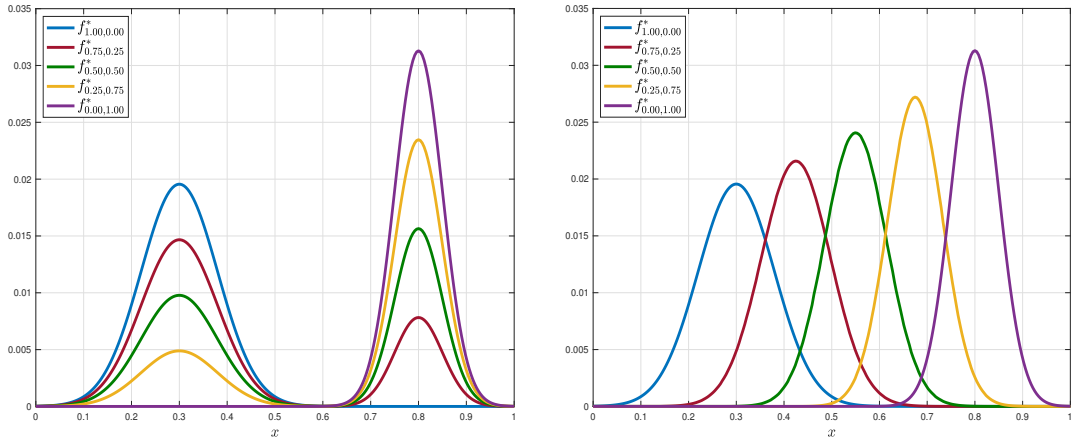


Figure III.2: Comparison of 5 barycenters $f^*_{\lambda_1,\lambda_2}$ of (f_1, f_2) (plotted in Figure III.1) at barycentric coordinates (λ_1, λ_2) computed from the L^2-norm (left) and the L^2-Wasserstein distance (right).

III.2.3 Entropic-regularization of optimal transportation

In order to solve the optimal transport problem in the general case, many methods have been developed such as, for example, the linear programming-based approach, the resolution of the Monge-Ampère equation [84, 17], the proximal splitting method [85] or the Benamou-Brenier algorithm [16]. However, these approaches are computationally prohibitive for large-scale problems. For this reason, we consider in this work an approach [105] based on an entropic-regularization of the optimal transport problem, enabling fast computations of the solution. Let $\gamma > 0$, the entropy-regularized L^2-Wasserstein distance between f_1 and f_2 is defined by

$$\mathcal{W}_{2,\gamma}(f_1, f_2)^2 \coloneqq \inf_{\pi \in \Pi(f_1, f_2)} \left\{ \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \|\mathbf{x} - \mathbf{y}\|_2^2 \, \pi(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{y} - \gamma H(\pi) \right\},$$

where $H(\pi)$ denotes the entropy

$$H(\pi) = - \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \pi \log(\pi) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{y}.$$

This regularization allows to re-write the optimal transport problem as a projection:

$$\mathcal{W}_{2,\gamma}(f_1, f_2)^2 = \gamma \left(1 + \min_{\pi \in \Pi(f_1, f_2)} \mathrm{KL}(\pi | \mathcal{K}) \right),$$

where the Kullback-Leibler divergence [68] is defined by

$$\mathrm{KL}(\pi | \mathcal{K}) \coloneqq \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} \pi \left(\log \left(\frac{\pi}{\mathcal{K}} \right) - 1 \right) \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{y} \quad \text{with} \quad \mathcal{K}(\mathbf{x}, \mathbf{y}) = \exp \left(- \frac{\|\mathbf{x} - \mathbf{y}\|_2^2}{\gamma} \right).$$

In the ROM developed in Chapter II, the velocity space $\Omega_{\boldsymbol{\xi}}$ is discretized by a uniform cartesian grid. By encoding the distribution functions as the vectors $\mathbf{f}_1, \mathbf{f}_2 \in \mathbb{R}_+^{N_{\boldsymbol{\xi}}}$ and the transportation plan as the matrix $\boldsymbol{\pi} \in \mathbb{R}_+^{N_{\boldsymbol{\xi}} \times N_{\boldsymbol{\xi}}}$, the discrete Kullback-Leibler divergence is defined by

$$\mathrm{KL}(\boldsymbol{\pi} | \mathbf{K}) = (\Delta \boldsymbol{\xi})^2 \sum_{i=1}^{N_{\boldsymbol{\xi}}} \sum_{j=1}^{N_{\boldsymbol{\xi}}} \pi_{i,j} \left(\log \left(\frac{\pi_{i,j}}{K_{i,j}} \right) - 1 \right) \quad \text{with} \quad K_{i,j} = \exp \left(- \frac{\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|_2^2}{\gamma} \right).$$

According to the Sinkhorn's theorem [101], the discrete transportation plan can be written as

$$\boldsymbol{\pi} = \mathrm{diag}(\mathbf{u}) \mathbf{K} \mathrm{diag}(\mathbf{v}),$$

where the vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{N_{\boldsymbol{\xi}}}$ satisfy the mass conservation laws (III.3):

$$\mathbf{u} = \mathbf{f}_1 \oslash (\mathbf{K}(\mathbf{a} \otimes \mathbf{v})) \quad \text{and} \quad \mathbf{v} = \mathbf{f}_2 \oslash (\mathbf{K}(\mathbf{a} \otimes \mathbf{u}))$$

with $\mathbf{a} = (\Delta \boldsymbol{\xi}, \ldots, \Delta \boldsymbol{\xi})^T \in \mathbb{R}^{N_{\boldsymbol{\xi}}}$ and $\otimes$ (resp. $\oslash$) denotes the Hadamard product (resp. division). This discrete problem is solved by the Sinkhorn-Knopp algorithm

[102], where the discrete transport plan $\boldsymbol{\pi}$ is iteratively projected onto the affine constraint sets

$$\Pi_1 = \left\{\boldsymbol{\pi} \in \mathbb{R}_+^{N_{\boldsymbol{\xi}} \times N_{\boldsymbol{\xi}}} : \ \boldsymbol{\pi}\mathbf{a} = \mathbf{f}_1\right\} \quad \text{and} \quad \Pi_2 = \left\{\boldsymbol{\pi} \in \mathbb{R}_+^{N_{\boldsymbol{\xi}} \times N_{\boldsymbol{\xi}}} : \ \boldsymbol{\pi}^T\mathbf{a} = \mathbf{f}_2\right\}.$$

The discrete entropy-regularized L^2-Wasserstein distance is then recovered from $\mathbf{u}$ and $\mathbf{v}$ by

$$\mathcal{W}_{2,\gamma}(\mathbf{f}_1, \mathbf{f}_2)^2 = \gamma \mathbf{a}^T(\mathbf{f}_1 \otimes \log(\mathbf{u}) + \mathbf{f}_2 \otimes \log(\mathbf{v})),$$

as illustrated by Algorithm 1.

Algorithm 1 Entropy-regularized L^2-Wasserstein distance [105]

$\mathbf{v} \leftarrow \mathbf{1}$
repeat
  $\mathbf{u} \leftarrow \mathbf{f}_1 \oslash (\mathbf{K}(\mathbf{a} \otimes \mathbf{v}))$ ▷ Projection onto Π_1
  $\mathbf{v} \leftarrow \mathbf{f}_2 \oslash (\mathbf{K}(\mathbf{a} \otimes \mathbf{u}))$ ▷ Projection onto Π_2
until $\|\mathbf{u} \otimes (\mathbf{K}(\mathbf{a} \otimes \mathbf{v})) - \mathbf{f}_1\|_\infty < \epsilon$
return $\sqrt{\gamma \mathbf{a}^T(\mathbf{f}_1 \otimes \log(\mathbf{u}) + \mathbf{f}_2 \otimes \log(\mathbf{v}))}$

The entropy-regularized optimal transport also enables the computation of Wasserstein barycenters. Given K non-negative vectors $\mathbf{f}_k$, the entropy-regularized L^2-Wasserstein barycenter $\mathbf{f}_\lambda^*$ at barycentric coordinates $\{(\mathbf{f}_k, \lambda_k)\}_{k=1}^K$ is defined by

$$\mathbf{f}_\lambda^* = \arg\min_{\mathbf{f}} \sum_{k=1}^K \lambda_k \mathcal{W}_{2,\gamma}(\mathbf{f}_k, \mathbf{f})^2, \tag{III.8}$$

where $\sum_{k=1}^K \lambda_k = 1$ and $\lambda_k \geqslant 0$ for $k \in \{1, \dots, K\}$. By inserting the discrete transportation plans into equation (III.8), we obtain the minimization problem:

$$\begin{cases} \underset{\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_K}{\text{minimize}} & \displaystyle\sum_{k=1}^K \lambda_k \mathrm{KL}(\boldsymbol{\pi}_k | \mathbf{K}) \\ \text{subject to} & \boldsymbol{\pi}_k^T\mathbf{a} = \mathbf{f}_k \qquad \forall k \in \{1, \dots, K\} \\ & \boldsymbol{\pi}_k\mathbf{a} = \boldsymbol{\pi}_l\mathbf{a} \qquad \forall k, l \in \{1, \dots, K\}, \end{cases}$$

where the first (resp. second) constraint enforces the conservation of mass moved from $\mathbf{f}_k$ (resp. to $\mathbf{f}_\lambda^*$). As previously, this problem is solved by the iterative Bregman projection [26], where the discrete transport plans are iteratively projected onto the affine constraint sets

$$\Pi_1 = \left\{\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_K \in \mathbb{R}_+^{N_{\boldsymbol{\xi}} \times N_{\boldsymbol{\xi}}} : \ \boldsymbol{\pi}_k^T\mathbf{a} = \mathbf{f}_k \quad \text{for } k \in \{1, \dots, K\}\right\},$$
$$\Pi_2 = \left\{\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_K \in \mathbb{R}_+^{N_{\boldsymbol{\xi}} \times N_{\boldsymbol{\xi}}} : \ \boldsymbol{\pi}_k\mathbf{a} = \boldsymbol{\pi}_l\mathbf{a} \quad \text{for } k, l \in \{1, \dots, K\}\right\}.$$

Given $\boldsymbol{\pi}_k = \text{diag}(\mathbf{u}_k)\mathbf{K}\text{diag}(\mathbf{v}_k)$, the entropy-regularized L^2-Wasserstein barycenter is recovered from $\mathbf{u}_k, \mathbf{v}_k \in \mathbb{R}^{N_{\boldsymbol{\xi}}}$ by

$$\mathbf{f}_\lambda^* = \bigotimes_{k=1}^{K} \Big(\mathbf{v}_k \otimes (\mathbf{K}(\mathbf{a} \otimes \mathbf{u}_k)) \Big)^{\lambda_k},$$

as illustrated in Algorithm 2.

Algorithm 2 Entropy-regularized L^2-Wasserstein barycenter [105]

$\mathbf{v}_1, \dots, \mathbf{v}_K \leftarrow \mathbf{1}$
repeat
  for $k = 1, \dots, K$ **do**
    $\mathbf{u}_k \leftarrow \mathbf{f}_k \oslash (\mathbf{K}(\mathbf{a} \otimes \mathbf{v}_k))$ ▷ Projection onto Π_1
  end for
  $\mathbf{f}_\lambda^* \leftarrow \bigotimes_{k=1}^{K} \Big(\mathbf{v}_k \otimes (\mathbf{K}(\mathbf{a} \otimes \mathbf{u}_k)) \Big)^{\lambda_k}$ ▷ Wasserstein barycenter
  $\mathbf{f}_\lambda^* \leftarrow \text{Entropic-sharpening}(\mathbf{f}_\lambda^*, H_0)$ ▷ Algorithm 3 (optional)
  for $k = 1, \dots, K$ **do**
    $\mathbf{v}_k \leftarrow \mathbf{f}_\lambda^* \oslash (\mathbf{K}(\mathbf{a} \otimes \mathbf{u}_k))$ ▷ Projection onto Π_2
  end for
until $\max_{1 \leqslant k \leqslant K} \| \mathbf{u}_k \otimes (\mathbf{K}(\mathbf{a} \otimes \mathbf{v}_k)) - \mathbf{f}_k \|_\infty < \epsilon$
return $\mathbf{f}_\lambda^*$

The main drawback of this method is that the entropy-regularized L^2-Wasserstein barycenter may appear too diffuse. To cure this issue, a constraint on the entropy $H(\mathbf{f}_\lambda^*) \leq H_0$ is added in the minimization problem (III.8), as explained in [105]. This modification of the computation of the entropy-regularized L^2-Wasserstein barycenter is described in Algorithm 3.

Algorithm 3 Entropic-sharpening($\mathbf{f}_\lambda^*, H_0$) [105]

if $H_0 < H(\mathbf{f}_\lambda^*)$ **then**
  $\eta \leftarrow \text{find}\,(\eta \in \mathbb{R}_+ : \; H_0 = H((\mathbf{f}_\lambda^*)^\eta))$ ▷ The function "find" is given in [52].
  $\mathbf{f}_\lambda^* \leftarrow (\mathbf{f}_\lambda^*)^\eta$
end if
return $\mathbf{f}_\lambda^*$

The entropy-regularized transportation problem is particularly well suited to the ROM developed in Chapter II. Since the velocity space $\Omega_{\boldsymbol{\xi}}$ is discretized by a uniform cartesian grid, the matrix-vector multiplications $\mathbf{K}(\mathbf{a} \otimes \mathbf{u}_k)$ and $\mathbf{K}(\mathbf{a} \otimes \mathbf{v}_k)$ are replaced by a convolution with a gaussian kernel. Moreover, this kernel is separable and the convolution is written as successive 1D convolutions, leading to fast computations of the optimal transport solution. In addition, the run time

can be further improved by using GPU acceleration [45, 46]. Figure III.3 shows examples of entropy-regularized L^2-Wasserstein barycenters $\mathbf{f}_\lambda^*$ in 2D.

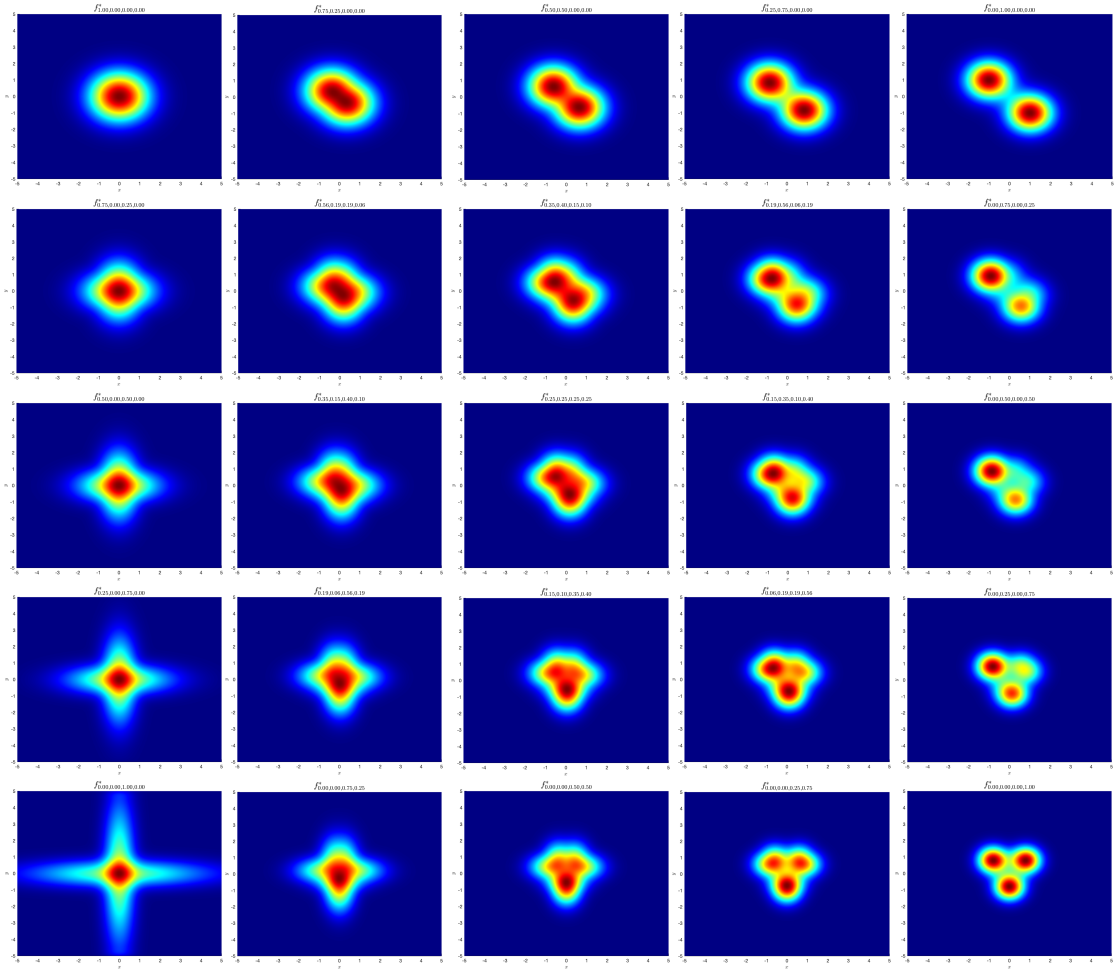


Figure III.3: Examples of 2D entropy-regularized L^2-Wasserstein barycenters $\mathbf{f}_\lambda^*$.

III.3 Application to snapshot database enrichment

The first application of the optimal transport problem concerns the interpolation of distribution functions. In Chapter II, we have presented a ROM for the simulation of gas flows in both hydrodynamic and rarefied regimes. In this model, snapshots of the high-fidelity solution are collected at different points $\mathbf{x}$, time instances t and input parameters $\boldsymbol{\mu}$ in order to learn the solution manifold. Since the trial subspace is then constructed to approximate these snapshots, the reliability of the ROM depends on the sampling of the solution manifold. However, the number of high-dimensional simulations for exploring the parameter space is limited due

to the expensive computational cost of the HDM. In particular, if the training snapshots are too different from the new predicted solutions, the ROM may lead to inaccurate approximations. For example, let $\mathrm{supp}(f_h) \subseteq \Omega_{\boldsymbol{\xi}}$ be the subspace containing at least 99.99% of the distribution function f_h:

$$\int_{\mathrm{supp}(f_h)} f_h(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi} > 99.99\% \int_{\Omega_{\boldsymbol{\xi}}} f_h(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi},$$

since the distribution functions decrease rapidly. If the support of the training snapshots $\mathrm{supp}(s_l)$ and the support of the new predicted distribution function $\mathrm{supp}(f_h)$ are disjoint sets (i.e. $\mathrm{supp}(f_h) \cap (\bigcup_l \mathrm{supp}(s_l)) = \emptyset$), then the basis functions (associated with strictly positive singular values) will be zero on $\mathrm{supp}(f_h)$ and will not be able to represent f_h. For this reason, we propose to complete the sampling of the solution manifold with new artificial snapshots [82, 113, 22]. In this strategy, only snapshots that bring new information are created, enabling a fast enrichment of the snapshot database with respect to the cost of the HDM. In addition, these additional snapshots are computed by optimal transport, which provides natural interpolations of the distribution functions in velocity space without employing the computationally expensive HDM.

III.3.1 Snapshot interpolation via optimal transport

Let $\mathbf{S}^{\mathrm{hf}} \in \mathbb{R}^{N_{\boldsymbol{\xi}} \times K_{\mathrm{hf}}}$ be the database containing K_{hf} high-fidelity snapshots s_l of the distribution functions (f_h and M_{f_h}) collected at point $\mathbf{x}_{i(l),j(l),k(l)}$, time instance $t_{p(l)}$ and input parameter $\boldsymbol{\mu}_{q(l)}$. To enrich the database $\mathbf{S}^{\mathrm{hf}}$ with new snapshots, optimal transport is used to interpolate the distribution functions in velocity space. These additional artificial snapshots s^* are defined as the Wasserstein barycenters of the high-fidelity snapshots:

$$s^* = \arg\min_{s} \sum_{l=1}^{K_{\mathrm{hf}}} \lambda_l \mathcal{W}_2(s_l, s)^2,$$

where $\sum_{l=1}^{K_{\mathrm{hf}}} \lambda_l = 1$ and $\lambda_l \geqslant 0$ for $l \in \{1, \ldots, K_{\mathrm{hf}}\}$. Before computing artificial snapshots, the high-fidelity snapshots s_l are normalized because they may have different total mass $\rho(\mathbf{x}_{i(l),j(l),k(l)}, t_{p(l)}; \boldsymbol{\mu}_{q(l)})$. The artificial snapshot s^* is then rescaled by the weighted total mass $\sum_{l=1}^{K_{\mathrm{hf}}} \lambda_l \rho(\mathbf{x}_{i(l),j(l),k(l)}, t_{p(l)}; \boldsymbol{\mu}_{q(l)})$. Given K_{lf} new low-fidelity snapshots s_l^*, these ones are stored in the matrix

$$\mathbf{S}^{\mathrm{lf}} = \begin{pmatrix} s_1^*(\boldsymbol{\xi}_1) & s_2^*(\boldsymbol{\xi}_1) & \cdots & s_{K_{\mathrm{lf}}}^*(\boldsymbol{\xi}_1) \\ s_1^*(\boldsymbol{\xi}_2) & s_2^*(\boldsymbol{\xi}_2) & \cdots & s_{K_{\mathrm{lf}}}^*(\boldsymbol{\xi}_2) \\ \vdots & \vdots & \ddots & \vdots \\ s_1^*(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) & s_2^*(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) & \cdots & s_{K_{\mathrm{lf}}}^*(\boldsymbol{\xi}_{N_{\boldsymbol{\xi}}}) \end{pmatrix} \in \mathbb{R}^{N_{\boldsymbol{\xi}} \times K_{\mathrm{lf}}},$$

and the enriched snapshot database is

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}^{\mathrm{hf}} & \mathbf{S}^{\mathrm{lf}} \end{pmatrix} \in \mathbb{R}^{N_{\boldsymbol{\xi}} \times (K_{\mathrm{hf}} + K_{\mathrm{lf}})}. \tag{III.9}$$

The resulting ROM is then the same as the one described in Section II.3, with the exception of the snapshot database (II.16), which also contains the artificial snapshots (III.9).

In this work, the low-fidelity snapshots are computed by the entropic regularization of the optimal transport problem presented in Section III.2.3, which enables fast computations of the solution. In Algorithm 2, we choose $H_0 = \max_{1 \leqslant l \leqslant K_{\mathrm{hf}}} H(s_l)$, $\gamma = 5 \times 10^{-4}$ and $\epsilon = 10^{-4}$ from the experiments. The run time of interpolating the snapshots is evaluated with respect to the cost of sampling the high-fidelity solution at one point, one time-step and one input parameter. In this respect, we included the cost of interpolating a snapshot of the density distribution function f_h and a snapshot of the discrete Maxwellian distribution function M_{f_h}. Over 100 different runs, the computational time of Algorithm 2 is on average about half that of the HDM. The overall run time of the artificial snapshot procedure will also depend on the strategy adopted to enrich the database: only snapshots that bring new information to the snapshot database are created. In addition, these artificial snapshots can also be generated independently in parallel, while the high-fidelity snapshots are computed sequentially in time.

III.3.2 Prediction of a shock wave

The enrichment of the snapshot database is demonstrated for an application where the predicted solution is significantly different from the training snapshots provided by the sampling of the solution manifold.

We consider a shock tube problem at $Kn = 10^{-5}$. We want to predict the flow solution at input parameter $\mu \in [-2, 2]$ corresponding to different initial macroscopic velocities. The physical space $\Omega_{\mathbf{x}} =]0, 1[$ is discretized using $N_{\mathbf{x}} = 100$ cells, and the velocity space $\Omega_{\boldsymbol{\xi}} =]-20, 20[$ is discretized using $N_{\boldsymbol{\xi}} = 500$ points. The final time is $t_{max} = 0.1$ and the CFL number is 0.1. The initial condition is

$$\begin{cases} \rho_0(x;\mu) = 1,\ u_0(x;\mu) = \mu,\ T_0(x;\mu) = 0.5 & \text{if } x \in]0, 0.5[\\ \rho_0(x;\mu) = 0.125,\ u_0(x;\mu) = \mu,\ T_0(x;\mu) = 0.4 & \text{otherwise,} \end{cases}$$

and we consider free flow boundary conditions. To explore the parameter space, two high-fidelity simulations corresponding to $\mu \in \{-2, 2\}$ are available. The snapshot database $\mathbf{S}^{\mathrm{hf}}_{\phi}$ (resp. $\mathbf{S}^{\mathrm{hf}}_{\psi}$) contains snapshots of ϕ_h and M_{ϕ_h} (resp. ψ_h and M_{ψ_h}) taken at each point in space, every 0.005 time units and for $\mu \in \{-2, 2\}$. In

this case, the training snapshots may be different from the distribution functions that we want to predict.

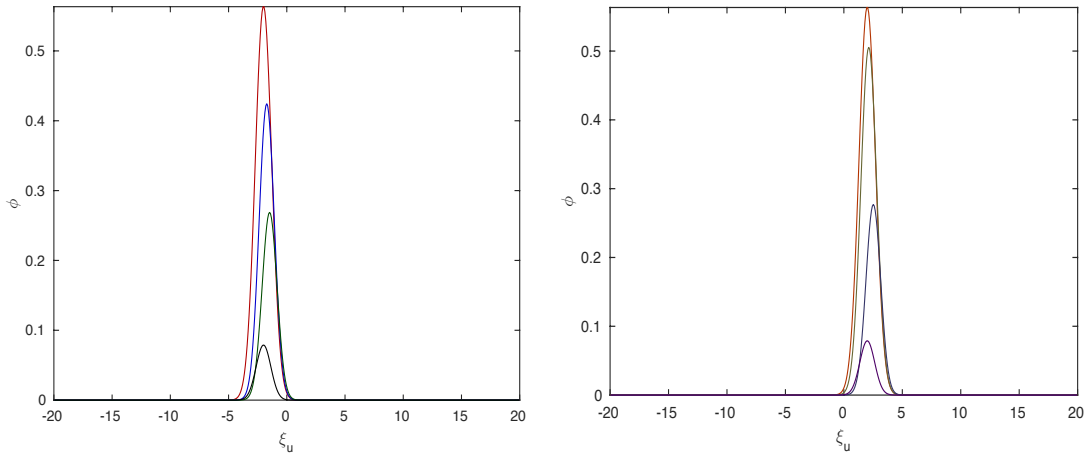


Figure III.4: Examples of 4 high-fidelity snapshots of the simulations $\mu = -2$ (left) and $\mu = 2$ (right) randomly chosen.

As shown in Figure III.4, the support of the high-fidelity snapshots corresponding to these two simulations is almost disjoint. Moreover, the macroscopic velocity of the snapshots, which corresponds to the mean of the distribution functions in velocity space $\big($i.e. $u(x,t;\mu) = \int_{\mathbb{R}} \xi_u \phi(x,\xi_u,t;\mu)\,\mathrm{d}\xi_u\big)$, is around -2.5 (resp. 2.5) in the simulation $\mu = -2.5$ (resp. $\mu = -2.5$). If we want to predict a new distribution function with macroscopic velocity 0, the approximation error may be high even with a large number of basis functions. For this reason, optimal transport is employed to interpolate the high-fidelity snapshots and thus provides additional distribution functions with intermediate macroscopic velocities between -2.5 and 2.5. In this work, we add new snapshots with macroscopic velocities around 0 to the database, as illustrated in Figure III.5. More precisely, at each point in space x and every 0.005 time units, we compute the Wasserstein barycenter s^* between the high-fidelity snapshot s_1 of the simulation $\mu = -2$ and the high-fidelity snapshot s_2 of the simulation $\mu = 2$ at barycentric coordinates $\big\{(s_1, \frac{1}{2}), (s_2, \frac{1}{2})\big\}$.

We evaluate two different ROMs depending on the snapshot database used to construct the basis functions. The first one is built from the high-fidelity snapshots of the simulations $\mu \in \{-2, 2\}$, while the second one is constructed from the high- and low-fidelity snapshots. Figure III.6 shows that the artificial snapshots significantly improve the approximation of the solutions corresponding to $\mu \in [-1.5, 1.5]$. For the training input parameters $\mu = -2$ and $\mu = 2$, the approximation is slightly less accurate because the low-fidelity snapshots bring useless information to represent the distribution functions corresponding to $\mu \in \{-2, 2\}$. On average, the ROM is significantly more robust with the artificial snapshots.

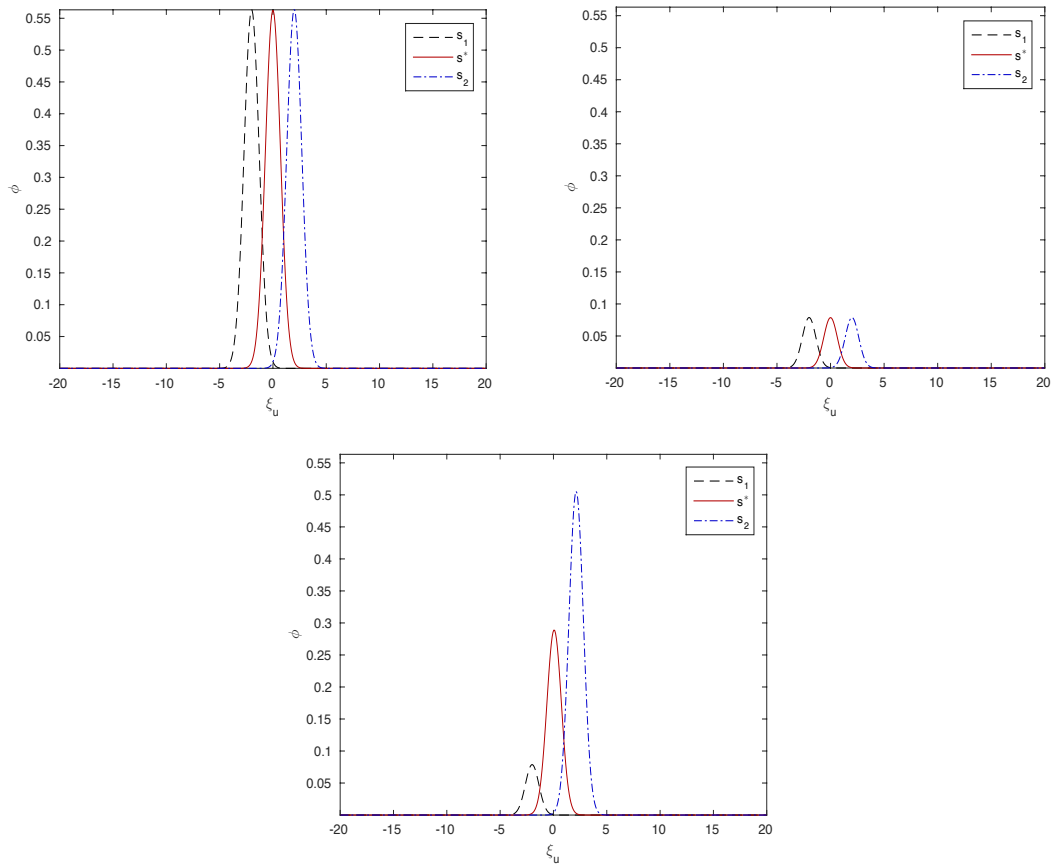


Figure III.5: Examples of artificial snapshots (red) generated from the simulations corresponding to $\mu = -2$ (black) and $\mu = 2$ (blue).

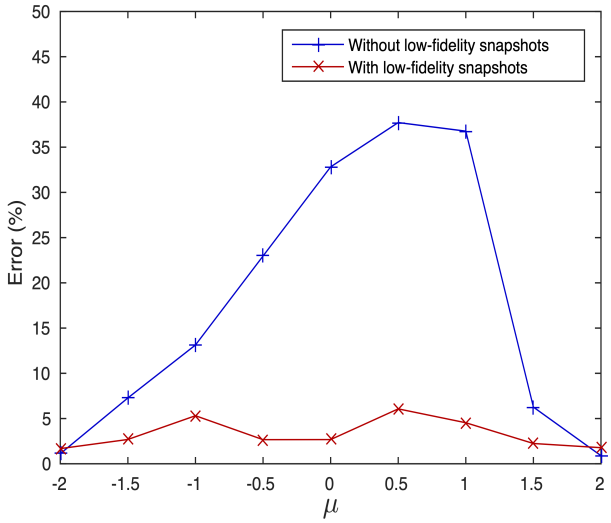


Figure III.6: Performance of the ROMs for the shock wave prediction with $N_{pod}^{\phi} = N_{pod}^{\psi} = 9$.

III.4 Application to snapshots clustering

The second application of the optimal transport problem is to quantify the similarity between the solution snapshots. In Chapter II, we have developed a ROM for the simulation of gas flows in both hydrodynamic and rarefied regimes. In this model, the solution is approximated in velocity space by a small number of basis functions in order to reduce the computational complexity of the simulations. This reduced basis is the same in all the physical domain $\Omega_{\mathbf{x}}$, but different local reduced bases can also be used in each cell to improve the ROM accuracy. However, the computational memory required to store $N_{\mathbf{x}}$ local reduced bases may be prohibitive due to the large number of cells. For this reason, we propose to employ $N_c \in \{1, \dots, N_{\mathbf{x}}\}$ local reduced bases depending on the amount of computational memory available. The physical domain $\Omega_{\mathbf{x}}$ is then partitioned into N_c non-overlapping subdomains $\Omega_i \subseteq \Omega_{\mathbf{x}}$, and the solution is approximated in each subdomain by the corresponding local reduced basis. This decomposition is learned automatically by a cluster analysis method [25] from the snapshot database. The objective is to identify regions where the behaviour of the solution is similar to decompose the domain [30, 5, 65]. In addition, this clustering analysis problem is combined with the L^2-Wasserstein distance instead of the classical L^2-norm to measure the similarity between observations. This metric offers in particular a relevant way to compare the distribution functions contained in the snapshot database. The resulting cluster analysis problem is solved by the k-means algorithm [106, 74], which is a popular method due to its fast execution.

III.4.1 Partitioning of the physical space

Let $\mathbf{S} \in \mathbb{R}^{N_{\boldsymbol{\xi}} \times K}$ be the database containing K snapshots s_l of the distribution functions (f_h and M_{f_h}) collected at point $\mathbf{x}_{i(l)}$ (where a multi-index is used to simplify notation), time instance $t_{p(l)}$ and input parameter $\boldsymbol{\mu}_{q(l)}$. In addition, the snapshot database can also contain artificial snapshots, as described in Section III.3. To measure the similarity between pairs of observations, each physical point $\mathbf{x}_i$ is associated with the observation

$$X_i(\boldsymbol{\xi}) = \sum_{\substack{l=1 \\ i(l)=i}}^{K} s_l(\boldsymbol{\xi}),$$

which represents the distribution of gas particles in velocity space observed at point $\mathbf{x}_i$ over time and parameter space. As these observations have different total mass $\int_{\Omega_{\boldsymbol{\xi}}} X_i(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi}$, they are first normalized. The distance between two points $\mathbf{x}_i$ and $\mathbf{x}_j$ is then defined as the L^2-Wasserstein distance $\mathcal{W}_2(X_i, X_j)$ between the corresponding observations. The objective of clustering is to organise data in a way that maximizes the inner-cluster similarity while minimizing the inter-cluster

similarity. In this work, the $N_{\mathbf{x}}$ points are partitioned into N_c clusters $\mathcal{C}_j$ by the k-means algorithm, which minimizes the within-cluster sum of squares

$$\min_{\mathcal{C}_1,\dots,\mathcal{C}_{N_c}} \sum_{j=1}^{N_c} \sum_{X_i \in \mathcal{C}_j} \mathcal{W}_2(X_i, C_j)^2,$$

where C_j denotes the centroid of the cluster $\mathcal{C}_j$. This centroid is defined as the L^2-Wasserstein centroid of the observations belonging to $\mathcal{C}_j$:

$$C_j = \arg\min_{C} \sum_{X_i \in \mathcal{C}_j} \frac{1}{|\mathcal{C}_j|} \mathcal{W}_2(X_i, C)^2.$$

The k-means algorithm is based on two steps: the assignment and update steps. Given N_c centroids, each observation X_i is assigned to its closest centroid with respect to the Wasserstein distance. Then, in the update step, each centroid is re-computed so as to be the L^2-Wasserstein centroid of the observations belonging to the corresponding cluster. These two steps are repeated until the assignments no longer change. Moreover, the centroids are initialized by the k-means++ [9] since the k-means algorithm is sensitive to the initial choice of centroids. The resulting k-means algorithm is presented in Algorithm 4.

Algorithm 4 k-means algorithm [74]

$C_1, \dots, C_{N_c} \leftarrow k\text{-means++}(X_1, \dots, X_{N_{\mathbf{x}}})$
repeat
  for $j = 1, \dots, N_c$ **do**
    $\mathcal{C}_j \leftarrow \{X_i : \ \mathcal{W}_2(X_i, C_j) \leqslant \mathcal{W}_2(X_i, C_k) \text{ for } 1 \leqslant k \leqslant N_c\}$ ▷ Assignment
  end for
  for $j = 1, \dots, N_c$ **do**
    $C_j \leftarrow \arg\min_{C} \sum_{X_i \in \mathcal{C}_j} \frac{1}{|\mathcal{C}_j|} \mathcal{W}_2(X_i, C)^2$ ▷ Update
  end for
until the assignments no longer change
return $\mathcal{C}_1, \dots, \mathcal{C}_{N_c}$

III.4.2 Local ROM for the BGK equation

Given the clusters $\mathcal{C}_l$ resulting from the snapshot partitioning, the non-overlapping subdomains $\Omega_l \subseteq \Omega_{\mathbf{x}}$ are defined as the union of the cells K_i containing the points $\mathbf{x}_i$ associated with the observations X_i belonging to $\mathcal{C}_l$. The density distribution function is then approximated in each subdomain by

$$\forall \mathbf{x} \in \Omega_l : \quad \widetilde{f_h}(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \sum_{n=1}^{N_{pod}^l} a_n^f(\mathbf{x}, t; \boldsymbol{\mu}) \Phi_n^{f,l}(\boldsymbol{\xi}),$$

where the basis functions are constructed by Proper Orthogonal Decomposition (POD) during the training stage, and the reduced coordinates are determined by the residual minimization method in the prediction stage.

Training stage. As in the global ROM presented in Section II.3, the N_c local reduced bases are constructed by POD (see Sections I.4 and II.3.2.2). First, the sampling of the solution manifold provides in each subdomain a local database $\mathbf{S}^{(l)}$ containing snapshots collected at different points $\mathbf{x} \in \Omega_l$, time instances and input parameters. Then the POD is applied independently to each local snapshot database $\mathbf{S}^{(l)}$, providing N_c local reduced bases $\mathbf{\Phi}^{(l)}$. Compared to the global approach, the local reduced bases $\mathbf{\Phi}^{(l)}$ are more accurate to represent the solution locally because they are specialized to approximate only the snapshots collected in their respective subdomain Ω_l.

Prediction stage. Since the reduced basis is no longer necessarily the same in all the physical domain, the reduced coordinates cannot be determined by the Galerkin method described in Section II.3.3.1. By writing the high-dimensional systems (II.12) and (II.13) as

$$\forall \mathbf{x}_{i,j,k} \in \Omega : \; r_h[f_h](\mathbf{x}_{i,j,k}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = 0,$$

the reduced coordinates are determined in each subdomain Ω_l by substituting the approximate solution for the discrete solution and projecting the residual onto the local basis functions

$$\forall \mathbf{x}_{i,j,k} \in \Omega_l, \; \forall n \in \{1, \dots, N^l_{pod}\} : \; \left\langle r_h[\widetilde{f_h}](\mathbf{x}_{i,j,k}, \boldsymbol{\xi}, t; \boldsymbol{\mu}), \Phi^{f,l}_n(\boldsymbol{\xi}) \right\rangle_{\mathbf{\Theta}} = 0.$$

Note that this Galerkin projection is equivalent to the residual minimization method since the time discretization is in explicit form according to Section I.3.2.2. In this way, the intermediate time-step (II.12) becomes

$$\begin{aligned} \forall \mathbf{x}_{i,j,k} \in \Omega_l : \; \mathbf{a}^{(1)}(\mathbf{x}_{i,j,k}; \boldsymbol{\mu}) = &\left(\mathbf{a}^f(\mathbf{x}_{i,j,k}, t_p; \boldsymbol{\mu}) + \Delta t \frac{\mathbf{a}^{M_f}(\mathbf{x}_{i,j,k}, t_p; \boldsymbol{\mu})}{\tau(\mathbf{x}_{i,j,k}, t_p; \boldsymbol{\mu})} \right) \\ &\times \frac{\tau(\mathbf{x}_{i,j,k}, t_p; \boldsymbol{\mu})}{\Delta t + \tau(\mathbf{x}_{i,j,k}, t_p; \boldsymbol{\mu})}, \end{aligned}$$

and the next time-step (II.13) becomes

$$\begin{aligned} \mathbf{a}^f(\mathbf{x}_{i,j,k}, t_{p+1}; \boldsymbol{\mu}) = \mathbf{a}^f(\mathbf{x}_{i,j,k}, t_p; \boldsymbol{\mu}) &- (\mathbf{A}^{i,j,k} + \mathring{\mathbf{A}}^{i,j,k} + \overset{\star}{\mathbf{A}}^{i,j,k}) \mathbf{a}^{(1)}(\mathbf{x}_{i,j,k}; \boldsymbol{\mu}) \\ &+ \mathbf{B}^{i,j,k} \mathbf{a}^{(1)}(\mathbf{x}_{i-1,j,k}; \boldsymbol{\mu}) - \mathbf{C}^{i,j,k} \mathbf{a}^{(1)}(\mathbf{x}_{i+1,j,k}; \boldsymbol{\mu}) \\ &+ \mathring{\mathbf{B}}^{i,j,k} \mathbf{a}^{(1)}(\mathbf{x}_{i,j-1,k}; \boldsymbol{\mu}) - \mathring{\mathbf{C}}^{i,j,k} \mathbf{a}^{(1)}(\mathbf{x}_{i,j+1,k}; \boldsymbol{\mu}) \\ &+ \overset{\star}{\mathbf{B}}^{i,j,k} \mathbf{a}^{(1)}(\mathbf{x}_{i,j,k-1}; \boldsymbol{\mu}) - \overset{\star}{\mathbf{C}}^{i,j,k} \mathbf{a}^{(1)}(\mathbf{x}_{i,j,k+1}; \boldsymbol{\mu}) \\ &+ \Delta t \frac{\mathbf{a}^{M_f}(\mathbf{x}_{i,j,k}, t_p; \boldsymbol{\mu}) - \mathbf{a}^{(1)}(\mathbf{x}_{i,j,k}; \boldsymbol{\mu})}{\tau(\mathbf{x}_{i,j,k}, t_p; \boldsymbol{\mu})}, \end{aligned}$$

where

$$
\begin{aligned}
A_{n,m}^{i,j,k} &= \frac{\Delta t}{\Delta x}\left\langle |\xi_u|\Phi_m^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k}), \Phi_n^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k})\right\rangle_{\boldsymbol{\Theta}},\\
B_{n,m}^{i,j,k} &= \frac{\Delta t}{\Delta x}\left\langle \max(\xi_u,0)\Phi_m^f(\boldsymbol{\xi};\mathbf{x}_{i-1,j,k}), \Phi_n^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k})\right\rangle_{\boldsymbol{\Theta}},\\
C_{n,m}^{i,j,k} &= \frac{\Delta t}{\Delta x}\left\langle \min(\xi_u,0)\Phi_m^f(\boldsymbol{\xi};\mathbf{x}_{i+1,j,k}), \Phi_n^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k})\right\rangle_{\boldsymbol{\Theta}},\\
\mathring{A}_{n,m}^{i,j,k} &= \frac{\Delta t}{\Delta y}\left\langle |\xi_v|\Phi_m^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k}), \Phi_n^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k})\right\rangle_{\boldsymbol{\Theta}},\\
\mathring{B}_{n,m}^{i,j,k} &= \frac{\Delta t}{\Delta y}\left\langle \max(\xi_v,0)\Phi_m^f(\boldsymbol{\xi};\mathbf{x}_{i,j-1,k}), \Phi_n^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k})\right\rangle_{\boldsymbol{\Theta}},\\
\mathring{C}_{n,m}^{i,j,k} &= \frac{\Delta t}{\Delta y}\left\langle \min(\xi_v,0)\Phi_m^f(\boldsymbol{\xi};\mathbf{x}_{i,j+1,k}), \Phi_n^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k})\right\rangle_{\boldsymbol{\Theta}},\\
\overset{\star}{A}_{n,m}^{i,j,k} &= \frac{\Delta t}{\Delta z}\left\langle |\xi_w|\Phi_m^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k}), \Phi_n^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k})\right\rangle_{\boldsymbol{\Theta}},\\
\overset{\star}{B}_{n,m}^{i,j,k} &= \frac{\Delta t}{\Delta z}\left\langle \max(\xi_w,0)\Phi_m^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k-1}), \Phi_n^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k})\right\rangle_{\boldsymbol{\Theta}},\\
\overset{\star}{C}_{n,m}^{i,j,k} &= \frac{\Delta t}{\Delta z}\left\langle \min(\xi_w,0)\Phi_m^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k+1}), \Phi_n^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k})\right\rangle_{\boldsymbol{\Theta}},
\end{aligned}
$$

and $\Phi_n^f(\boldsymbol{\xi};\mathbf{x}_{i,j,k}) = \Phi_n^{f,l}(\boldsymbol{\xi})$ denotes the local basis functions employed at point $\mathbf{x}_{i,j,k} \in \Omega_l$. The time-step size Δt is the same as the one used in the HDM. In addition, the reduced coordinates of the approximate Maxwellian distribution function are determined to conserve the mass, momentum and energy of the gas, as described in Section II.3.3.2.

III.4.3 Reproduction of a shock wave

The clustering of the snapshot database is evaluated for an application containing a shock wave moving in a part of the domain. In the nonshoked subdomain, the solution can be approximated by a small number of basis functions, while in the shoked subdomain, the dimensionality reduction is limited due to advection-dominated phenomena.

We consider a shock tube problem at $Kn = 10^{-5}$. The physical space $\Omega_{\mathbf{x}} =]0,1[$ is discretized using $N_{\mathbf{x}} = 200$ cells, and the velocity space $\Omega_{\boldsymbol{\xi}} =]-5,5[$ is discretized using $N_{\boldsymbol{\xi}} = 501$ points. The final time is $t_{max} = 0.3$ and the CFL number is 0.25. The initial condition is

$$
\begin{cases}
\rho_0(x) = 10,\ u_0(x) = 0,\ T_0(x) = 0.1 & \text{if } x \in]0,0.5[\\
\rho_0(x) = 0.0125,\ u_0(x) = 0,\ T_0(x) = 0.08 & \text{otherwise,}
\end{cases}
$$

and we consider free flow boundary conditions. To decompose the physical domain, the snapshot database $\mathbf{S}_\phi$ (resp. $\mathbf{S}_\psi$) contains snapshots of ϕ_h and M_{ϕ_h}(resp. ψ_h

and M_{ψ_h}) taken at each point in space and every 0.006 time units. In Figure III.7, we plot the density, macroscopic velocity and temperature of the gas at final time t_{max}.

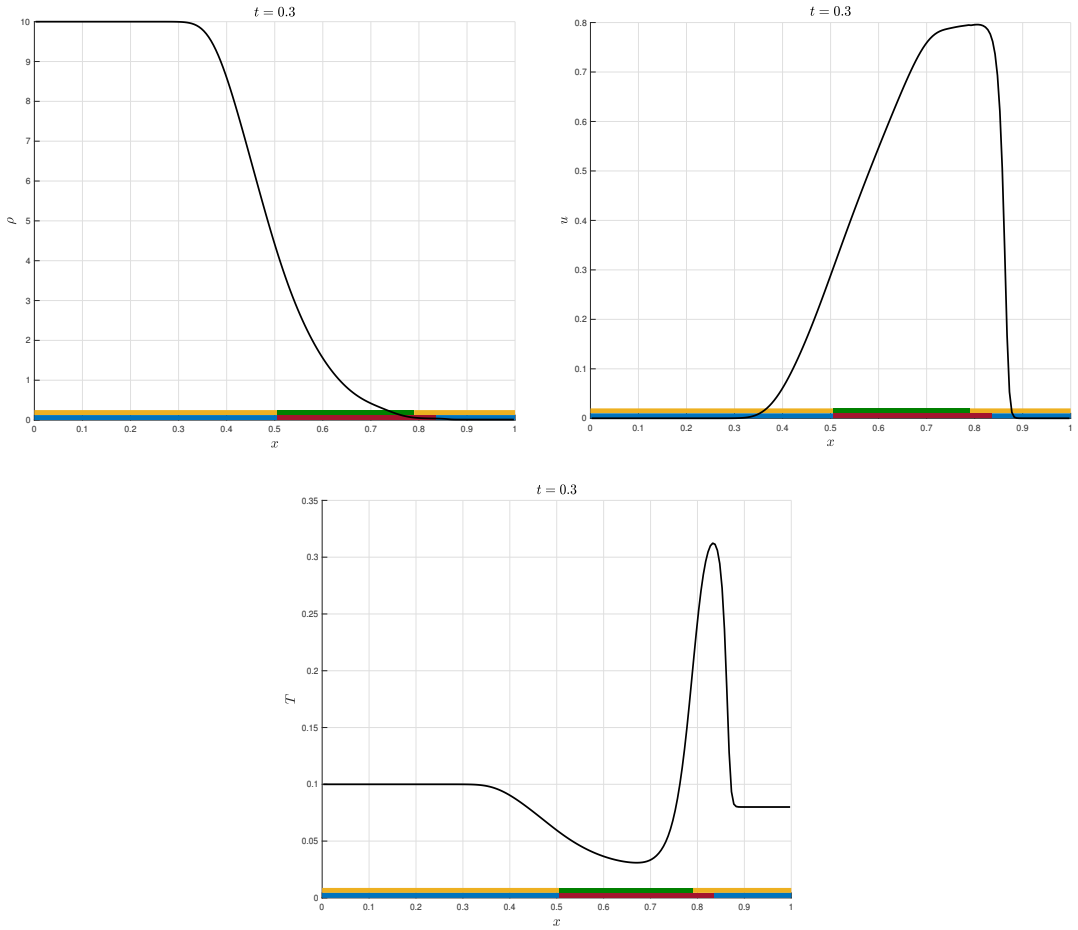


Figure III.7: Decomposition of the physical domain. The yellow and green (resp. blue and red) regions represent the two clusters for ϕ_h (resp. ψ_h). The black line denotes the solution at final time of the local ROM with $N_{pod} = 15$ basis functions.

For $x \in]0, 0.3[\cup]0.9, 1[$, the flow corresponds to the initial condition, and two modes are sufficient to approximate the left and right initial density distribution functions. For $x \in]0.5, 0.8[$, the distribution functions are transported away from the initial state due to the moving shock wave. In particular, for $x \in]0.7, 0.8[$, the mean of the density distribution function in velocity space, which corresponds to the macroscopic velocity of the gas $\left(\text{i.e. } u(x,t) = \int_{\mathbb{R}} \xi_u \phi(x, \xi_u, t)\, \mathrm{d}\xi_u\right)$, is zero at the initialization and then about 0.78 at final time. As presented in Section I.4.4.2, this convection of the distribution functions may lead to a slow decay of the squared singular values of the snapshot matrix. For this reason, the physical domain is decomposed into two subdomains by the k-means algorithm 4. The objective is to identify the subdomains where a significant dimensionality reduction can be

achieve. For simplicity, the one-dimensional optimal transport problem is solved analytically, as described in Section III.2.2.1.

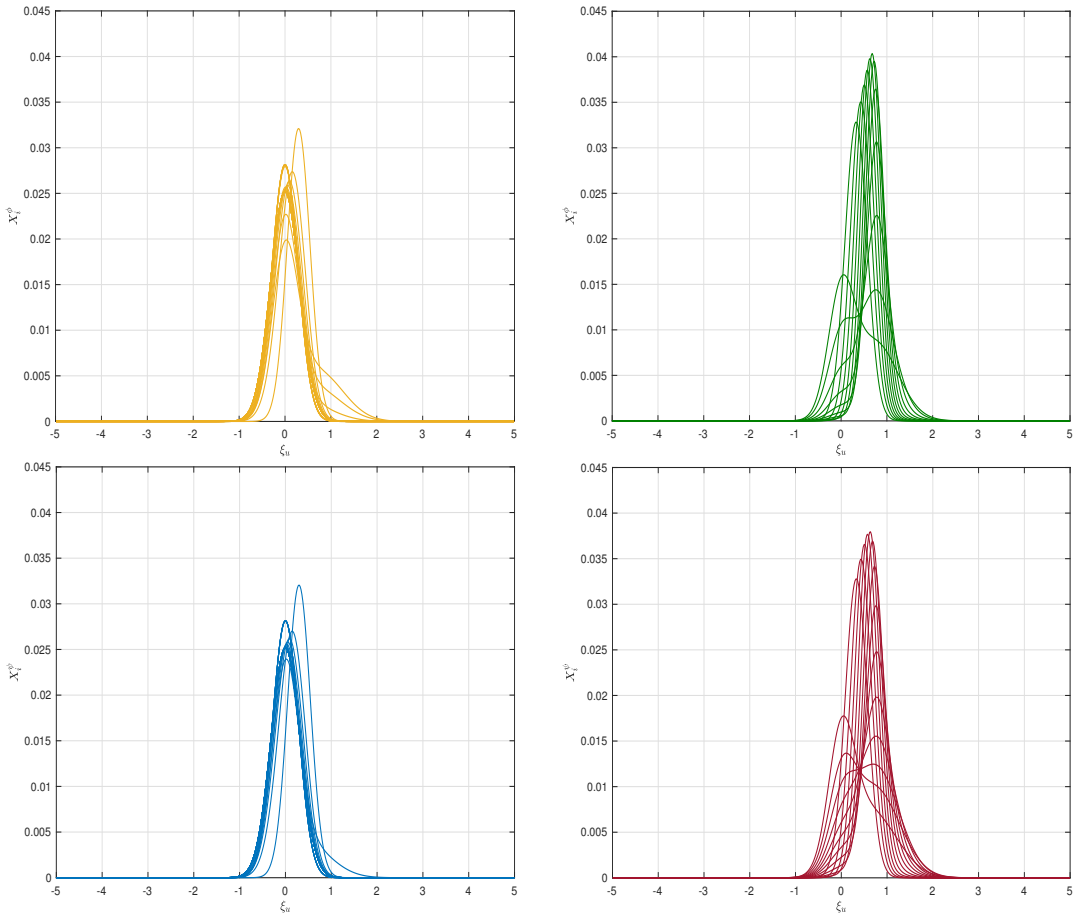


Figure III.8: Examples of local observations X_i randomly chosen in the first cluster (left) and in the second cluster (right), for ϕ_h (top) and ψ_h (bottom).

Figures III.7 and III.8 present the results of the k-means algorithm 4. The domain decomposition for ϕ_h and ψ_h is approximately the same since in the hydrodynamic regime ($Kn = 10^{-5}$), we have $\psi_h(x, \xi_u, t) \approx M_{\psi_h}(x, \xi_u, t) \approx T(x,t) M_{\phi_h}(x, \xi_u, t)$ $\approx T(x,t)\phi_h(x, \xi_u, t)$. As expected, the first cluster contains the distribution functions close to the initial condition, while in the second cluster, the distribution functions are transported by advection. Moreover, the dimensionality reduction of the global and local approaches is presented in Figure III.9. The decay of the squared singular values of the global ROM and of the second cluster of the local ROM is approximately the same since the snapshot database contains the transported distribution functions. In the first cluster of the local ROM, the distribution functions are close to the initial condition, and the decay of the squared singular values is faster.

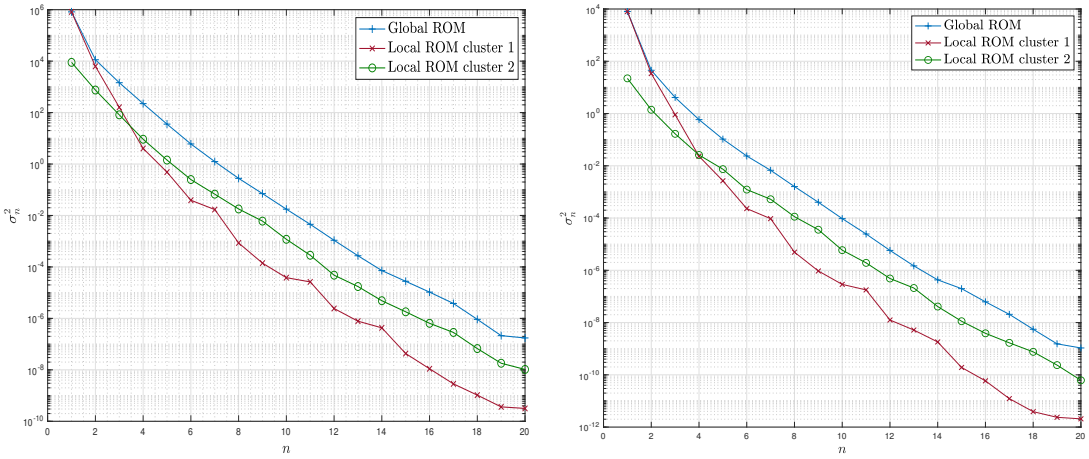


Figure III.9: Squared singular values for the global and local approaches for ϕ_h (left) and ψ_h (right).

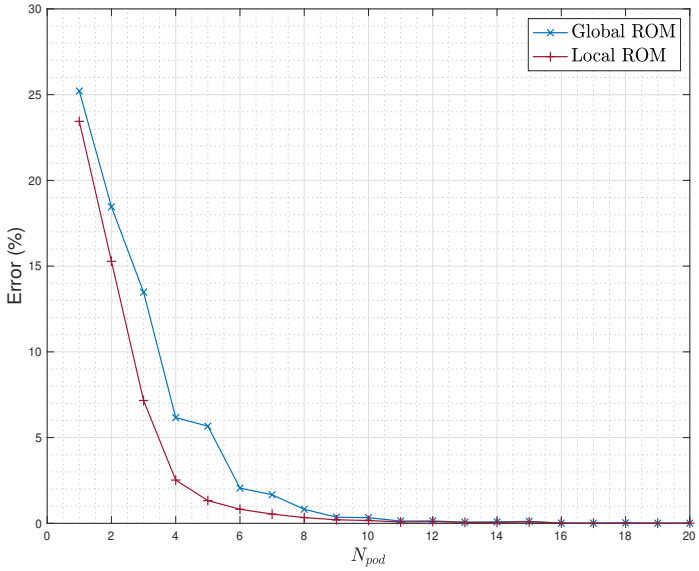


Figure III.10: Comparison of the accuracy of the global and local ROMs.

In Figure III.10, we compare the approximation error of the global ($N_c = 1$) and local ($N_c = 2$) ROMs as a function of the number of basis functions $N_{pod} = N_{pod}^{\phi,1} = N_{pod}^{\phi,2} = N_{pod}^{\psi,1} = N_{pod}^{\psi,2}$. The local ROM is more accurate than the global ROM since the local reduced bases improve the accuracy of the solution approximation. Moreover, in this case, the run time of the global and local ROMs is the same because the number of basis functions used in the global and local ROMs is the same at each point $\mathbf{x}$. However, the number of basis functions can also be different, and in this way, the local approach may allow to reduce the computational complexity of the ROM since less local basis functions are required to obtain accurate approximations.

III.5 Conclusion

In this work, we have presented two applications of the optimal transport problem to improve the ROM developed in Chapter II for the simulation of gas flows in both hydrodynamic and rarefied regimes.

In the first application, we have proposed to complete the sampling of the solution manifold with artificial snapshots. In this strategy, only snapshots that bring new information are created, enabling a fast enrichment of the snapshot database with respect to the HDM. Moreover, we have proposed to define these additional artificial snapshots as the Wasserstein barycenters of the high-fidelity snapshots. In this way, the distribution functions are interpolated in velocity space by optimal transport to enrich the snapshot database without employing the computationally expensive HDM. This improvement has been evaluated on the prediction of a shock wave in 1D. The results show that the artificial snapshots improve the reliability of the ROM for the prediction of solutions corresponding to new input parameters $\boldsymbol{\mu}$.

In the second application, we have proposed to approximate the solution by different local reduced bases instead of employing the same reduced basis in all the physical domain. In this approach, the physical domain $\Omega_{\mathbf{x}}$ is decomposed into $N_c \in \{1, \dots, N_{\mathbf{x}}\}$ subdomains, and the solution is approximated in each subdomain by the corresponding local reduced basis. Moreover, this decomposition is learned automatically by a cluster analysis method from the snapshot database. To measure the similarity between observations, we have proposed to couple the clustering analysis problem with the Wasserstein distance instead of the classical L^2-norm. Furthermore, since the reduced basis is no longer the same everywhere, we have developed a local ROM based on the residual minimization method to compute approximations of the solution at low cost with respect to the HDM. This local ROM has been evaluated on the reproduction of a shock wave in 1D. The results demonstrate that the local approach is more accurate than the global approach. In addition, the local approach may also allow to reduce the computational complexity of the ROM since less local basis functions are required to obtain accurate approximations.

Chapter IV

The discontinuous Galerkin domain decomposition method for reduced-order modeling

IV.1 Introduction

In model order reduction (MOR), the most common method for obtaining the low-dimensional trial subspace is the Proper Orthogonal Decomposition (POD) [87, 48, 103, 20], which hierarchically rearranges the subspace spanned by the solution snapshots according to an energy criterion so that redundant information can be discarded to achieve dimensionality reduction. However, the nature of the problem strongly determines the extent to which one can reduce the dimensionality of the trial subspace. As the problem parameters are varied, singular solution features (e.g. discontinuities and fronts) or compact support phenomena can change their position and shape such that dimensionality reduction is limited. One proposed approach to overcome this limitation is to introduce a mapping applied to the solution in order to improve dimensionally reduction [64, 77, 83, 91]. Alternatively in this work, we adopt the strategy of employing the reduced-order model (ROM) only in those subdomains where a significant dimensionality reduction can be achieved and employing the high-dimensional model (HDM) elsewhere [76, 72, 73, 29, 19].

The next element in MOR is the formulation of the reduced-order system in the prediction stage. The classical approach employs a standard Galerkin projection of the HDM onto the trial subspace. For flow models dominated by advection, special care must be deployed to ensure stability of the resulting ROM. It is well known that standard Galerkin semi-discretization for a linear advection equation is only marginally stable in the discrete energy norm without the introduction of suitable additional numerical diffusion [63]. On the other hand, constructing the ROM based on a discontinuous Galerkin spatial discretization with upwind-

ing of the numerical fluxes is an alternative way to introduce suitable numerical dissipation [62]. The discontinuous Galerkin (DG) approach thus offers the advantage of allowing a modal approximation of the solution and a stable time-explicit discretization.

For the aforementioned reasons, we develop in this thesis a discontinuous Galerkin domain decomposition (DGDD) method [92] in which high-dimensional and reduced-order models coexist. Instead of using a global ROM, the domain is spatially partitioned *a priori* to isolate the subdomains that are anticipated to contain shocks or compact support phenomena. Spatially local ROMs are employed in the subdomains where a significant dimensionality reduction can be achieved, while the HDM is used elsewhere. For the coupling, the ROM is based on the discontinuous Galerkin method [62, 7, 116] in the prediction stage. Compared to the standard DG method [90, 43, 61], the polynomial shape functions are replaced by empirical modes constructed during the training stage by POD in order to best approximate the solution snapshots. In addition, the ROM is equipped with the energy-conserving mesh sampling and weighting (ECSW) hyper-reduction method [50, 51, 55], which provides an empirical quadrature rule enabling the efficient evaluation of the integrals involved in the DG formulation. With this framework, the domain decomposition is applied in a straightforward manner since the global solution is recovered by linking the local solutions at the interface between subdomains through the numerical flux. The accuracy and computational complexity of the resulting method depends on the domain decomposition. If the HDM is employed in a large part of the domain, the accuracy of the coupling model can be very high but the resulting model will be computationally expensive to solve. Conversely, if the ROM is sufficient to approximate the solution in most of the domain, this method allows to significantly reduce the computational cost associated with obtaining model solutions in the prediction stage.

The presentation of the discontinuous Galerkin domain decomposition (DGDD) method is organized as follows. In Section IV.2, we introduces the Euler equations and the HDM employed for their numerical solution. Then, Section IV.3 describes the ROM based on POD in the training stage and on the DG method in the prediction stage. In Section IV.4, we present the domain decomposition and the coupling between the HDM and ROMs. Finally, Section IV.5 demonstrates the accuracy of the proposed method and the reduction of the computational cost versus the HDM for three different applications.

IV.2 High-dimensional model

In this work, we consider the modeling of inviscid compressible flows governed by the Euler equations. The HDM implemented during this thesis to solve these equations is constructed using the discontinuous Galerkin method [90, 43, 61] in space and a TVD Runge-Kutta scheme [100] in time.

IV.2.1 Euler equations

Let the parameter domain $\mathcal{D} \in \mathbb{R}^p$ be a closed and bounded subset of the Euclidean space $\mathbb{R}^p$ with $p \in \mathbb{N}^*$. Moreover, let the physical domain $\Omega \in \mathbb{R}^d$ be a smooth bounded open set with boundary $\partial\Omega$, where $d \in \{1, 2\}$ is the space dimension. In this work, we consider the parameterized Euler equations for $\mathbf{x} \in \Omega$, $t \in \mathbb{R}_+^*$ and $\boldsymbol{\mu} \in \mathcal{D}$:

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) = \mathbf{s}(\mathbf{q}), \tag{IV.1}$$

subject to appropriate initial and boundary conditions. Here, $\mathbf{q} \in \mathbb{R}^{d+2}$ denotes the conservative state variable, $\mathbf{F} = (\mathbf{f}, \mathbf{g})$ denotes the flux and $\mathbf{s}$ denotes the source term. In particular, we will focus on the quasi-1D Euler equations, with

$$\mathbf{q} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{pmatrix}, \quad \mathbf{s} = \begin{pmatrix} -\rho u \frac{1}{A}\frac{\partial A}{\partial x} \\ -\rho u^2 \frac{1}{A}\frac{\partial A}{\partial x} \\ -u(E+p)\frac{1}{A}\frac{\partial A}{\partial x} \end{pmatrix}, \tag{IV.2}$$

and the 2D Euler equations, defined by

$$\mathbf{q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}, \quad \mathbf{s} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{IV.3}$$

where ρ is the density, $\mathbf{u} = (u, v)^T$ is the velocity, E is the total energy, $A(x) \in \mathcal{C}^1(\mathbb{R})$ is a smooth function, for example the cross sectional area of a nozzle, and p is the pressure, given by the equation of state

$$p = (\gamma - 1)\left(E - \rho\frac{\|\mathbf{u}\|_2^2}{2}\right)$$

with γ, the specific heat ratio, taken as $\gamma = 1.4$ in the following.

IV.2.2 Space discretization

The Euler equations (IV.1) are semi-discretized by the discontinuous Galerkin method [90, 43, 61] in space.

IV.2.2.1 Discrete solution

The domain Ω is partitioned into a conforming mesh of N_K non-overlapping micro-cells K_j. In 1D, the domain $\Omega = [x_{min}, x_{max}]$ is divided into uniform intervals $K_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ with $x_j = x_{min} + (j - \frac{1}{2})h$ and $h = \frac{x_{max} - x_{min}}{N_K}$ for $j \in \{1, \ldots, N_K\}$, while in 2D, the domain is decomposed into triangular cells K_j. Each flow variable

q^i (i.e. the density, momentum and energy) is then approximated on each of these cells by a polynomial function

$$\forall \mathbf{x} \in K_j : \; q_h^i(\mathbf{x}, t; \boldsymbol{\mu}) = \sum_{n=1}^{N_\phi} b_n^{i,j}(t; \boldsymbol{\mu}) \phi_n^j(\mathbf{x}), \tag{IV.4}$$

where $b_n^{i,j}$ denote the polynomial coefficients, $\phi_n^j \in H^1(K_j)$ denote the polynomial shape functions, taken as the Legendre polynomials in 1D and the Dubiner polynomials [47] in 2D, and N_ϕ denotes the number of basis functions, which depends on the dimension and on the order of the scheme. In this way, the discrete solution belongs to the space of square-integrable functions $L^2(\Omega)$ (Definition 3) and its restriction on each cell K_j belongs to the Sobolev space $H^1(K_j)$ (Definition 4).

Definition 3. *(L^2 **space**) $L^2(\Omega)$ is the space of square-integrable functions on Ω:*

$$L^2(\Omega) \coloneqq \left\{ f : \Omega \mapsto \mathbb{R}, \text{ such that } \int_\Omega |f^2| \, \mathrm{d}\mathbf{x} < \infty \right\},$$

with the inner product and the norm

$$\langle f, g \rangle_{L^2(\Omega)} = \int_\Omega f(\mathbf{x}) g(\mathbf{x}) \, \mathrm{d}\mathbf{x}, \qquad \|f\|_{L^2(\Omega)} = \sqrt{\langle f, f \rangle_{L^2(\Omega)}}.$$

Definition 4. *(H^1 **space**) $H^1(K_j)$ is the space of square-integrable functions on K_j whose derivatives are also square-integrable on K_j:*

$$H^1(K_j) \coloneqq \left\{ f \in L^2(K_j), \text{ such that } \nabla f \in (L^2(K_j))^d \right\}.$$

Legendre basis. In 1D, the polynomial shape functions are Legendre polynomials $\mathbb{L}_n$, defined by the recurrence formula

$$(n+1)\mathbb{L}_{n+1}(r) = (2n+1) r \mathbb{L}_n(r) - n \mathbb{L}_{n-1}(r),$$

where $\mathbb{L}_0(r) = 1$, $\mathbb{L}_1(r) = r$ and $r \in [-1, 1]$. These polynomials are orthogonal with respect to the L^2-inner product:

$$\int_{-1}^{1} \mathbb{L}_n(r) \mathbb{L}_m(r) \, \mathrm{d}r = \frac{2}{2n+1} \delta_{nm}.$$

Let the change of variables between the reference element $r \in [-1, 1]$ and the interval $x \in K_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ be

$$x = x_j + \frac{h}{2} r.$$

Then, the polynomial shape functions are defined after normalization by

$$\phi_n^j(x) = \sqrt{\frac{2n-1}{2}} \mathbb{L}_{n-1}\left(2 \frac{x - x_j}{h}\right)$$

for $n \in \{1, \ldots, N_\phi\}$ and $j \in \{1, \ldots, N_K\}$.

Dubiner basis. In 2D, the polynomial shape functions are the tensorial product of Jacobi polynomials $\mathbb{J}_n^{(\alpha,\beta)}$, defined by the recurrence formula

$$c_1 \mathbb{J}_{n+1}^{(\alpha,\beta)}(r) = (c_2 + c_3 r)\mathbb{J}_n^{(\alpha,\beta)}(r) - c_4 \mathbb{J}_{n-1}^{(\alpha,\beta)}(r),$$

where

$$\begin{aligned} c_1 &= 2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta), \\ c_2 &= (2n+\alpha+\beta+1)(\alpha^2-\beta^2), \\ c_3 &= (2n+\alpha+\beta)(2n+\alpha+\beta+1)(2n+\alpha+\beta+2), \\ c_4 &= 2(n+\alpha)(n+\beta)(2n+\alpha+\beta+2), \end{aligned}$$

with $\mathbb{J}_0^{(\alpha,\beta)}(r) = 1$, $\mathbb{J}_1^{(\alpha,\beta)}(r) = \frac{\alpha-\beta}{2} + \frac{\alpha+\beta+2}{2} r$ and $r \in [-1,1]$. These polynomials are orthogonal with respect to the following inner product

$$\int_{-1}^{1} (1-r)^\alpha (1+r)^\beta \mathbb{J}_n^{(\alpha,\beta)}(r)\mathbb{J}_m^{(\alpha,\beta)}(r)\,\mathrm{d}r = \frac{2^{\alpha+\beta+1}(n+\alpha)!(n+\beta)!}{(2n+\alpha+\beta+1)(n+\alpha+\beta)!n!}\delta_{nm}.$$

In addition, let the change of variables between the reference element $\mathbf{r} \in T = \left\{\mathbf{r} = (r,s) \in [0,1]^2 : \ r+s \leqslant 1\right\}$ and the triangle $\mathbf{x} \in K_j$ with vertices $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ be

$$\mathbf{x} = (1-r-s)\mathbf{v}_1 + r\mathbf{v}_2 + s\mathbf{v}_3.$$

Then, the polynomial shape functions are defined after normalization by

$$\phi_{n(p,q)}^j(\mathbf{x}) = \sqrt{2(2p+1)(p+q+1)}\mathbb{J}_p^{(0,0)}\left(\frac{2r}{1-s} - 1\right)(1-s)^p \mathbb{J}_q^{(2p+1,0)}(2s-1),$$

where we use the multi-index

$$n(p,q) = 1 + p(q+1) + \frac{q(q+1)}{2} + \frac{p(p+1)}{2}$$

for $0 \leqslant p+q \leqslant -\frac{1}{2} + \sqrt{\frac{1}{4} + 2N_\phi}$ and $1 \leqslant j \leqslant N_K$.

IV.2.2.2 Discontinuous Galerkin method

The polynomial coefficients $b_n^{i,j}$ are determined by the discontinuous Galerkin method. Inserting the discrete solution (IV.4) into the Euler equations (IV.1) leads to the residual

$$\mathbf{r}[\mathbf{q}_h](\mathbf{x},t;\mu) = \frac{\partial \mathbf{q}_h}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}_h) - \mathbf{s}(\mathbf{q}_h).$$

This residual is then enforced to be orthogonal on each micro-cell to the polynomial shape functions. By projecting the residual onto the polynomial shape functions $\phi_n^j \in H^1(K_j)$ and performing an integration by parts, we obtain the variational formulation

$$\int_{K_j} \frac{\partial \mathbf{q}_h}{\partial t}\phi_n^j \,\mathrm{d}\mathbf{x} = \int_{K_j} \mathbf{F}(\mathbf{q}_h)\cdot\nabla\phi_n^j + \mathbf{s}(\mathbf{q}_h)\phi_n^j \,\mathrm{d}\mathbf{x} - \int_{\partial K_j} \mathbf{F}(\mathbf{q}_h)\cdot\mathbf{n}\ \phi_n^j \,\mathrm{d}\boldsymbol{\sigma},$$

where $\mathbf{n} = (n_x, n_y)^T$ denotes the outward unit normal. In the discontinuous Galerkin method, the discrete solution is not assumed to be continuous at the interface between micro-cells. The flux is therefore multiply defined, and for this reason, it is replaced by a numerical flux defining the correct solution at the interface ∂K_j. By replacing the flux in the variational formulation by a numerical flux at micro-cell boundaries, we obtain the semi-discrete discontinuous Galerkin formulation

$$\int_{K_j} \frac{\partial \mathbf{q}_h}{\partial t} \phi_n^j \, \mathrm{d}\mathbf{x} = \int_{K_j} \mathbf{F}(\mathbf{q}_h) \cdot \nabla \phi_n^j + \mathbf{s}(\mathbf{q}_h) \phi_n^j \, \mathrm{d}\mathbf{x} - \int_{\partial K_j} \widehat{\mathbf{F}}(\mathbf{q}_h^-, \mathbf{q}_h^+, \mathbf{n}) \phi_n^j \, \mathrm{d}\boldsymbol{\sigma}, \quad \text{(IV.5)}$$

where $\widehat{\mathbf{F}}$ denotes the numerical flux, i.e. $\widehat{\mathbf{F}} = \widehat{\mathbf{F}}(\mathbf{q}_h^-, \mathbf{q}_h^+, \mathbf{n})$ with $\mathbf{q}_h^-$ and $\mathbf{q}_h^+$, the negative and positive trace, respectively, and $\mathbf{q}_h^+ = \mathbf{q}_{bc}$ at the boundary $\partial\Omega$. In this way, the surface integral is responsible for recovering the global solution from the local solutions and imposing the boundary conditions $\mathbf{q}_{bc}$.

IV.2.2.3 Numerical flux

As the discrete solution is discontinuous at the interface between micro-cells, the flux is multiply defined. For this reason, the flux is replaced by a numerical flux defining the correct solution at micro-cell boundaries. In this work, we consider the local Lax-Friedrich flux [100] and the Harten-Lax-van Leer flux [59]. These numerical fluxes are consistent (Definition 5) and ensure the numerical scheme is consistent according to Lemma 1. In addition, these numerical fluxes are conservative (Definition 6) and since the test subspace contains the constant function, the numerical scheme is also conservative according to Lemma 2.

Definition 5. ***(Consistent flux)*** *The numerical flux is consistent if*

$$\widehat{\mathbf{F}}(\mathbf{q}_h, \mathbf{q}_h, \mathbf{n}) = \mathbf{F}(\mathbf{q}_h) \cdot \mathbf{n}.$$

Lemma 1. ***(Consistent scheme)*** *The scheme is consistent:*

$$\sum_{j=1}^{N_K} \left(\int_{K_j} \frac{\partial \mathbf{q}}{\partial t} \phi_n^j - \mathbf{F}(\mathbf{q}) \cdot \nabla \phi_n^j - \mathbf{s}(\mathbf{q}) \phi_n^j \, \mathrm{d}\mathbf{x} + \int_{\partial K_j} \widehat{\mathbf{F}}(\mathbf{q}^-, \mathbf{q}^+, \mathbf{n}) \phi_n^j \, \mathrm{d}\boldsymbol{\sigma} \right) = 0$$

with $\mathbf{q}$ *the exact solution to the Euler equations* (IV.1)*, if and only if the numerical flux is consistent.*

Proof. Substituting the exact solution for the discrete solution (i.e. $\mathbf{q}_h = \mathbf{q}$) in the discontinuous Galerkin formulation (IV.5) yields

$$\sum_{j=1}^{N_K} \int_{K_j} \frac{\partial \mathbf{q}}{\partial t} \phi_n^j - \mathbf{F}(\mathbf{q}) \cdot \nabla \phi_n^j - \mathbf{s}(\mathbf{q}) \phi_n^j \, \mathrm{d}\mathbf{x} = - \sum_{j=1}^{N_K} \int_{\partial K_j} \widehat{\mathbf{F}}(\mathbf{q}, \mathbf{q}, \mathbf{n}) \phi_n^j \, \mathrm{d}\boldsymbol{\sigma},$$

where $\mathbf{q}^- = \mathbf{q}^+ = \mathbf{q}$ by continuity of the exact solution. By performing an integration by parts, we then obtain

$$\sum_{j=1}^{N_K} \int_{K_j} \left(\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) - \mathbf{s}(\mathbf{q}) \right) \phi_n^j \, \mathrm{d}\mathbf{x} = \sum_{j=1}^{N_K} \int_{\partial K_j} \left(\mathbf{F}(\mathbf{q}) \cdot \mathbf{n} - \widehat{\mathbf{F}}(\mathbf{q}, \mathbf{q}, \mathbf{n}) \right) \phi_n^j \, \mathrm{d}\boldsymbol{\sigma}.$$

Since $\mathbf{q}$ is the solution to the Euler equations (IV.1), the left hand side vanishes, leading to

$$\sum_{j=1}^{N_K} \int_{\partial K_j} \left(\mathbf{F}(\mathbf{q}) \cdot \mathbf{n} - \widehat{\mathbf{F}}(\mathbf{q}, \mathbf{q}, \mathbf{n}) \right) \phi_n^j \, \mathrm{d}\boldsymbol{\sigma} = 0.$$

This equation holds if and only if the numerical flux is consistent. □

Definition 6. ***(Conservative flux)*** *The numerical flux is conservative if*

$$\widehat{\mathbf{F}}(\mathbf{q}_h^-, \mathbf{q}_h^+, \mathbf{n}) = -\widehat{\mathbf{F}}(\mathbf{q}_h^+, \mathbf{q}_h^-, -\mathbf{n}).$$

Lemma 2. ***(Conservative scheme)*** *The numerical scheme is conservative:*

$$\int_\Omega \frac{\partial \mathbf{q}_h}{\partial t} \, \mathrm{d}\mathbf{x} + \int_{\partial\Omega} \widehat{\mathbf{F}}(\mathbf{q}_h^-, \mathbf{q}_{bc}, \mathbf{n}) \, \mathrm{d}\boldsymbol{\sigma} = \int_\Omega \mathbf{s}(\mathbf{q}) \, \mathrm{d}\mathbf{x},$$

if the numerical flux is conservative and the constant function belongs to the test subspace.

Proof. See [41] and Lemma 2.2 in [54]. □

Local Lax-Friedrich flux. The LLF (local Lax-Friedrich) flux [100] is defined by

$$\widehat{\mathbf{F}}(\mathbf{q}_h^-, \mathbf{q}_h^+, \mathbf{n}) = \frac{\mathbf{F}(\mathbf{q}_h^-) + \mathbf{F}(\mathbf{q}_h^+)}{2} \cdot \mathbf{n} + \alpha \frac{\mathbf{q}_h^- - \mathbf{q}_h^+}{2},$$

where the local maximum of the directional flux Jacobian applied to the Euler equations is approximated by

$$\alpha = \max \left(\left\| \mathbf{u}_h^- \right\|_2 + c_h^-, \left\| \mathbf{u}_h^+ \right\|_2 + c_h^+ \right)$$

with $c = \sqrt{\gamma p / \rho}$ the speed of sound.

Harten-Lax-van Leer flux. For the two-dimensional Euler equations, the Riemann problem is first aligned with the face normal direction:

$$\mathbf{q}_n^\pm = \begin{pmatrix} \rho_h^\pm \\ n_x (\rho u)_h^\pm + n_y (\rho v)_h^\pm \\ n_x (\rho u)_h^\pm - n_y (\rho v)_h^\pm \\ E_h^\pm \end{pmatrix},$$

as proposed in [61]. The Riemann problem then consists of three states, separated by two waves propagating at speed s^- and s^+, with $s^- < s^+$. The HLL (Harten-Lax-van Leer) flux [59] associated with this Riemann problem is defined by

$$\widehat{\mathbf{f}}^{hll}(\mathbf{q}_n^-, \mathbf{q}_n^+) = \begin{cases} \mathbf{f}(\mathbf{q}_n^-) & \text{if } 0 \leqslant s^- \\ \dfrac{s^+\mathbf{f}(\mathbf{q}_n^-) - s^-\mathbf{f}(\mathbf{q}_n^+) + s^-s^+(\mathbf{q}_n^- - \mathbf{q}_n^+)}{s^+ - s^-} & \text{if } s^- \leqslant 0 \leqslant s^+ \\ \mathbf{f}(\mathbf{q}_n^+) & \text{if } s^+ \leqslant 0. \end{cases}$$

Following the approach presented in [59], the wave speeds are estimated from

$$s^- = \min(u_n^- - c_n^-, u^* - c^*) \quad \text{and} \quad s^+ = \max(u_n^+ + c_n^+, u^* + c^*),$$

where the minimum/maximum is taken over the eigenvalues of the linearized flux Jacobian, and the intermediate state between the two waves is taken to be the Roe average:

$$\begin{aligned} u^* &= \frac{\sqrt{\rho_n^-}u_n^- + \sqrt{\rho_n^+}u_n^+}{\sqrt{\rho_n^-} + \sqrt{\rho_n^+}}, \\ v^* &= \frac{\sqrt{\rho_n^-}v_n^- + \sqrt{\rho_n^+}v_n^+}{\sqrt{\rho_n^-} + \sqrt{\rho_n^+}}, \\ H^* &= \frac{\sqrt{\rho_n^-}H_n^- + \sqrt{\rho_n^+}H_n^+}{\sqrt{\rho_n^-} + \sqrt{\rho_n^+}}, \\ c^* &= \sqrt{(\gamma-1)\left[H^* - \frac{(u^*)^2 + (v^*)^2}{2}\right]}, \end{aligned}$$

where $H = (E + p)/\rho$ denotes the total enthalpy. Finally, the numerical flux is rotated back to Cartesian coordinates as follows

$$\widehat{\mathbf{F}}(\mathbf{q}_h^-, \mathbf{q}_h^+, \mathbf{n}) = \begin{pmatrix} \widehat{f}_1^{hll}(\mathbf{q}_n^-, \mathbf{q}_n^+) \\ n_x\widehat{f}_2^{hll}(\mathbf{q}_n^-, \mathbf{q}_n^+) - n_y\widehat{f}_3^{hll}(\mathbf{q}_n^-, \mathbf{q}_n^+) \\ n_x\widehat{f}_2^{hll}(\mathbf{q}_n^-, \mathbf{q}_n^+) + n_y\widehat{f}_3^{hll}(\mathbf{q}_n^-, \mathbf{q}_n^+) \\ \widehat{f}_4^{hll}(\mathbf{q}_n^-, \mathbf{q}_n^+) \end{pmatrix}.$$

IV.2.2.4 Numerical integration

The integrals on the real line (i.e. the volume integral in 1D and the surface integral in 2D) are evaluated by the Gauss-Legendre quadrature rule, while the two-dimensional volume integrals in (IV.5) are approximated by the symmetric rule [115]. The discrete inner product $\langle \cdot, \cdot \rangle_{\mathbf{\Theta}}$ associated with the L^2-norm is then induced by the diagonal matrix $\mathbf{\Theta} \in \mathbb{R}^{(N_K N_\mathbf{r}) \times (N_K N_\mathbf{r})}$ containing the weights of the quadrature rule on the diagonal.

Gauss-Legendre quadrature rule. Let $I = [-1, 1]$ be the reference element and $[v_1, v_2]$ be the domain of integration, i.e. an interval K_j in 1D or an edge of the triangle K_j in 2D. The (continuously differentiable) change of variables $\varphi : I \to [v_1, v_2]$ between the reference element and the domain of integration is defined by

$$\varphi(r) = \frac{1-r}{2} v_1 + \frac{1+r}{2} v_2.$$

The integrals are approximated by the $N_{\mathbf{r}}$-point Gauss-Legendre quadrature rule, which is exact for polynomials up to degree $2N_{\mathbf{r}} - 1$,

$$\begin{aligned}\int_{v_1}^{v_2} f(x)\,\mathrm{d}x &= \int_I f(\varphi(r))\varphi'(r)\,\mathrm{d}r \\ &= \frac{v_2 - v_1}{2} \int_I f(\varphi(r))\,\mathrm{d}r \\ &\approx \frac{v_2 - v_1}{2} \sum_{n=1}^{N_{\mathbf{r}}} \omega_n f(\varphi(r_n)).\end{aligned}$$

Here, r_n and ω_n denote the quadrature points and weights, respectively, and they are tabulated below.

$N_{\mathbf{r}}$	2		3		
r_n	$-\sqrt{\frac{1}{3}}$	$\sqrt{\frac{1}{3}}$	$-\sqrt{\frac{5}{3}}$	0	$\sqrt{\frac{5}{3}}$
ω_n	1	1	$\frac{5}{9}$	$\frac{8}{9}$	$\frac{5}{9}$

Table IV.1: Points and weights of the second ($N_{\mathbf{r}} = 2$) and third ($N_{\mathbf{r}} = 3$) order Gauss-Legendre quadrature rules.

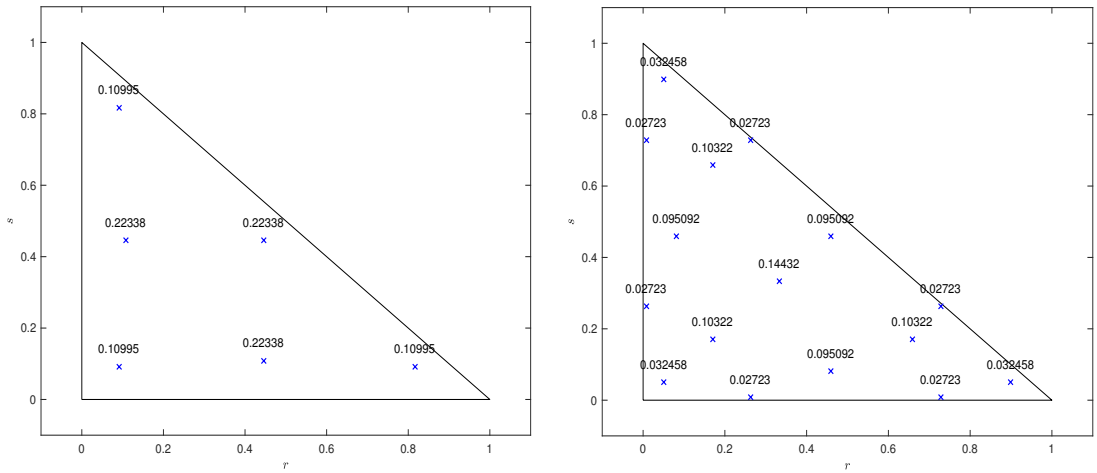
Symmetric rule. Let the reference element be $T = \left\{\mathbf{r} = (r, s) \in [0, 1]^2 : r + s \leqslant 1\right\}$ and the domain of integration be the triangle K_j with vertices $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$. The (injective and continuously differentiable) change of variables $\varphi : T \to K_j$ between the reference element and the domain of integration is defined by

$$\varphi(\mathbf{r}) = (1 - r - s)\mathbf{v}_1 + r\mathbf{v}_2 + s\mathbf{v}_3.$$

To evaluate the integrals, we consider the symmetric rule:

$$\begin{aligned}\int_{K_j} f(\mathbf{x})\,\mathrm{d}\mathbf{x} &= \int_T f(\varphi(\mathbf{r}))|J_\varphi(\mathbf{r})|\,\mathrm{d}\mathbf{r} \\ &= |\det(\mathbf{v}_2 - \mathbf{v}_1, \mathbf{v}_3 - \mathbf{v}_1)| \int_T f(\varphi(\mathbf{r}))\,\mathrm{d}\mathbf{r} \\ &\approx |\det(\mathbf{v}_2 - \mathbf{v}_1, \mathbf{v}_3 - \mathbf{v}_1)| \sum_{n=1}^{N_{\mathbf{r}}} \omega_n f(\varphi(\mathbf{r}_n)),\end{aligned}$$

where J_φ denotes the Jacobian determinant of φ. Here, $\mathbf{r}_n$ and ω_n denote the quadrature points and weights, respectively, and they are displayed in Figure IV.1.



(a) $N_{\mathbf{r}} = 6$ (b) $N_{\mathbf{r}} = 16$

Figure IV.1: Points and weights of the symmetric quadrature rule for the second ($N_{\mathbf{r}} = 6$) and third ($N_{\mathbf{r}} = 16$) order schemes.

IV.2.2.5 Slope limiter

In high-order polynomial approximations, spurious oscillations may appear in the presence of shock or large gradient due to the Gibbs phenomenon. For this reason, a slope limiter is introduced in non-smooth regions. In this work, we consider the minmod limiter [93] in 1D and the Barth-Jespersen limiter [69] in 2D.

Minmod limiter. In 1D, the discrete solution is reduced to a linear polynomial in non-smooth regions

$$\forall x \in K_j : \; q_h^i(x,t;\boldsymbol{\mu}) = \overline{q}_{i,j}(t;\boldsymbol{\mu}) + \alpha_{i,j}(t;\boldsymbol{\mu}) q'_{i,j}(t;\boldsymbol{\mu})(x - x_j),$$

where $\overline{q}_{i,j}\left(= \frac{\sqrt{2}}{2} b_1^{i,j}\right)$ denotes the mean value, $q'_{i,j}\left(= \frac{\sqrt{6}}{2} b_2^{i,j}\right)$ denotes the slope and $\alpha_{i,j}$ denotes the correction factor. In the minmod limiter [93], the correction factor is defined by

$$\alpha_{i,j} = \frac{1}{q'_{i,j}} \text{minmod}\left(q'_{i,j}, \frac{\overline{q}_{i,j+1} - \overline{q}_{i,j}}{h}, \frac{\overline{q}_{i,j} - \overline{q}_{i,j-1}}{h}\right),$$

where

$$\text{minmod}(a,b,c) := \begin{cases} \text{sign}(a)\min(|a|,|b|,|c|) & \text{if } \text{sign}(a) = \text{sign}(b) = \text{sign}(c) \\ 0 & \text{otherwise.} \end{cases}$$

This limiter ensures the TVD property of the scheme [58] and reduces the scheme to first order (i.e. $\alpha_{i,j} = 0$) in regions with strong gradients.

Barth-Jespersen limiter. In 2D, the discrete solution is reduced to a linear function in non-smooth regions

$$\forall \mathbf{x} \in K_j : \; q_h^i(\mathbf{x}, t; \boldsymbol{\mu}) = \overline{q}_{i,j}(t; \boldsymbol{\mu}) + \alpha_{i,j}(t; \boldsymbol{\mu}) \left(\nabla q_{i,j}(t; \boldsymbol{\mu})\right) \cdot (\mathbf{x} - \overline{\mathbf{x}}_j),$$

where $\overline{\mathbf{x}}_j$ denotes the centroid of the micro-cell K_j, $\overline{q}_{i,j}(t; \boldsymbol{\mu}) = q_h^i(\overline{\mathbf{x}}_j, t; \boldsymbol{\mu})$ denotes the mean value, $\nabla q_{i,j}$ denotes the slope and $\alpha_{i,j}$ denotes the correction factor. In the Barth-Jespersen limiter [69], the maximum admissible slope is defined so that the discrete solution is bounded by the maximum and minimum values found in K_j or in one of its neighbours

$$\forall \mathbf{x} \in K_j : \; q_{min}^{i,j}(t; \boldsymbol{\mu}) \leqslant q_h^i(\mathbf{x}, t; \boldsymbol{\mu}) \leqslant q_{max}^{i,j}(t; \boldsymbol{\mu}). \tag{IV.6}$$

Due to linearity, the discrete solution q_h^i reaches its extrema at the vertices $\mathbf{v}_1$, $\mathbf{v}_2$ or $\mathbf{v}_3$ of the triangle K_j. The correction factor enforcing condition (IV.6) is therefore given by

$$\alpha_{i,j}(t; \boldsymbol{\mu}) = \min_{1 \leqslant k \leqslant 3} \begin{cases} \min\left(1, \frac{q_{max}^{i,j}(t;\boldsymbol{\mu}) - \overline{q}_{i,j}(t;\boldsymbol{\mu})}{q_h^i(\mathbf{v}_k^-, t;\boldsymbol{\mu}) - \overline{q}_{i,j}(t;\boldsymbol{\mu})}\right) & \text{if } q_h^i(\mathbf{v}_k^-, t; \boldsymbol{\mu}) > \overline{q}_{i,j}(t; \boldsymbol{\mu}) \\ 1 & \text{if } q_h^i(\mathbf{v}_k^-, t; \boldsymbol{\mu}) = \overline{q}_{i,j}(t; \boldsymbol{\mu}) \\ \min\left(1, \frac{q_{min}^{i,j}(t;\boldsymbol{\mu}) - \overline{q}_{i,j}(t;\boldsymbol{\mu})}{q_h^i(\mathbf{v}_k^-, t;\boldsymbol{\mu}) - \overline{q}_{i,j}(t;\boldsymbol{\mu})}\right) & \text{if } q_h^i(\mathbf{v}_k^-, t; \boldsymbol{\mu}) < \overline{q}_{i,j}(t; \boldsymbol{\mu}) \end{cases},$$

where $q_h^i(\mathbf{v}_k^-, t; \boldsymbol{\mu})$ denotes the negative trace of the discrete solution q_h^i at vertices $\mathbf{v}_k$ of the triangle K_j.

IV.2.3 Time discretization

The time is discretized by an explicit TVD Runge-Kutta scheme [100]. Writing the semi-discrete system (IV.5) as

$$\frac{\mathrm{d}\mathbf{b}}{\mathrm{d}t} = \mathbf{L}_h(\mathbf{b}, t; \boldsymbol{\mu}),$$

where the vector $\mathbf{b} \in \mathbb{R}^{(d+2)N_K N_\phi}$ contains the polynomial coefficients $b_n^{i,j}$ and $\mathbf{L}_h(\mathbf{b}, t; \boldsymbol{\mu}) \in \mathbb{R}^{(d+2)N_K N_\phi}$ results from the spatial discretization, the second-order scheme reads

$$\begin{aligned} \mathbf{b}^{(1)} &= \mathbf{b}(t_k; \boldsymbol{\mu}) + \Delta t \mathbf{L}_h(\mathbf{b}(t_k; \boldsymbol{\mu}), t_k; \boldsymbol{\mu}), \\ \mathbf{b}(t_{k+1}; \boldsymbol{\mu}) &= \frac{1}{2}\mathbf{b}(t_k; \boldsymbol{\mu}) + \frac{1}{2}\mathbf{b}^{(1)} + \frac{\Delta t}{2}\mathbf{L}_h(\mathbf{b}^{(1)}, t_{k+1}; \boldsymbol{\mu}), \end{aligned}$$

and the third-order scheme reads

$$\begin{aligned} \mathbf{b}^{(1)} &= \mathbf{b}(t_k; \boldsymbol{\mu}) + \Delta t \mathbf{L}_h(\mathbf{b}(t_k; \boldsymbol{\mu}), t_k; \boldsymbol{\mu}), \\ \mathbf{b}^{(2)} &= \frac{3}{4}\mathbf{b}(t_k; \boldsymbol{\mu}) + \frac{1}{4}\mathbf{b}^{(1)} + \frac{\Delta t}{4}\mathbf{L}_h(\mathbf{b}^{(1)}, t_{k+1}; \boldsymbol{\mu}), \\ \mathbf{b}(t_{k+1}; \boldsymbol{\mu}) &= \frac{1}{3}\mathbf{b}(t_k; \boldsymbol{\mu}) + \frac{2}{3}\mathbf{b}^{(2)} + \frac{2\Delta t}{3}\mathbf{L}_h(\mathbf{b}^{(2)}, t_{k+\frac{1}{2}}; \boldsymbol{\mu}). \end{aligned}$$

The initial solution $\mathbf{b}(t_0;\boldsymbol{\mu})$ of these systems is given by the orthogonal projection of the initial condition $\mathbf{q}_0(\mathbf{x};\boldsymbol{\mu})$ onto the polynomial shape functions:

$$b_n^{i,j}(t_0;\boldsymbol{\mu}) = \int_{K_j} q_0^i(\mathbf{x};\boldsymbol{\mu})\phi_n^j(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

IV.3 Reduced-order model based on the discontinuous Galerkin method

To reduce the computational cost of the HDM, we develop in this thesis [92] a ROM based on Proper Orthogonal Decomposition [87, 48, 103, 20] in the training stage and on the discontinuous Galerkin method [62, 7, 116] in the prediction stage. Compared to the standard DG method, the polynomial shape functions are replaced by POD modes in order to best approximate the solution snapshots. In addition, the ROM is equipped with hyper-reduction techniques to ensure the computational complexity of the ROM is independent of the size of the mesh.

IV.3.1 Solution approximation

In the ROM, each component of the solution is approximated in space by a small number of basis functions Φ_n^i

$$\forall i \in \{1,\dots,d+2\}:\ \widetilde{q}_h^i(\mathbf{x},t;\boldsymbol{\mu}) = q_o^i(\mathbf{x}) + \sum_{n=1}^{M_i} a_n^i(t;\boldsymbol{\mu})\Phi_n^i(\mathbf{x}) \tag{IV.7}$$

in order to reduce the number of degrees of freedom. The offset q_o^i and the basis functions Φ_n^i are constructed during the training stage by Proper Orthogonal Decomposition (POD), and the reduced coordinates a_n^i are determined in the prediction stage by the discontinuous Galerkin method.

IV.3.2 Training stage

For the sampling of the solution manifold, the HDM provides a database of N_s snapshots of the high-fidelity solution collected at different time instances and input parameters. Let $s_h^{i,l}(\mathbf{x}) = q_h^i(\mathbf{x}, t_{k(l)};\boldsymbol{\mu}_{j(l)})$ be the l^{th} snapshot of the conserved variable q_h^i collected at time instance $t_{k(l)}$ and input parameter $\boldsymbol{\mu}_{j(l)}$. The offset is defined as the mean of the snapshots over time and parameter space:

$$q_o^i(\mathbf{x}) = \frac{\sum_{l=1}^{N_s} s_h^{i,l}(\mathbf{x})}{N_s}.$$

The basis functions Φ_n^i are then constructed by POD (Section I.4) to minimize, in the least-squares sense, the difference between the snapshots $s_h^{i,l}$ and their projections $\widehat{s}_h^{i,l}$ onto the trial subspace. That is, the basis functions Φ_n^i are the solution to the minimization problem

$$\begin{cases} \underset{\boldsymbol{\Phi}^{(i)} \in \mathbb{R}^{N_{\mathbf{x}} \times M_i}}{\text{minimize}} & \left\| \mathbf{S}^{(i)} - \boldsymbol{\Phi}^{(i)} (\boldsymbol{\Phi}^{(i)})^T \boldsymbol{\Theta} \mathbf{S}^{(i)} \right\|_{F_{\boldsymbol{\Theta}}}^2 \\ \text{subject to} & (\boldsymbol{\Phi}^{(i)})^T \boldsymbol{\Theta} \boldsymbol{\Phi}^{(i)} = \mathbf{I}_{M_i}, \end{cases}$$

where the snapshots are stored in the matrix

$$\mathbf{S}^{(i)} = \begin{pmatrix} \overline{s}_h^{i,1}(\mathbf{x}_1) & \overline{s}_h^{i,2}(\mathbf{x}_1) & \cdots & \overline{s}_h^{i,N_s}(\mathbf{x}_1) \\ \overline{s}_h^{i,1}(\mathbf{x}_2) & \overline{s}_h^{i,2}(\mathbf{x}_2) & \cdots & \overline{s}_h^{i,N_s}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \overline{s}_h^{i,1}(\mathbf{x}_{N_{\mathbf{x}}}) & \overline{s}_h^{i,2}(\mathbf{x}_{N_{\mathbf{x}}}) & \cdots & \overline{s}_h^{i,N_s}(\mathbf{x}_{N_{\mathbf{x}}}) \end{pmatrix} \in \mathbb{R}^{N_{\mathbf{x}} \times N_s}$$

with $\overline{s}_h^{i,1}(\mathbf{x}) = s_h^{i,1}(\mathbf{x}) - q_o^i(\mathbf{x})$, the basis functions are stored in the matrix

$$\boldsymbol{\Phi}^{(i)} = \begin{pmatrix} \Phi_1^i(\mathbf{x}_1) & \Phi_2^i(\mathbf{x}_1) & \cdots & \Phi_{M_i}^i(\mathbf{x}_1) \\ \Phi_1^i(\mathbf{x}_2) & \Phi_2^i(\mathbf{x}_2) & \cdots & \Phi_{M_i}^i(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_1^i(\mathbf{x}_{N_{\mathbf{x}}}) & \Phi_2^i(\mathbf{x}_{N_{\mathbf{x}}}) & \cdots & \Phi_{M_i}^i(\mathbf{x}_{N_{\mathbf{x}}}) \end{pmatrix} \in \mathbb{R}^{N_{\mathbf{x}} \times M_i},$$

and $\boldsymbol{\Theta} \in \mathbb{R}^{N_{\mathbf{x}} \times N_{\mathbf{x}}}$ corresponds to the SPD matrix defined in Section IV.2.2.4. According to the Schmidt-Eckart-Young-Mirsky theorem 1, the basis functions are given by

$$\boldsymbol{\Phi}^{(i)} = (\boldsymbol{\Theta}^{\frac{1}{2}})^{-T} \begin{pmatrix} U_{1,1} & \cdots & U_{1,M_i} \\ \vdots & & \vdots \\ U_{N_{\mathbf{x}},1} & \cdots & U_{N_{\mathbf{x}},M_i} \end{pmatrix},$$

where $\boldsymbol{\Theta} = \boldsymbol{\Theta}^{\frac{1}{2}} (\boldsymbol{\Theta}^{\frac{1}{2}})^T$ is the Cholesky decomposition of $\boldsymbol{\Theta}$, $\widetilde{\mathbf{S}}^{(i)} = (\boldsymbol{\Theta}^{\frac{1}{2}})^T \mathbf{S}^{(i)}$ and $\widetilde{\mathbf{S}}^{(i)} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^T$ is the singular value decomposition (SVD) of $\widetilde{\mathbf{S}}^{(i)}$.

In the discontinuous Galerkin formulation (IV.8), the derivatives of the basis functions $\nabla \Phi_n^i$ are also required. As the basis functions are a linear combination of the snapshots, they are derived analytically to obtain

$$\frac{\partial \boldsymbol{\Phi}^{(i)}}{\partial x} = \frac{\partial \mathbf{S}^{(i)}}{\partial x} \begin{pmatrix} V_{1,1} & \cdots & V_{1,M_i} \\ \vdots & & \vdots \\ V_{N_s,1} & \cdots & V_{N_s,M_i} \end{pmatrix} \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_{M_i} \end{pmatrix}^{-1}$$

and

$$\frac{\partial \boldsymbol{\Phi}^{(i)}}{\partial y} = \frac{\partial \mathbf{S}^{(i)}}{\partial y} \begin{pmatrix} V_{1,1} & \cdots & V_{1,M_i} \\ \vdots & & \vdots \\ V_{N_s,1} & \cdots & V_{N_s,M_i} \end{pmatrix} \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_{M_i} \end{pmatrix}^{-1},$$

where $\{\sigma_n\}_{1\leqslant n\leqslant N_i}$ are the singular values of $\widetilde{\mathbf{S}}^{(i)}$. If the method of snapshots (Section I.4.3.2) is employed, then $\mathbf{V}$ and $\boldsymbol{\Sigma}$ can be obtained from the SVD of $(\mathbf{S}^{(i)})^T\mathbf{S}^{(i)}$. In the same way, the derivatives are given in the classical method (Section I.4.3.1) by

$$\frac{\partial \boldsymbol{\Phi}^{(i)}}{\partial x} = \frac{\partial \mathbf{S}^{(i)}}{\partial x}\widetilde{\mathbf{S}}^{(i)} \begin{pmatrix} U_{1,1} & \cdots & U_{1,M_i} \\ \vdots & & \vdots \\ U_{N_{\mathbf{x}},1} & \cdots & U_{N_{\mathbf{x}},M_i} \end{pmatrix} \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_{M_i} \end{pmatrix}^{-2}$$

and

$$\frac{\partial \boldsymbol{\Phi}^{(i)}}{\partial y} = \frac{\partial \mathbf{S}^{(i)}}{\partial y}\widetilde{\mathbf{S}}^{(i)} \begin{pmatrix} U_{1,1} & \cdots & U_{1,M_i} \\ \vdots & & \vdots \\ U_{N_{\mathbf{x}},1} & \cdots & U_{N_{\mathbf{x}},M_i} \end{pmatrix} \begin{pmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_{M_i} \end{pmatrix}^{-2},$$

where $\mathbf{U}$ and $\boldsymbol{\Sigma}$ are obtained from the SVD of $\mathbf{S}^{(i)}(\mathbf{S}^{(i)})^T$.

IV.3.3 Prediction stage

The reduced coordinates a_n^i are determined at low cost during the prediction stage by the discontinuous Galerkin method [62, 7, 116]. Compared to the DG method employed in the HDM, the polynomial shape functions are replaced by the basis functions constructed by POD. The resulting nonlinear system of ODEs is equipped with hyper-reduction techniques, such as the precomputation-based approach and the energy-conserving mesh sampling and weighting (ECSW) method [50, 51, 55], which enable the efficient evaluation of the integrals involved in the DG formulation. The ROM is finally discretized in time by the same explicit TVD Runge-Kutta scheme [100] used in the HDM.

IV.3.3.1 Discontinuous Galerkin method

In the discontinuous Galerkin method, the approximate solution (IV.7) is inserted into the Euler equations (IV.1), leading to the residual

$$\mathbf{r}[\widetilde{\mathbf{q}}_h](\mathbf{x}, t; \boldsymbol{\mu}) = \frac{\partial \widetilde{\mathbf{q}}_h}{\partial t} + \nabla \cdot \mathbf{F}(\widetilde{\mathbf{q}}_h) - \mathbf{s}(\widetilde{\mathbf{q}}_h).$$

Projecting this residual onto the basis functions Φ_n^i, performing an integration by parts over each micro-cell K_j and replacing the flux by a numerical flux at micro-cell interfaces ∂K_j, we obtain the system of ODEs for $i \in \{1, \ldots, d+2\}$ and $n \in \{1, \ldots, M_i\}$:

$$\begin{aligned} \frac{\mathrm{d}a_n^i}{\mathrm{d}t} &= \sum_{j=1}^{N_K} \left(\int_{K_j} \mathbf{F}_i(\widetilde{\mathbf{q}}_h) \cdot \nabla \Phi_n^i + s_i(\widetilde{\mathbf{q}}_h)\Phi_n^i \,\mathrm{d}\mathbf{x} - \int_{\partial K_j} \widehat{F_i}(\widetilde{\mathbf{q}}_h^-, \widetilde{\mathbf{q}}_h^+, \mathbf{n})\Phi_n^i \,\mathrm{d}\boldsymbol{\sigma} \right) \\ &= \int_{\bigcup_{j=1}^{N_K} K_j} \mathbf{F}_i(\widetilde{\mathbf{q}}_h) \cdot \nabla \Phi_n^i + s_i(\widetilde{\mathbf{q}}_h)\Phi_n^i \,\mathrm{d}\mathbf{x} - \int_{\bigcup_{j=1}^{N_K} \partial K_j} \widehat{F_i}(\widetilde{\mathbf{q}}_h^-, \widetilde{\mathbf{q}}_h^+, \mathbf{n})\Phi_n^i \,\mathrm{d}\boldsymbol{\sigma}, \end{aligned} \tag{IV.8}$$

where the orthonormality of the basis functions has been used. Compared to the standard Galerkin projection presented in Section I.3.2.1, the additional numerical flux allows, on the one hand, to enforce in a weak sense the boundary conditions and, on the other hand, to introduce numerical diffusion/dissipation through, for example, an upwind convection flux accounting for the flow direction in order to stabilize the ROM. In addition, the resulting scheme is consistent since the numerical flux is consistent according to Lemma 1.

IV.3.3.2 Hyper-reduction

In order to evaluate the volume integral in system (IV.8), we have to compute the integrand

$$H^v_{i,n}[\widetilde{\mathbf{q}}_h] = \mathbf{F}_i(\widetilde{\mathbf{q}}_h) \cdot \nabla \Phi^i_n + s_i(\widetilde{\mathbf{q}}_h)\Phi^i_n$$

at each point $\mathbf{x} \in \bigcup_{j=1}^{N_K} K_j$, which is prohibitively computationally expensive. To resolve this issue, the integrands that are polynomial with respect to $\widetilde{\mathbf{q}}_h$ (e.g. $f_1 = \rho u$ in (IV.2)) are computed by the precomputed-based approach described in Section I.5.1. Alternatively, the ECSW method is employed for the evaluation of the non-polynomial integrands (e.g. $f_2 = \rho u^2 + p$ and $f_3 = u(E+p)$ in (IV.2)), where the pre-computation-based approach is not applicable. In this method, the integrands are evaluated at only L_v points $\widetilde{\mathbf{x}} \in \bigcup_{j=1}^{N_K} K_j$ to ensure the computational complexity of the ROM does not scale with the size of the mesh ($L_v \ll N_{\mathbf{x}}$). The volume integral is then approximated by the following empirical quadrature rule

$$\int_{\bigcup_{j=1}^{N_K} K_j} H^v_{i,n}[\widetilde{\mathbf{q}}_h](\mathbf{x})\,\mathrm{d}\mathbf{x} \approx \sum_{l=1}^{L_v} \widetilde{\omega}^v_l H^v_{i,n}[\widetilde{\mathbf{q}}_h](\widetilde{\mathbf{x}}_l),$$

where $\widetilde{\mathbf{x}}_l$ and $\widetilde{\omega}^v_l > 0$ denote the quadrature points and weights, respectively. Theses empirical quadrature points and weights are determined simultaneously during the training stage to best approximate the exact quadrature rule:

$$\underbrace{\begin{pmatrix} H^v_{i,1}[\mathbf{s}^l_h](\mathbf{x}_1) & \cdots & H^v_{i,1}[\mathbf{s}^l_h](\mathbf{x}_{N_{\mathbf{x}}}) \\ \vdots & & \vdots \\ H^v_{i,M_i}[\mathbf{s}^l_h](\mathbf{x}_1) & \cdots & H^v_{i,M_i}[\mathbf{s}^l_h](\mathbf{x}_{N_{\mathbf{x}}}) \end{pmatrix}}_{\mathbf{H}^{(i)}[\mathbf{s}^l_h]} \underbrace{\begin{pmatrix} \omega^v_1 \\ \vdots \\ \omega^v_{N_{\mathbf{x}}} \end{pmatrix}}_{\boldsymbol{\omega}} \approx \underbrace{\begin{pmatrix} \int_{\bigcup_{j=1}^{N_K} K_j} H^v_{i,1}[\mathbf{s}^l_h](\mathbf{x})\,\mathrm{d}\mathbf{x} \\ \vdots \\ \int_{\bigcup_{j=1}^{N_K} K_j} H^v_{i,M_i}[\mathbf{s}^l_h](\mathbf{x})\,\mathrm{d}\mathbf{x} \end{pmatrix}}_{\mathbf{c}_i[\mathbf{s}^l_h]},$$

where the approximate solution $\widetilde{\mathbf{q}}_h$ in $H^v_{i,n}$ is replaced by the snapshot $\mathbf{s}^l_h$ of the conserved variables collected during the training stage. Combining the contributions

of all the integrands $H^v_{i,n}$ leads to the approximation problem

$$\underbrace{\begin{pmatrix} \mathbf{H}^{(1)}[\mathbf{s}_h^1] \\ \mathbf{H}^{(1)}[\mathbf{s}_h^2] \\ \vdots \\ \mathbf{H}^{(2)}[\mathbf{s}_h^1] \\ \vdots \\ \mathbf{H}^{(d+2)}[\mathbf{s}_h^{N_s}] \end{pmatrix}}_{\mathbf{G}} \underbrace{\begin{pmatrix} \omega_1^v \\ \vdots \\ \omega_{N_\mathbf{x}}^v \end{pmatrix}}_{\boldsymbol{\omega}} \approx \underbrace{\begin{pmatrix} \mathbf{c}_1[\mathbf{s}_h^1] \\ \mathbf{c}_1[\mathbf{s}_h^2] \\ \vdots \\ \mathbf{c}_2[\mathbf{s}_h^1] \\ \vdots \\ \mathbf{c}_{d+2}[\mathbf{s}_h^{N_s}] \end{pmatrix}}_{\mathbf{d}}.$$

In the ECSW method, $\boldsymbol{\omega}$ is the solution of the non-negative least-squares problem

$$\min_{\boldsymbol{\omega} \in \mathbb{R}_+^{N_\mathbf{x}}} \|\mathbf{G}\boldsymbol{\omega} - \mathbf{d}\|_2^2, \tag{IV.9}$$

which is solved by the algorithm described in [70]. This algorithm promotes sparsity in the solution and terminates when the stopping criterion $\|\mathbf{G}\boldsymbol{\omega} - \mathbf{d}\|_2 \leqslant \epsilon \|\mathbf{d}\|_2$ is satisfied for a given level of hyper-reduction accuracy ϵ. The weights $\widetilde{\omega}_l^v$ are finally obtained by keeping only the nonzero components of the solution to problem (IV.9), and the points $\widetilde{\mathbf{x}}_l$ are the points associated with these weights $\widetilde{\omega}_l^v$.

The discontinuous Galerkin ROM (IV.8) then becomes

$$\frac{\mathrm{d}a_n^i}{\mathrm{d}t} = \sum_{l=1}^{L_v} \widetilde{\omega}_l^v H_{i,n}^v[\widetilde{\mathbf{q}}_h](\widetilde{\mathbf{x}}_l) - \int_{\bigcup_{j=1}^{N_K} \partial K_j} \widehat{F_i}(\widetilde{\mathbf{q}}_h^-, \widetilde{\mathbf{q}}_h^+, \mathbf{n}) \Phi_n^i \, \mathrm{d}\boldsymbol{\sigma}.$$

In the same way, the ECSW method is also employed for the evaluation of the surface integrals by defining

$$H_{i,n}^s[\widetilde{\mathbf{q}}_h] = \widehat{F_i}(\widetilde{\mathbf{q}}_h^-, \widetilde{\mathbf{q}}_h^+, \mathbf{n}) \Phi_n^i$$

for $\boldsymbol{\sigma} \in \bigcup_{j=1}^{N_K} \partial K_j$. Finally, the hyper-reduced discontinuous Galerkin ROM becomes

$$\frac{\mathrm{d}a_n^i}{\mathrm{d}t} = \sum_{l=1}^{L_v} \widetilde{\omega}_l^v H_{i,n}^v[\widetilde{\mathbf{q}}_h](\widetilde{\mathbf{x}}_l) - \sum_{l=1}^{L_s} \widetilde{\omega}_l^s H_{i,n}^s[\widetilde{\mathbf{q}}_h](\widetilde{\boldsymbol{\sigma}}_l) \tag{IV.10}$$

for $i \in \{1, \ldots, d+2\}$ and $n \in \{1, \ldots, M_i\}$.

IV.3.3.3 Time discretization

The ROM is discretized in time by the same explicit TVD Runge-Kutta scheme [100] used in the HDM. Writing the system of ODEs (IV.10) as

$$\frac{\mathrm{d}\mathbf{a}}{\mathrm{d}t} = \mathbf{L}(\mathbf{a}(t; \boldsymbol{\mu}), t; \boldsymbol{\mu}),$$

where the vector $\mathbf{a}$ contains the reduced coordinates a_n^i for $i \in \{1, \ldots, d+2\}$ and $n \in \{1, \ldots, M_i\}$, the second-order scheme reads

$$\begin{aligned}\mathbf{a}^{(1)} &= \mathbf{a}(t_k;\boldsymbol{\mu}) + \Delta t \mathbf{L}(\mathbf{a}(t_k;\boldsymbol{\mu}), t_k;\boldsymbol{\mu}),\\ \mathbf{a}(t_{k+1};\boldsymbol{\mu}) &= \frac{1}{2}\mathbf{a}(t_k;\boldsymbol{\mu}) + \frac{1}{2}\mathbf{a}^{(1)} + \frac{\Delta t}{2}\mathbf{L}(\mathbf{a}^{(1)}, t_{k+1};\boldsymbol{\mu}),\end{aligned}$$

and the third-order scheme reads

$$\begin{aligned}\mathbf{a}^{(1)} &= \mathbf{a}(t_k;\boldsymbol{\mu}) + \Delta t \mathbf{L}(\mathbf{a}(t_k;\boldsymbol{\mu}), t_k;\boldsymbol{\mu}),\\ \mathbf{a}^{(2)} &= \frac{3}{4}\mathbf{a}(t_k;\boldsymbol{\mu}) + \frac{1}{4}\mathbf{a}^{(1)} + \frac{\Delta t}{4}\mathbf{L}(\mathbf{a}^{(1)}, t_{k+1};\boldsymbol{\mu}),\\ \mathbf{a}(t_{k+1};\boldsymbol{\mu}) &= \frac{1}{3}\mathbf{a}(t_k;\boldsymbol{\mu}) + \frac{2}{3}\mathbf{a}^{(2)} + \frac{2\Delta t}{3}\mathbf{L}(\mathbf{a}^{(2)}, t_{k+\frac{1}{2}};\boldsymbol{\mu}).\end{aligned}$$

The initial solution $\mathbf{a}(t_0;\boldsymbol{\mu})$ of these systems is given by the orthogonal projection of the initial condition $\mathbf{q}_0(\mathbf{x};\boldsymbol{\mu})$ onto the affine trial subspace:

$$a_n^i(t_0;\boldsymbol{\mu}) = \int_\Omega \left(q_0^i(\mathbf{x};\boldsymbol{\mu}) - q_o^i(\mathbf{x})\right) \Phi_n^i(\mathbf{x})\, \mathrm{d}\mathbf{x}.$$

IV.4 Discontinuous Galerkin domain decomposition method

Given the ROM based on the discontinuous Galerkin method developed in the previous section, the domain decomposition is applied in a straightforward manner since the coupling between the HDM and the ROM is performed through the numerical flux.

IV.4.1 Domain decomposition

In the HDM, the domain is divided into N_K micro-cells K_j as illustrated by Figure IV.2. In the standard global MOR approach, these micro-cells are generally agglomerated into a single macro-cell Ω. Thanks to the ROM developed in Section IV.3, we formulate a spatially local approach for the case of several non-overlapping micro- and macro-cells, as illustrated in Figure IV.2. The HDM is employed in the micro-cells K_j, while the ROM is used in the macro-cells Ω_j. The domain is spatially decomposed into smooth and non-smooth regions in order to isolate shocks or compact support phenomena. For simplicity, the partitioning is based in this work on *a priori* knowledge of the solution, and we anticipate the subdomains representable via POD.

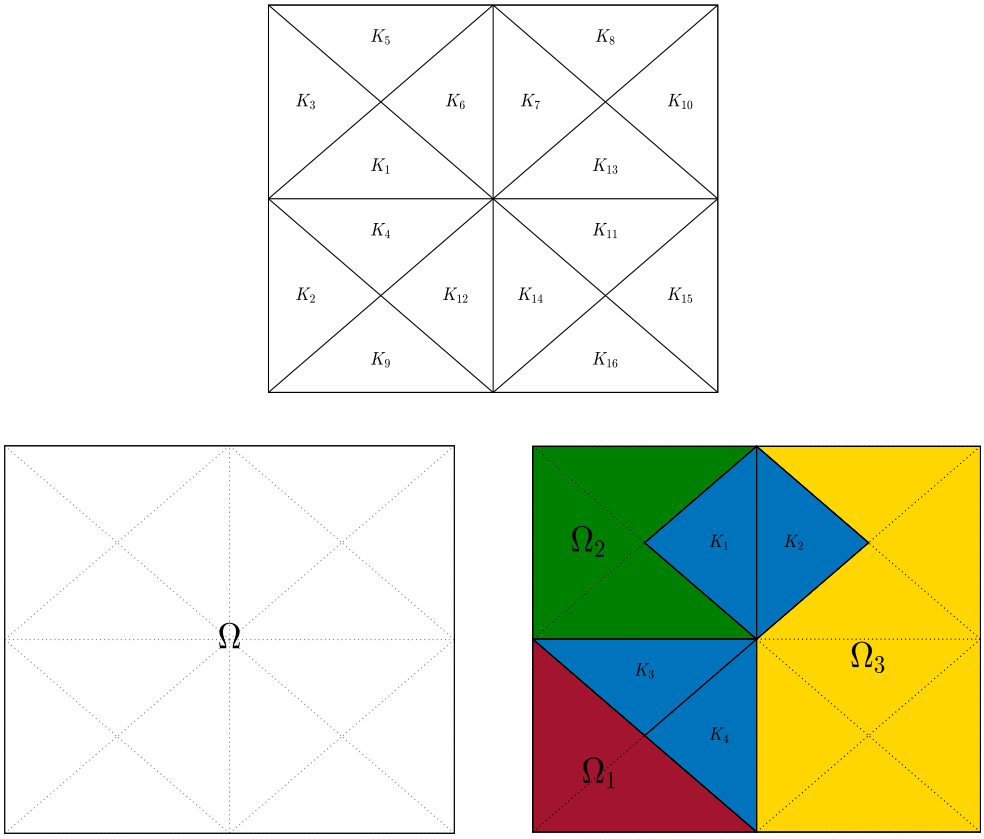


Figure IV.2: Top: example of mesh employed in the HDM containing 16 micro-cells K_j. Bottom: examples of domain decomposition. On the left, the domain is agglomerated into a single macro-cell. On the right, the domain is divided into 4 micro-cells and 3 macro-cells.

IV.4.2 Coupling between the HDM and ROMs

The restriction of the approximate solution to each macro-cell Ω_j is written as

$$\forall \mathbf{x} \in \Omega_j : \ \widetilde{q}_h^i(\mathbf{x}, t; \boldsymbol{\mu}) = q_o^{i,j}(\mathbf{x}) + \sum_{n=1}^{M_{i,j}} a_n^{i,j}(t; \boldsymbol{\mu}) \Phi_n^{i,j}(\mathbf{x}), \tag{IV.11}$$

where we proceed exactly as described in Section IV.3 for the training and prediction stages. The discontinuous Galerkin ROM (IV.10) now becomes

$$\frac{\mathrm{d}a_n^{i,j}}{\mathrm{d}t} = \sum_{l=1}^{L_v^j} \widetilde{\omega}_{j,l}^v H_{i,j,n}^v[\widetilde{\mathbf{q}}_h](\widetilde{\mathbf{x}}_{j,l}) - \sum_{l=1}^{L_s^j} \widetilde{\omega}_{j,l}^s H_{i,j,n}^s[\widetilde{\mathbf{q}}_h](\widetilde{\boldsymbol{\sigma}}_{j,l}), \tag{IV.12}$$

where $H_{i,j,n}^v[\widetilde{\mathbf{q}}] = \mathbf{F}_i(\widetilde{\mathbf{q}}) \cdot \nabla \Phi_n^{i,j} + s_i(\widetilde{\mathbf{q}}) \Phi_n^{i,j}$ and $H_{i,j,n}^s[\widetilde{\mathbf{q}}_h] = \widehat{F_i}(\widetilde{\mathbf{q}}_h^-, \widetilde{\mathbf{q}}_h^+, \mathbf{n}) \Phi_n^{i,j}$. In this way, the global solution is recovered by linking the local solutions at the interface between micro- and macro-cells through the numerical flux.

IV.5 Applications

The performance of the DGDD method is evaluated for three applications based on the quasi-1D and 2D Euler equations. In each case, the accuracy of the ROM with respect to the HDM is evaluated using the relative approximation error in the predicted Mach number:

$$\text{Error} = \sqrt{\frac{\displaystyle\int_{t_0}^{t_{max}} \int_{\Omega} \left\| \mathbf{M}_{hdm} - \mathbf{M}_{rom} \right\|_2^2 \mathrm{d}\mathbf{x}\, \mathrm{d}t}{\displaystyle\int_{t_0}^{t_{max}} \int_{\Omega} \left\| \mathbf{M}_{hdm} \right\|_2^2 \mathrm{d}\mathbf{x}\, \mathrm{d}t}} \times 100\%,$$

where $\mathbf{M} = \mathbf{u}/c$ denotes the Mach number and $c = \sqrt{\gamma p/\rho}$ denotes the speed of sound. Furthermore, the computational speedup of the ROM with respect to the HDM is evaluated in each case in order to quantify the reduction in computational cost provided by the ROM based on the proposed DGDD method.

IV.5.1 Reproduction of an isentropic vortex

The first application seeks to validate the DGDD method on a reproductive test case where the predicted ROM solution is obtained at the same input parameter used in the training stage. We consider an isentropic vortex for $\mathbf{x} \in [0, 12] \times [-2.5, 2.5]$ and $t \in [0, 7]$. The initial and boundary conditions are supplied by the exact solution to the 2D Euler equations (IV.3):

$$\begin{cases} \rho = \left(1 - \left(\dfrac{\gamma - 1}{16\gamma\pi^2}\right) \beta^2 e^{2(1-r^2)}\right)^{\frac{1}{\gamma-1}} \\ u = 1 - \beta e^{1-r^2} \dfrac{y - y_0}{2\pi} \\ v = \beta e^{1-r^2} \dfrac{x - x_0}{2\pi} \\ p = \rho^{\gamma} \end{cases}$$

with $r = \sqrt{(x - t - x_0)^2 + (y - y_0)^2}$, $x_0 = 2.5$, $y_0 = 0$ and $\beta = 5$.

The HDM is constructed by discretizing the 2D Euler equations (IV.3) using a third-order discontinuous Galerkin method with the local Lax-Friedrichs flux in space and the third-order TVD Runge-Kutta method in time. The domain is discretized using $N_K = 960$ triangular micro-cells, and the time-step size is $\Delta t = 0.01$. We compare the approximate solutions computed using two ROMs: the first one is a global ROM where the micro-cells are agglomerated into a single macro-cell (i.e. no domain decomposition), and the second one is a local ROM where the domain is divided randomly into 8 contiguous macro-cells Ω_j shown in Figure IV.3.

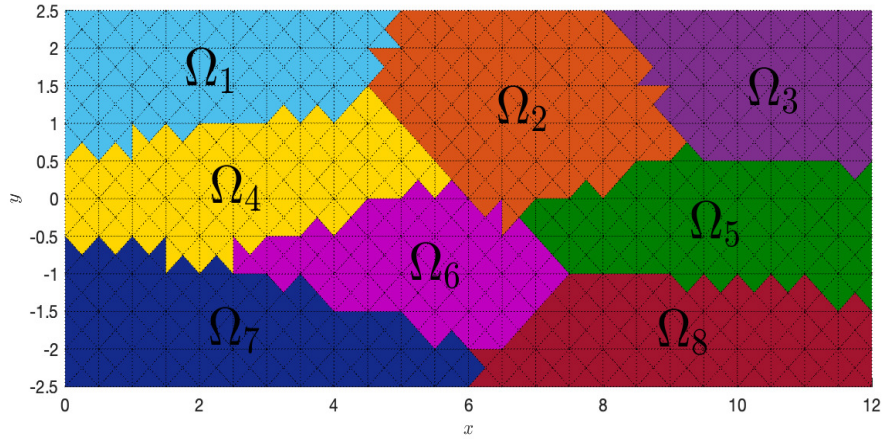


Figure IV.3: Decomposition of the domain into 8 macro-cells for the reproduction of an isentropic vortex.

For this example, no hyper-reduction is used to compare directly the errors introduced by the discontinuous Galerkin formulation of the ROM. Snapshots of the high-fidelity solution are collected every time-step for the construction of the basis functions. Figure IV.4 shows snapshots of the Mach number solution predicted by the local ROM at different time instances.

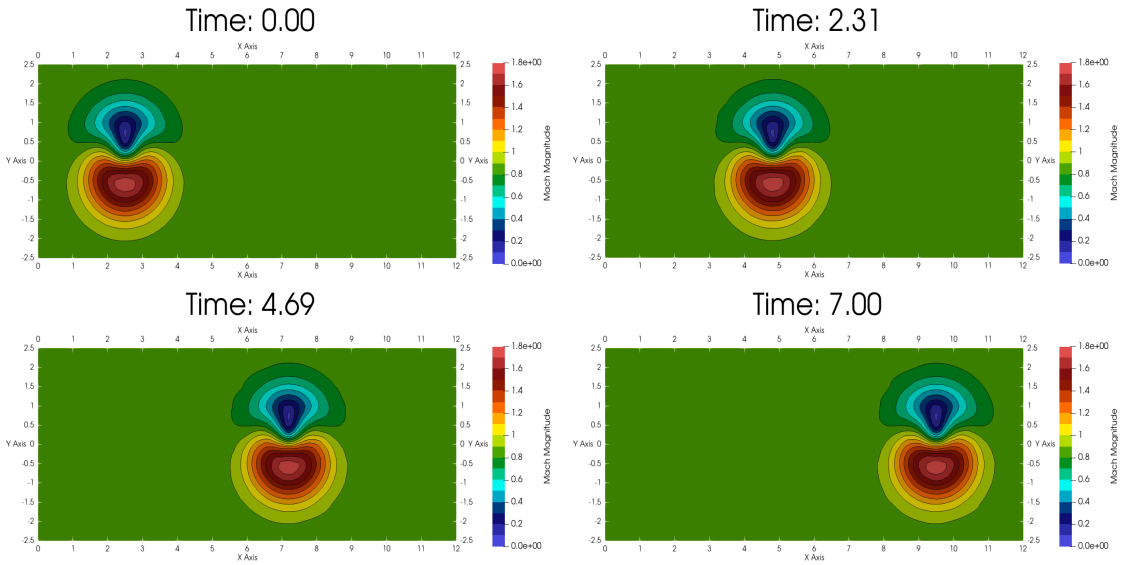


Figure IV.4: Snapshots of the Mach number solution for the reproduction of an isentropic vortex with 8 macro-cells and $M = 15$ basis functions in each macro-cell, as computed using the DGDD-based ROM. The isolines of the corresponding high-fidelity solution are plotted in black.

In Figure IV.5, we compare the error of the global and local ROMs as a function of the number of basis functions M. Here, the number of basis functions is the

same for all macro-cells and components of the solution, i.e. $\forall i, j : \ M_{i,j} = M$. The error of the global and local ROMs tends to decrease as M increases, even though the convergence behaviour is not necessarily monotonic. Moreover, the error of the ROMs is close to the L^2-projection error of individual state, and the local ROM is more accurate than the global ROM, which validates the proposed DGDD approach.

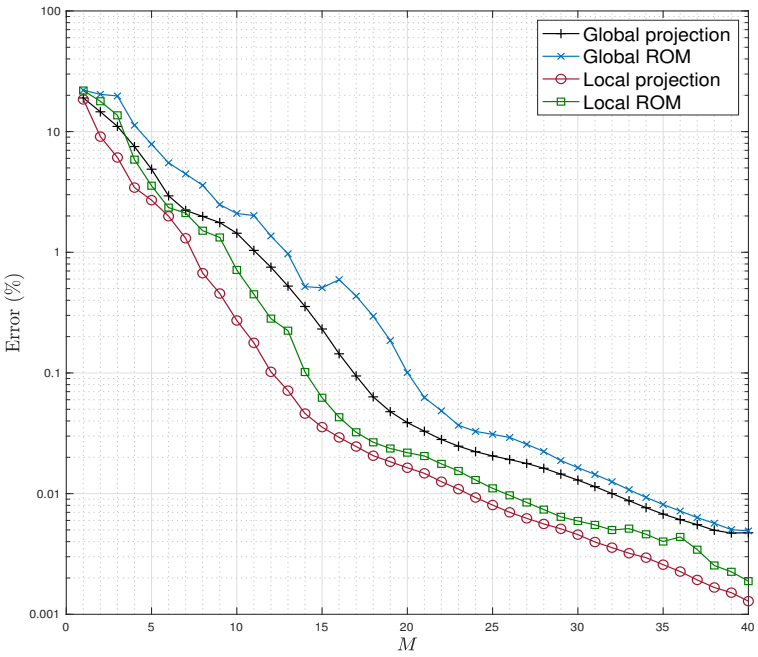


Figure IV.5: Accuracy of the global and local approches for the reproduction of an isentropic vortex.

IV.5.2 Prediction of a transonic flow in a converging-diverging nozzle

The second application considers the prediction of a transonic flow in a converging-diverging nozzle for $x \in [0, 1]$ and $t \in [0, 5]$. The cross sectional area of the nozzle

$$A(x) = \frac{1}{0.5 + 1.3x} \left(\frac{2 + (\gamma - 1)(0.5 + 1.3x)^2}{1 + \gamma} \right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

is illustrated in Figure IV.6.

The steady state solution is determined by the total pressure P_{tot} and total temperature T_{tot} at the inlet and by the pressure p_{out} at the outlet. In this example, the total temperature at the inlet is fixed to $T_{tot} = 1$, and the input parameters $\boldsymbol{\mu} = (P_{tot}, x_s)$ are the inlet total pressure P_{tot} and the position of the shock wave x_s, which is a function of P_{tot} and p_{out}. We consider the unsteady problem, starting

from the initial solution given in Figure IV.7 and with time-dependent boundary conditions that move the position of the shock wave as follows

$$P_{tot}^{bc}(t;\boldsymbol{\mu}) = \begin{cases} 1 + \frac{P_{tot}-1}{0.1}t & \text{if } t < 0.1 \\ P_{tot} & \text{else,} \end{cases} \qquad x_s^{bc}(t;\boldsymbol{\mu}) = \begin{cases} 0.7 + \frac{x_s-0.7}{0.1}t & \text{if } t < 0.1 \\ x_s & \text{else.} \end{cases}$$

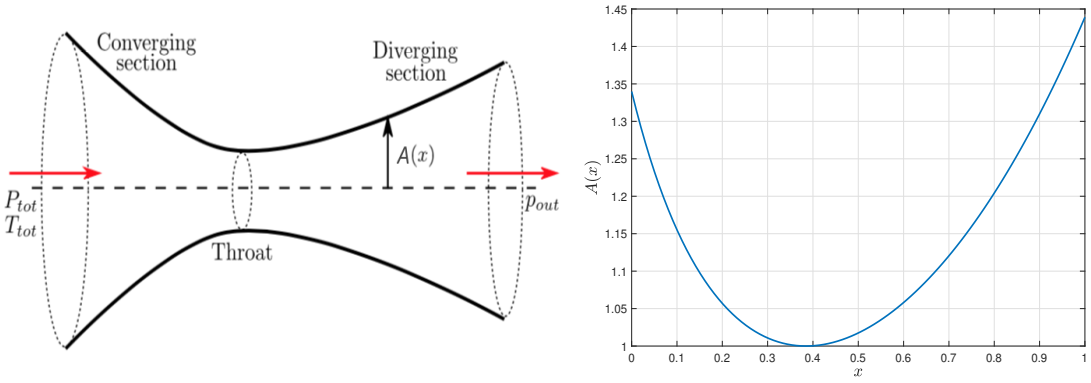


Figure IV.6: On the left, converging-diverging nozzle [1]. On the right, cross sectional area profile of the nozzle for this application.

The quasi-1D Euler equations (IV.2) are discretized using a second-order Discontinuous Galerkin method equipped with the local Lax-Friedrichs flux and the minmod limiter in space and the second-order TVD Runge-Kutta scheme in time. The domain is discretized using $N_K = 500$ micro-cells and the time-step size is $\Delta t = 0.0008$. Domain decomposition is performed from *a priori* knowledge of the solution by dividing the physical domain into three regions shown in Figure IV.7. In non-shocked regions 1 and 3, the micro-cells are agglomerated into a single macro-cell for each region, and spatially local ROMs are employed. Region 2 consists of 100 micro-cells where the HDM is used in order to accurately capture the moving shock wave.

Figure IV.8 illustrates the parameter domain of interest chosen in order to place the shock wave in the interval $x_s \in [0.61, 0.79]$. It also shows the sampled input parameters used to build the snapshot database in the training stage. Note that the training input parameters corresponding to $\boldsymbol{\mu} = (P_{tot}, 0.7)$ have been removed from the original sampling since they correspond to the initial solution and are already in the snapshot database.

For each unsteady simulation corresponding to a sampled input parameter, we collect one snapshot every 5 time-steps from the HDM simulation. The squared singular values of the snapshot matrix for each of the conservative variables are shown in Figure IV.9. The squared singular values decrease rapidly, and 2 basis functions are sufficient to obtain a relative squared projection error lower than 0.001% for all variables in each macro-cell.

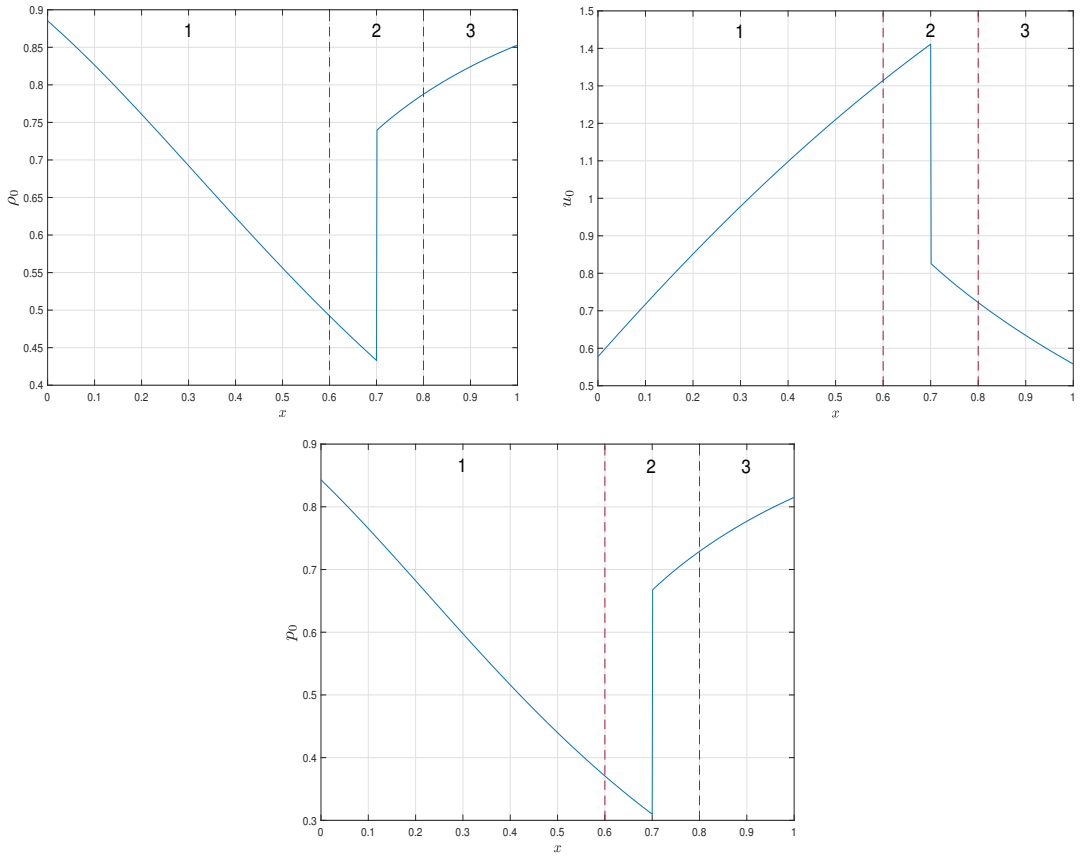


Figure IV.7: Initial condition corresponding to the steady state solution for $P_{tot} = 1$ and $x_s = 0.7$. The domain is divided into 2 macro-cells, denoted by regions 1 and 3, and 100 micro-cells in region 2.

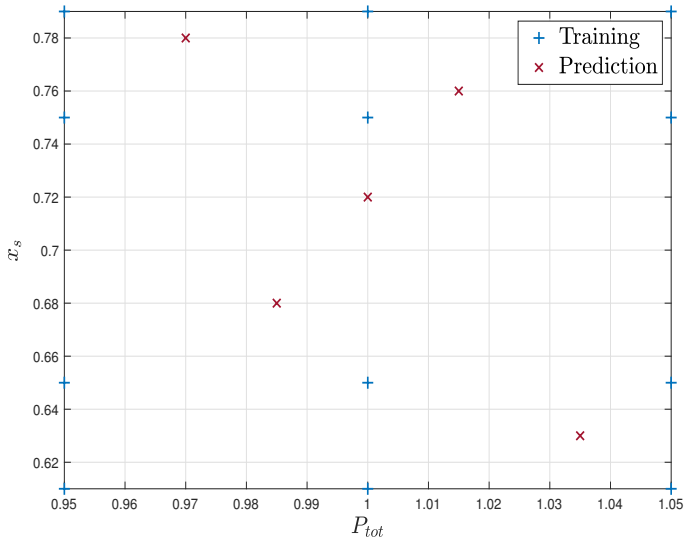


Figure IV.8: Input parameter sampling used during the training and prediction stage for the converging-diverging nozzle problem.

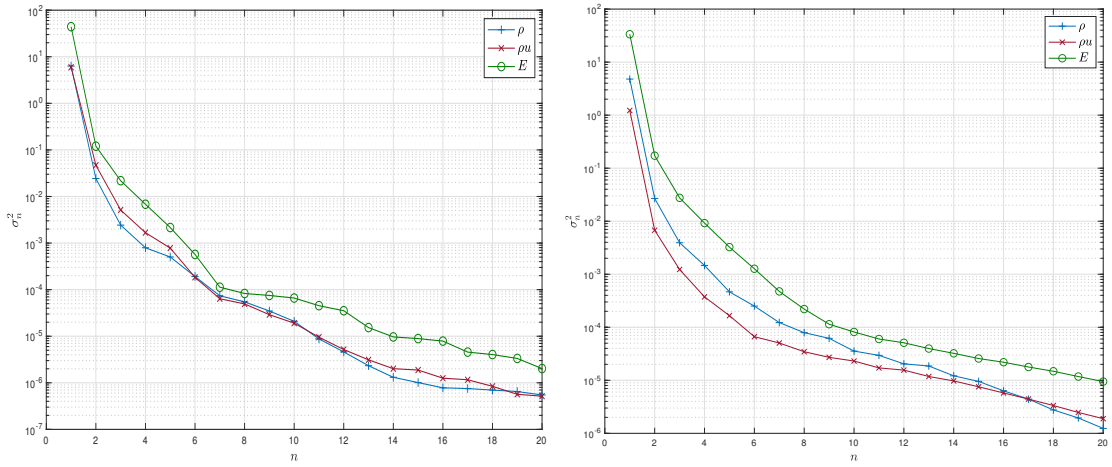


Figure IV.9: Squared singular values of the snapshot matrix corresponding to region 1 on the left and region 3 on the right for the converging-diverging nozzle problem.

The hyper-reduction training tolerance for this problem is chosen by a trial and error approach as $\epsilon = 10^{-5}$ in regions 1 and 3, which is sufficient to yield an accurate approximation of the integrals in the prediction stage. In region 1, the ECSW procedure identifies $L_v^1 = 23$ (resp. $L_s^1 = 9$) points $\widetilde{x}_l$ (resp. $\widetilde{\sigma}_l$) among the 900 (resp. 301) quadrature points to evaluate the volume (resp. surface) integrals. In region 3, the ECSW procedure identifies 49 (resp. 33) points $\widetilde{x}_l$ (resp. $\widetilde{\sigma}_l$) among the 300 (resp. 101) quadratures points to evaluate the volume (resp. surface) integrals. The reduced mesh delivered by the ECSW method is displayed in Figure IV.10. Notably, the ECSW procedure identifies more points in region 3, where a wave is moving at the beginning of the simulations (see Figure IV.11), than in region 1, where the flow solution is more amenable to a low-dimensional representation.

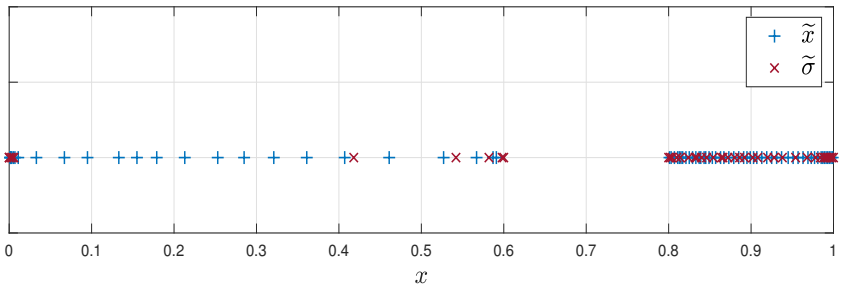


Figure IV.10: Quadrature points delivered by the ECSW method to approximate the volume (blue) and surface (red) integrals for $M = 8$.

Figures IV.11 and IV.12 show the pressure solutions obtained by the DGDD-based ROM for the different prediction tests denoted in Figure IV.8.

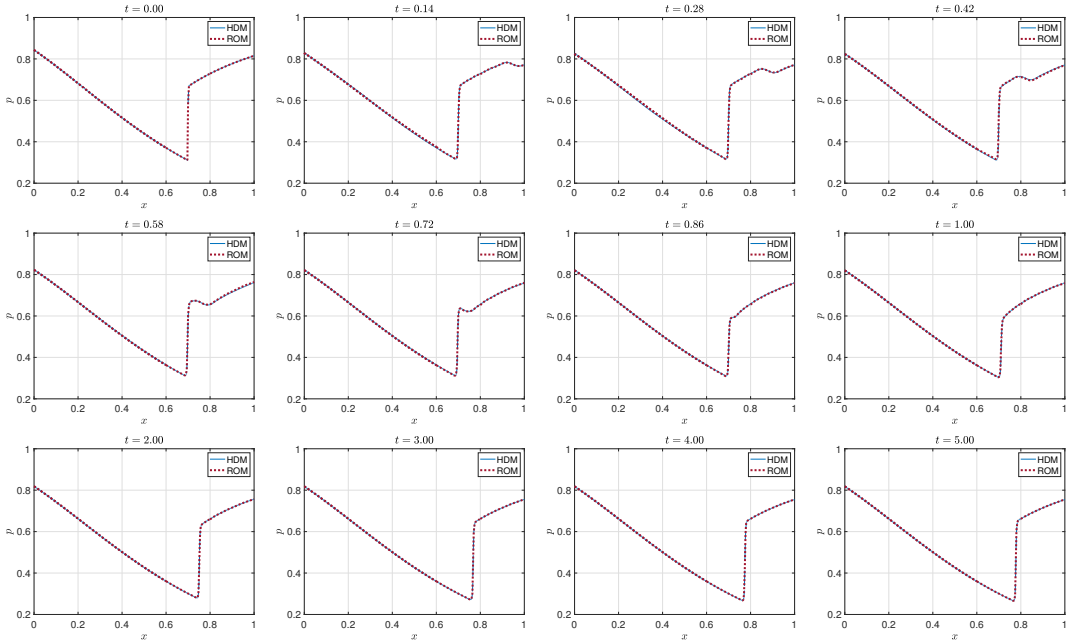
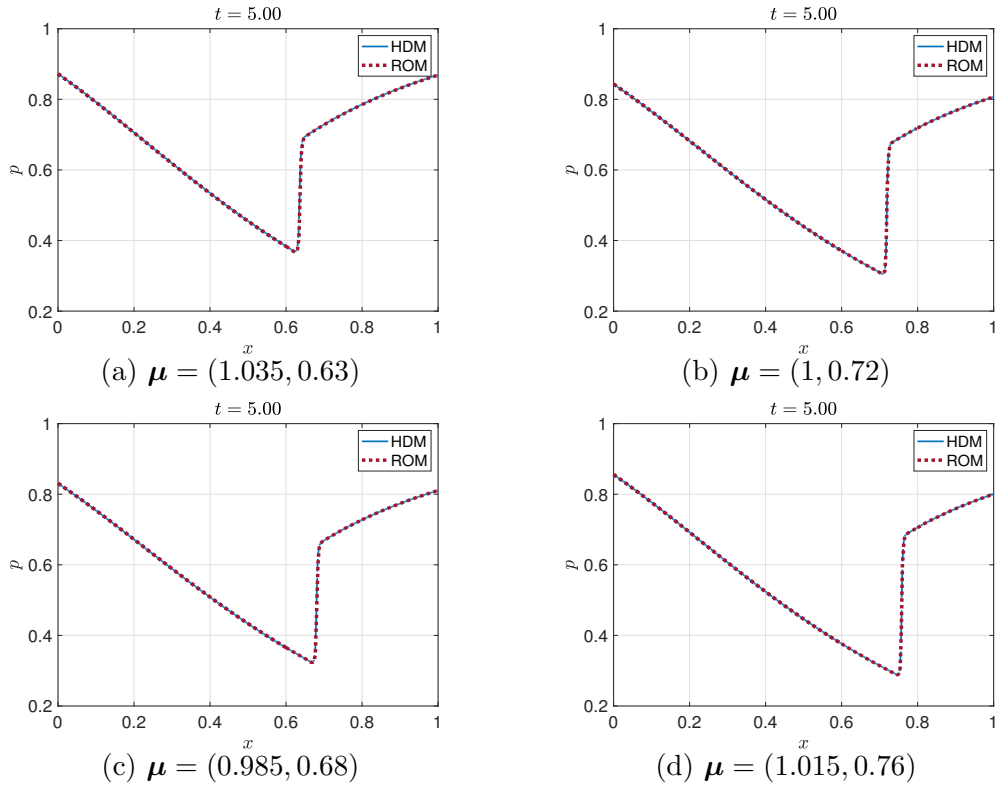


Figure IV.11: Computed pressure solution snapshots for the prediction test $\boldsymbol{\mu} = (0.97, 0.78)$ with $M = 8$ for the converging-diverging nozzle problem at different time instances.



(a) $\boldsymbol{\mu} = (1.035, 0.63)$ (b) $\boldsymbol{\mu} = (1, 0.72)$

(c) $\boldsymbol{\mu} = (0.985, 0.68)$ (d) $\boldsymbol{\mu} = (1.015, 0.76)$

Figure IV.12: Computed pressure solutions for the prediction tests at steady state with $M = 8$ for the converging-diverging nozzle problem.

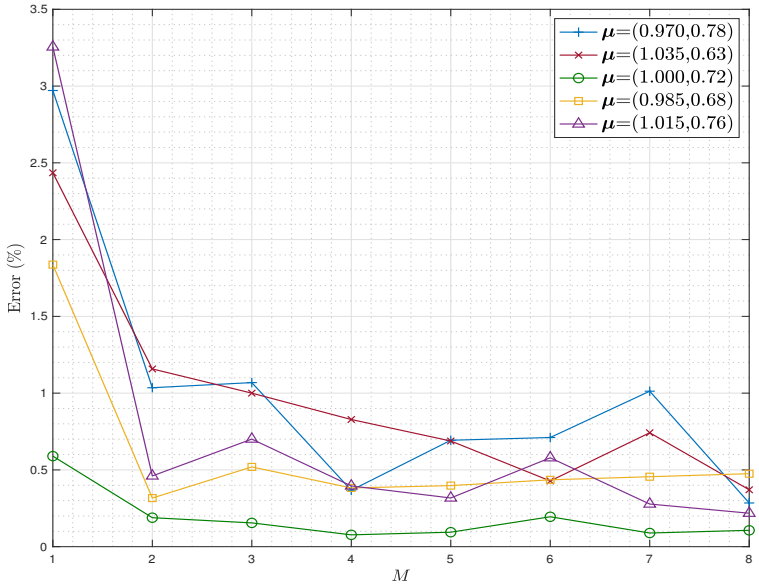


Figure IV.13: Error of the prediction tests for the converging-diverging nozzle problem as a function of the number of basis functions in region 3.

In Figure IV.13, we compare the error of the prediction tests as a function of the number of basis functions in region 3. The number of basis functions is chosen to be the same for all conservative variables, and the number of basis functions in region 1 is fixed to $M = 2$. As expected, the error tends to decrease when the number of basis functions M increases, and the error is less than 1% when $M = 4$ for all prediction tests. Notably, the error comes mainly from region 2, since a small perturbation of the shock wave position results in a large approximation error. For this small problem, using $M = 4$, the computational speedup factor for the solution of the ROM versus the HDM is 3.54. Of the time required for the solution of the DGDD-based ROM, 70.76% is spent in the computation of the HDM solution in region 2, and the remaining 29.24% is spent for the local ROMs in regions 1 and 3.

IV.5.3 Prediction of a transonic flow over a NACA 0012 airfoil

For the final application, we consider a 2D transonic flow over a NACA 0012 airfoil. We want to predict the flow solution at input parameters $\boldsymbol{\mu} = (M_\infty, \alpha)$ corresponding to different free-stream Mach numbers M_∞ and angles of attack α. The initial condition is a uniform flow at Mach M_∞

$$\forall \mathbf{x} \in \Omega : \ \rho_0(\mathbf{x}; \boldsymbol{\mu}) = \gamma, \ u_0(\mathbf{x}; \boldsymbol{\mu}) = M_\infty, \ v_0(\mathbf{x}; \boldsymbol{\mu}) = 0, \ p_0(\mathbf{x}; \boldsymbol{\mu}) = 1,$$

and the unsteady solution is computed for $t \in [0, 21]$. Slip boundary conditions are applied at the airfoil surface, and the far-field boundary condition is set to be a uniform flow at Mach M_∞.

The HDM is constructed by discretizing the 2D Euler equations (IV.3) using a second-order discontinuous Galerkin method equipped with the HLL flux and the Barth-Jespersen limiter in space and the second-order TVD Runge-Kutta scheme in time. The domain is discretized using $N_K = 4150$ triangular micro-cells, and the time-step size is $\Delta t = 0.003$. As shown in Figure IV.14, the domain is divided into two regions: the HDM is employed in the region near the airfoil to accurately capture the moving shock wave, while the ROM is used elsewhere since the solution is amenable to accurate low-dimensional representation in the parameter domain of interest.

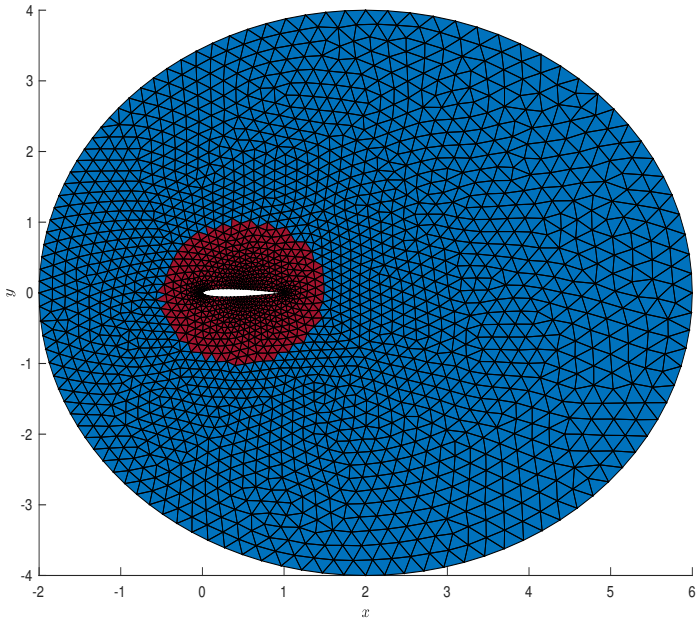


Figure IV.14: Decomposition of the domain in 1065 micro-cells (red) and one macro-cell (blue) for the transonic NACA airfoil problem.

In Figure IV.15, we plot the sampling of the parameter domain for the training and prediction stages. For each training HDM simulation, one snapshot is collected every 25 time-steps. A unique set of basis functions is constructed for each prediction input parameter. For each prediction input parameter, we apply POD on the snapshots corresponding to the four closest training input parameters to the prediction input parameter, defined by the square grid containing the predicted parameter value. For example, the basis functions for the queried input parameter corresponding to $M_\infty = 0.754$ and $\alpha = 0.2$ are computed using the snapshots from the simulations corresponding to $\boldsymbol{\mu} \in \{(0.75, 0), (0.76, 0), (0.75, 0.5), (0.76, 0.5))\}$. Figure IV.17 plots the decay of the squared singular values of the snapshot matrix for each of the conservative variables for this example. In this case, 6 basis functions are required to obtain a relative squared projection error of less than 0.001% for ρ, ρu and E. The momentum in the y-direction is close to zero, and the (absolute) squared projection error is below 0.3 with 6 basis functions for ρv.

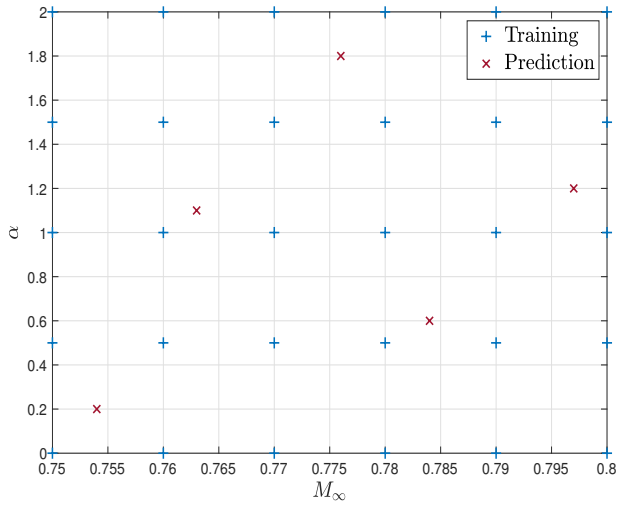


Figure IV.15: Input parameter sampling used during the training and prediction stage for the transonic NACA airfoil problem.

Lastly, the hyper-reduction training is performed using the tolerance $\epsilon = 10^{-4}$, which provides for sufficient accuracy in the prediction stage. For the prediction test corresponding to $\boldsymbol{\mu} = (0.754, 0.2)$, the ECSW procedure identifies $L_v = 403$ (resp. $L_s = 629$) points $\widetilde{\mathbf{x}}_l$ (resp. $\widetilde{\boldsymbol{\sigma}}_l$) among the 9255 (resp. 9314) quadrature points to evaluate the volume (resp. surface) integrals. Figure IV.16 shows the resulting reduced mesh. The quadrature points are notably located on the left side of the domain and in the airfoil's wake, where compression waves and shocks are propagating before the stationary solution is established. For the other prediction tests, the result of the ECSW procedure is similar.

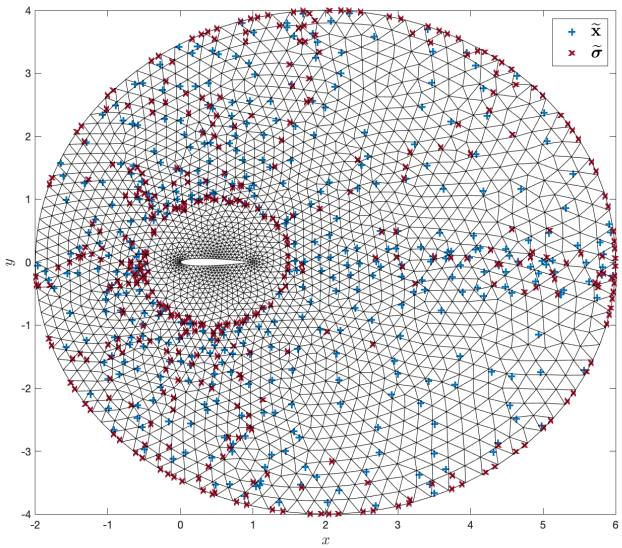


Figure IV.16: Quadrature points delivered by the ECSW method to approximate the volume (blue) and surface (red) integrals for $M = 16$.

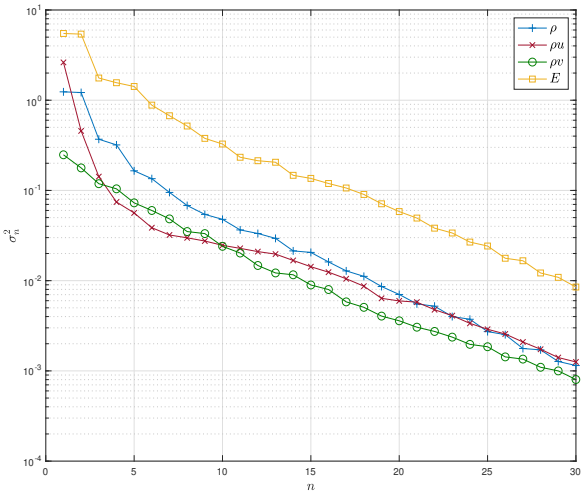


Figure IV.17: Squared singular values of the snapshot matrix corresponding to the prediction input parameter $\boldsymbol{\mu} = (0.754, 0.2)$ for the NACA airfoil problem.

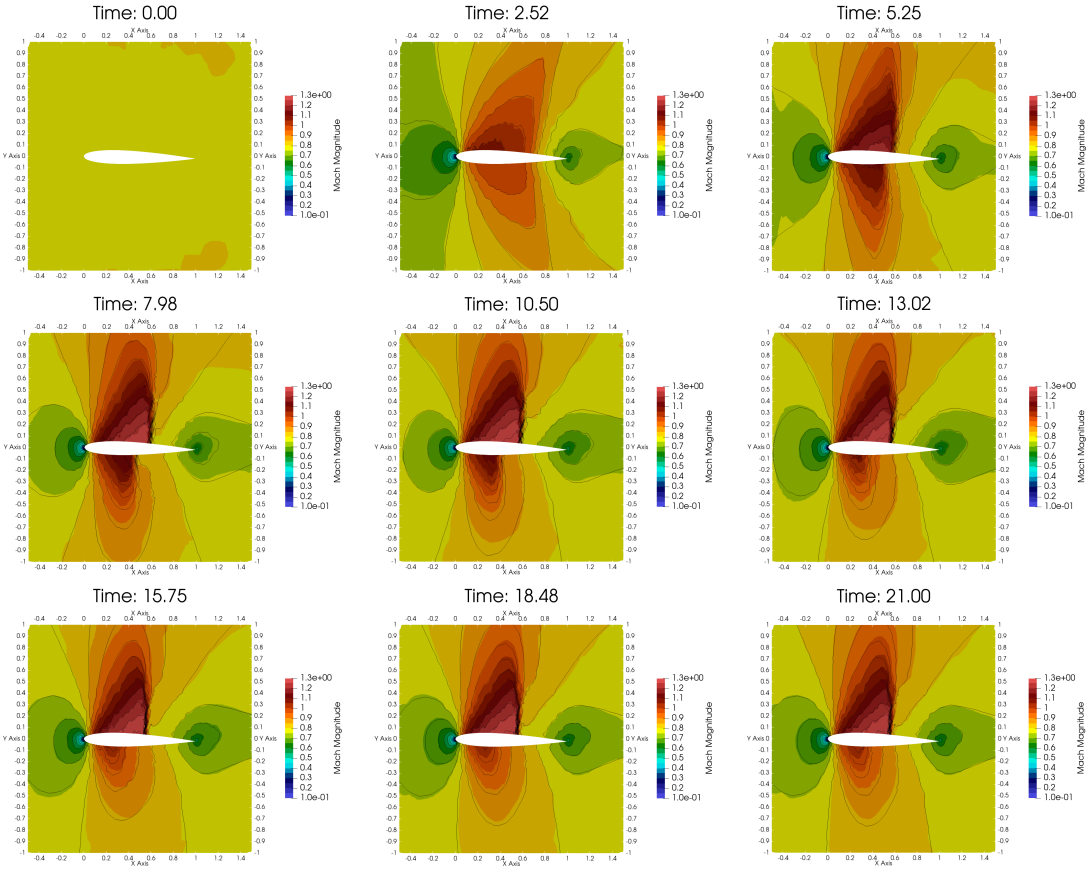
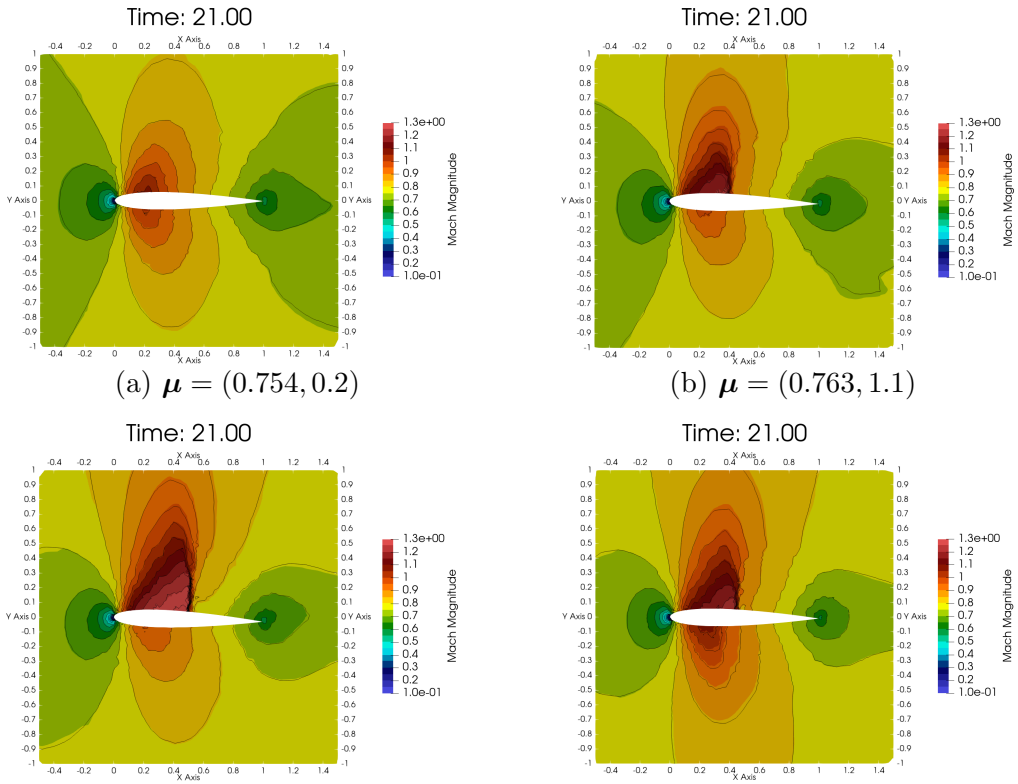


Figure IV.18: Mach number solution snapshots for the transonic NACA airfoil problem at different time instances computed using the ROM for the prediction test $\boldsymbol{\mu} = (0.797, 1.2)$ with $M = 16$. The isolines of the corresponding high-fidelity solution are plotted in black.



(a) $\boldsymbol{\mu} = (0.754, 0.2)$ (b) $\boldsymbol{\mu} = (0.763, 1.1)$

(c) $\boldsymbol{\mu} = (0.776, 1.8)$ (d) $\boldsymbol{\mu} = (0.784, 0.6)$

Figure IV.19: Mach number solution snapshots for the transonic NACA airfoil problem at steady state with $M = 16$. The isolines of the corresponding high-fidelity solution are plotted in black.

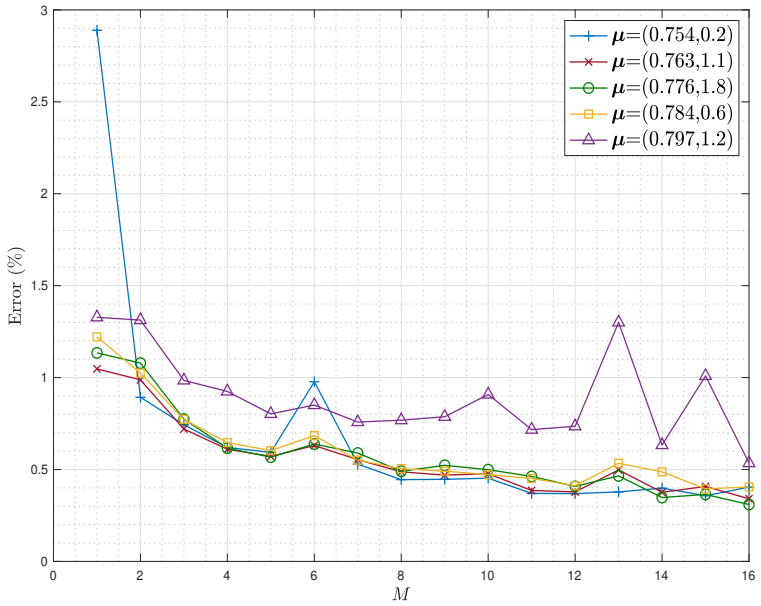


Figure IV.20: Error of the prediction tests for the transonic NACA airfoil problem as a function of the number of basis functions.

Figures IV.18 and IV.19 show snapshots of the computed Mach number using the DGDD-based ROM for the different prediction input parameters and for different times during the time-dependent flow simulations.

In Figure IV.20, we plot the space-time error depending on the number of basis functions for each of the prediction tests. The number of basis functions is again taken to be the same for all variables. It can be observed that the approximation error for the DGDD-based ROM is low even when using a small number of basis functions. When $M = 7$, the approximation error is less than 1% for all prediction tests. As the number of basis functions M increases, the approximation error decreases slowly, which is symptomatic of the slow singular value decay demonstrated in Figure IV.17.

With $M = 7$, the computational speedup factor delivered by the ROM over the HDM simulation is 4.54. Of the time required for the solution of the DGDD-based ROM, 94.41% comes from the micro-cell solution using the HDM while the remaining 5.59% comes from the single ROM macro-cell.

IV.6 Conclusion

In this work, we have presented a discontinuous Galerkin domain decomposition (DGDD) method for model order reduction. In this approach, the ROM approximates the solution in regions where significant dimensionality reduction can be achieved while the HDM is employed elsewhere. Notably, the discontinuous Galerkin formulation for the ROM offers a simple way to perform the coupling between the HDM and ROMs since the global solution is recovered by linking the local solutions at the interface between subdomains through the numerical fluxes. Compared to the standard DG method, the polynomial shape functions have been replaced by POD modes constructed during the training stage in order to best approximate the solution snapshots. In addition, the ROM has been equipped with hyper-reduction techniques such as the ECSW method, which is particularly well suited to approximate the volume and surface integrals involved in the DG formulation.

ROMs based on the proposed DGDD framework have been evaluated for parametric problems governed by the quasi-1D and 2D Euler equations. We have validated the DGDD method on the reproduction of an isentropic vortex. We have then investigated the prediction of unsteady flows in a converging-diverging nozzle and over a NACA 0012 airfoil. The results demonstrate the accuracy of the method, capable of delivering less than 1% of error over a range of predictive input parameters, and the significant reduction (approximately 78%) of the required computation time for the ROM simulations versus the associated HDM.

Conclusions and perspectives

In many industrial applications, efficient simulations are required, either due to runtime constraints in the case of extremely large-scale HDMs or due to the large number of simulations to perform for different input parameters in the case of many-query problems. For this reason, we have been interested during this thesis in significantly reducing the computational cost associated with numerical simulations of parametric problems governed by partial differential equations. To this end, we have considered ROMs, which typically consist of a training stage, in which high-fidelity solution snapshots are collected to define a low-dimensional trial subspace, and a prediction stage, where this data-driven trial subspace is then exploited in order to achieve fast or real-time simulations for new input parameters.

The first contribution of this thesis concerns the development of a new reduced-order approximation of the Boltzmann-BGK equation for the simulation of gas flows in both hydrodynamic and rarefied regimes. In this ROM, the distribution functions are represented in velocity space by a few basis functions in order to considerably reduce the number of degrees of freedom with respect to the HDM. The basis functions are constructed in the training stage by POD, and the approximate distribution functions are determined during the prediction stage by the Galerkin method. This approach has then been modified in order to preserve important properties of the HDM. In addition, we have derived the CFL condition ensuring a stable ROM in 1D.

The performance of the resulting ROM has been evaluated on the reproduction and prediction of unsteady flows containing shock waves, boundary layers and vortices in 1D and 2D. The results demonstrate the accuracy of the ROM (with less than 1% error) over a range of predictive input parameters and the significant computational speedup factor (approximately 45) delivered by the ROM with respect to the HDM simulations.

For future perspectives, we would present several interesting approaches in order to improve the ROM performance.

- *Residual minimization method.* The accuracy of the ROM could be improved by employing the residual minimisation method instead of the Galerkin method. To this end, the high-dimensional systems (II.12) and (II.13) can be projected onto the basis functions, as we proceeded in Section III.4.2, in

order to obtain the reduced-order system.

- *Hyper-reduction techniques.* The computational complexity of the ROM could be further improved by employing hyper-reduction techniques to compute the macroscopic state of the gas in equation (II.3). For this purpose, the project-and-approximate methods, presented in Sections I.5.2.2 and IV.3.3.2, are particularly well suited and relevant to approximate the integrals involved in the computation of the macroscopic state of the gas.

- *Preservation of the positivity property of the distribution functions.* The ROM could also be modified to ensure the $\mathcal{H}$ theorem [35] by enforcing the approximate distribution functions to be non-negative. However, this property may be prohibitively computationally expensive to preserve due to the large number of linear inequality constraints to be satisfied for all points of the velocity domain at each point of the physical domain and at each time-step.

- *Solution approximation.* In this work, the solution is approximated in velocity because the distribution functions are transported in velocity space over time. However, the solution could also be represented in velocity and physical spaces by a small number of basis functions:

$$\widetilde{f_h}(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \sum_{n=1}^{N_{pod}} a_n^f(t; \boldsymbol{\mu}) \Phi_n^f(\mathbf{x}, \boldsymbol{\xi}) \tag{IV.13}$$

  in order to further reduce the number of degrees of freedom. The ROM could then be constructed by adapting the approach described in Section II.3 to this reduced-order approximation (IV.13).

In the second part of this thesis, we have proposed two applications of the optimal transport problem to improve the accuracy and reliability of the ROM described in Chapter II.

In the first application, the sampling of the solution manifold has been completed with artificial snapshots generated by optimal transport. In this strategy, only snapshots that bring new information are created, enabling a fast enrichment of the snapshot database without employing the computationally expensive HDM. This improvement has been evaluated on the prediction of a shock wave in 1D. The results show that the snapshot database enrichment improves the reliability of the ROM for the prediction of new solutions.

In the second application, the Wasserstein distance has been coupled with a cluster analysis method to partition the snapshot database. The objective of this clustering is to automatically identify regions where the behaviour of the solution is similar to decompose the domain. The physical domain is then partitioned into subdomains, and different local trial subspaces are employed in each subdomain

to approximate the solution. This local approach has been evaluated on the reproduction of a shock wave in 1D. The results demonstrate that the local approach is more accurate than the global approach.

Depending on the application, several perspectives could be interesting to explore.

- *Selection of the artificial snapshots.* The enrichment of the snapshot database could be improved by automatically identifying the snapshots to create. For this purpose, the Wasserstein distance could be employed to select the snapshots that will bring new information to the snapshot database.

- *Partitioning of the temporal and parametric domains.* The snapshot database is partitioned with respect to the physical domain, but the snapshots could also be clustered with respect to time and input parameters [5]. In this way, the basis functions $\Phi_n^{f,l}(\boldsymbol{\xi})$ could be chosen at each point $\mathbf{x}$, time instance t and input parameter $\boldsymbol{\mu}$ in order to improve the approximation of the distribution functions.

- *Choice of the number of basis functions.* In the local ROM, we use the same number of basis function in all subdomains, but the size of the different local reduced bases can be different. In this way, the local approach could also improve the computational complexity of the ROM since less local basis functions are required to obtain accurate approximations.

- *Unbalanced optimal transport.* Since the distribution functions have not necessarily the same total mass, these ones are normalized before employing the optimal transport problem. To avoid this normalization step, the optimal transport problem can be replaced by the unbalanced optimal transport problem [15, 75] where the distribution functions can have different total mass. In particular, the entropic-regularization of the unbalanced optimal transport problem is derived in [39].

- *Extension to higher dimensions.* While these two applications have been evaluated here in 1D, this work could also be extended to higher dimensions. However, the entropic-regularization of the optimal transport problem may lead to unstable results for small values of γ, limiting its application to small-scale problems. Even though the accuracy of the ROM does not directly depend on the accuracy of the optimal transport solution, large values of γ may lead to poor approximations, causing difficulties to compare and interpolate the distribution functions. To address this limitation, a recent approach [99] have been developed for the entropic-regularization of the optimal transport problem. This work consider a log-domain implementation in order to obtain stable computations even for small values of γ.

The last contribution of this thesis concerns the development of a discontinuous Galerkin domain decomposition (DGDD) method for model order reduction. In this approach, the ROM approximates the solution in regions where significant dimensionality reduction can be achieved, while the HDM is employed elsewhere. Notably, the discontinuous Galerkin formulation for the ROM offers a simple way to perform the coupling between the HDM and ROMs since the global solution is recovered by linking the local solutions at the interface between subdomains though the numerical fluxes. Compared to the standard DG method, the polynomial shape functions have been replaced by POD modes constructed during the training stage in order to best approximate the solution snapshots. In addition, the ROM has been equipped with hyper-reduction techniques such as the ECSW method, which is particularly well suited to approximate the volume and surface integrals involved in the DG formulation.

ROMs based on the proposed DGDD framework have been evaluated for parametric problems governed by the quasi-1D and 2D Euler equations. We have validated the DGDD method on the reproduction of an isentropic vortex. We have then investigated the prediction of unsteady flows in a converging-diverging nozzle and over a NACA 0012 airfoil. The results demonstrate the accuracy of the method, capable of delivering less than 1% of error over a range of predictive input parameters, and the significant reduction (approximately 78%) of the required computation time for the ROM simulations versus the associated HDM.

In perspective, several approaches could be employed to further improve this method.

- *Automatic domain decomposition.* The computational complexity of the DGDD-based ROMs could be further reduced by optimally reducing the number of micro- and macro-cells. To this end, the domain can be decomposed based on an error indicator, as in [19], instead of using an *a priori* decomposition.

- *Nonlinear approximation.* Another perspective for reducing the computational cost of the DGDD method would be to replace the HDM by a ROM in the high-fidelity region. To approximate the local solution features (e.g. discontinuities and fronts), this ROM could employ a nonlinear trial subspace [64, 91, 71, 107] instead of a linear trial subspace in order to improve dimensionally reduction.

- *Extension to higher order differential equations.* The DGDD method could also be extended to higher order differential equations, such as the Navier-Stokes equations and elliptic problems, by adapting the discontinuous Galerkin method developed in [8, 13, 14, 42, 88, 108] to the ROM approach.

Appendix A

Preservation of properties of the HDM in 1D and 2D

In Section II.3.3.2, the approximate Maxwellian distribution function is computed to conserve the mass, momentum and energy of the gas and to be as close as possible to the Maxwellian distribution function M_f. In 1D and 2D, we use the same idea to satisfy equation (II.6) (resp. (II.7)) in 1D (resp. 2D):

$$\mathbf{a}^{M_f}(\mathbf{x},t;\boldsymbol{\mu}) = \boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{M}_f(\mathbf{x},t;\boldsymbol{\mu}) + \boldsymbol{\Psi}^+\left(\boldsymbol{\rho}(\mathbf{x},t;\boldsymbol{\mu}) - \boldsymbol{\Psi}\boldsymbol{\Phi}^T\boldsymbol{\Theta}\mathbf{M}_f(\mathbf{x},t;\boldsymbol{\mu})\right),$$

where $\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$, $\mathbf{a}^{M_f}(\mathbf{x},t;\boldsymbol{\mu})$, $\mathbf{M}_f(\mathbf{x},t;\boldsymbol{\mu})$ and $\boldsymbol{\rho}(\mathbf{x},t;\boldsymbol{\mu})$ are redefined as follows.

1D case. In 1D, the approximate equilibrium distribution functions are the solution of the minimization problem:

$$\begin{cases} \underset{\mathbf{a}^{M_\phi},\mathbf{a}^{M_\psi}\in\mathbb{R}^{N_{pod}}}{\text{minimize}} & \left\|\widetilde{M_{\phi_h}} - M_\phi\right\|_{\boldsymbol{\Theta}}^2 + \left\|\widetilde{M_{\psi_h}} - M_\psi\right\|_{\boldsymbol{\Theta}}^2 \\ \text{subject to} & \left\langle \widetilde{M_{\phi_h}}, \begin{pmatrix} 1 \\ \xi_u \\ \frac{\xi_u^2}{2} \end{pmatrix} \right\rangle_{\boldsymbol{\Theta}} + \left\langle \widetilde{M_{\psi_h}}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\rangle_{\boldsymbol{\Theta}} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}. \end{cases}$$

The objective function can be written using the $(2N_{\boldsymbol{\xi}}) \times (N_{pod}^{\phi} + N_{pod}^{\psi})$ system

$$\underbrace{\begin{pmatrix} \Phi_1^\phi(\xi_{u_1}) & \cdots & \Phi_{N_{pod}^\phi}^\phi(\xi_{u_1}) & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ \Phi_1^\phi(\xi_{u_{N_{\boldsymbol{\xi}}}}) & \cdots & \Phi_{N_{pod}^\phi}^\phi(\xi_{u_{N_{\boldsymbol{\xi}}}}) & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \Phi_1^\psi(\xi_{u_1}) & \cdots & \Phi_{N_{pod}^\psi}^\psi(\xi_{u_1}) \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & \Phi_1^\psi(\xi_{u_{N_{\boldsymbol{\xi}}}}) & \cdots & \Phi_{N_{pod}^\psi}^\psi(\xi_{u_{N_{\boldsymbol{\xi}}}}) \end{pmatrix}}_{\boldsymbol{\Phi}} \underbrace{\begin{pmatrix} a_1^{M_\phi}(x,t;\boldsymbol{\mu}) \\ \vdots \\ a_{N_{pod}^\phi}^{M_\phi}(x,t;\boldsymbol{\mu}) \\ a_1^{M_\psi}(x,t;\boldsymbol{\mu}) \\ \vdots \\ a_{N_{pod}^\psi}^{M_\psi}(x,t;\boldsymbol{\mu}) \end{pmatrix}}_{\mathbf{a}^{M_f}(x,t;\boldsymbol{\mu})} \approx \underbrace{\begin{pmatrix} M_\phi(x,\xi_{u_1},t;\boldsymbol{\mu}) \\ \vdots \\ M_\phi(x,\xi_{u_{N_{\boldsymbol{\xi}}}},t;\boldsymbol{\mu}) \\ M_\psi(x,\xi_{u_1},t;\boldsymbol{\mu}) \\ \vdots \\ M_\psi(x,\xi_{u_{N_{\boldsymbol{\xi}}}},t;\boldsymbol{\mu}) \end{pmatrix}}_{\mathbf{M}_f(x,t;\boldsymbol{\mu})},$$

and the equality constraints lead to the $3 \times (N_{pod}^{\phi} + N_{pod}^{\psi})$ system

$$\underbrace{\begin{pmatrix} \left\langle \Phi_1^{\phi}, 1 \right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}^{\phi}}^{\phi}, 1 \right\rangle_{\boldsymbol{\Theta}} & 0 & \cdots & 0 \\ \left\langle \Phi_1^{\phi}, \xi_u \right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}^{\phi}}^{\phi}, \xi_u \right\rangle_{\boldsymbol{\Theta}} & 0 & \cdots & 0 \\ \left\langle \Phi_1^{\phi}, \frac{\xi_u^2}{2} \right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}^{\phi}}^{\phi}, \frac{\xi_u^2}{2} \right\rangle_{\boldsymbol{\Theta}} & \left\langle \Phi_1^{\psi}, 1 \right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}^{\psi}}^{\psi}, 1 \right\rangle_{\boldsymbol{\Theta}} \end{pmatrix}}_{\boldsymbol{\Psi}} \underbrace{\begin{pmatrix} a_1^{M_\phi}(x,t;\boldsymbol{\mu}) \\ \vdots \\ a_{N_{pod}^{\phi}}^{M_\phi}(x,t;\boldsymbol{\mu}) \\ a_1^{M_\psi}(x,t;\boldsymbol{\mu}) \\ \vdots \\ a_{N_{pod}^{\psi}}^{M_\psi}(x,t;\boldsymbol{\mu}) \end{pmatrix}}_{\mathbf{a}^{M_f}(x,t;\boldsymbol{\mu})} = \underbrace{\begin{pmatrix} \rho(x,t;\boldsymbol{\mu}) \\ \rho(x,t;\boldsymbol{\mu})u(x,t;\boldsymbol{\mu}) \\ E(x,t;\boldsymbol{\mu}) \end{pmatrix}}_{\boldsymbol{\rho}(x,t;\boldsymbol{\mu})}.$$

2D case. Similarly in 2D, the approximate equilibrium distribution functions are solutions to the problem:

$$\begin{cases} \underset{\mathbf{a}^{M_\phi}, \mathbf{a}^{M_\psi} \in \mathbb{R}^{N_{pod}}}{\text{minimize}} & \left\| \widetilde{M}_{\phi_h} - M_\phi \right\|_{\boldsymbol{\Theta}}^2 + \left\| \widetilde{M}_{\psi_h} - M_\psi \right\|_{\boldsymbol{\Theta}}^2 \\ \text{subject to} & \left\langle \widetilde{M}_{\phi_h}, \begin{pmatrix} 1 \\ \xi_u \\ \xi_v \\ \frac{\|\boldsymbol{\xi}_2\|_2^2}{2} \end{pmatrix} \right\rangle_{\boldsymbol{\Theta}} + \left\langle \widetilde{M}_{\psi_h}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \right\rangle_{\boldsymbol{\Theta}} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}. \end{cases}$$

The objective function leads to the $(2N_{\boldsymbol{\xi}}) \times (N_{pod}^{\phi} + N_{pod}^{\psi})$ system

$$\underbrace{\begin{pmatrix} \Phi_1^{\phi}(\boldsymbol{\xi}_{2_1}) & \cdots & \Phi_{N_{pod}^{\phi}}^{\phi}(\boldsymbol{\xi}_{2_1}) & 0 & \cdots & 0 \\ \vdots & & \vdots & \vdots & & \vdots \\ \Phi_1^{\phi}(\boldsymbol{\xi}_{2_{N_{\boldsymbol{\xi}}}}) & \cdots & \Phi_{N_{pod}^{\phi}}^{\phi}(\boldsymbol{\xi}_{2_{N_{\boldsymbol{\xi}}}}) & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \Phi_1^{\psi}(\boldsymbol{\xi}_{2_1}) & \cdots & \Phi_{N_{pod}^{\psi}}^{\psi}(\boldsymbol{\xi}_{2_1}) \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & \Phi_1^{\psi}(\boldsymbol{\xi}_{2_{N_{\boldsymbol{\xi}}}}) & \cdots & \Phi_{N_{pod}^{\psi}}^{\psi}(\boldsymbol{\xi}_{2_{N_{\boldsymbol{\xi}}}}) \end{pmatrix}}_{\boldsymbol{\Phi}} \underbrace{\begin{pmatrix} a_1^{M_\phi}(\mathbf{x},t;\boldsymbol{\mu}) \\ \vdots \\ a_{N_{pod}^{\phi}}^{M_\phi}(\mathbf{x},t;\boldsymbol{\mu}) \\ a_1^{M_\psi}(\mathbf{x},t;\boldsymbol{\mu}) \\ \vdots \\ a_{N_{pod}^{\psi}}^{M_\psi}(\mathbf{x},t;\boldsymbol{\mu}) \end{pmatrix}}_{\mathbf{a}^{M_f}(\mathbf{x},t;\boldsymbol{\mu})} \approx \underbrace{\begin{pmatrix} M_\phi(\mathbf{x},\boldsymbol{\xi}_{2_1},t;\boldsymbol{\mu}) \\ \vdots \\ M_\phi(\mathbf{x},\boldsymbol{\xi}_{2_{N_{\boldsymbol{\xi}}}},t;\boldsymbol{\mu}) \\ M_\psi(\mathbf{x},\boldsymbol{\xi}_{2_1},t;\boldsymbol{\mu}) \\ \vdots \\ M_\psi(\mathbf{x},\boldsymbol{\xi}_{2_{N_{\boldsymbol{\xi}}}},t;\boldsymbol{\mu}) \end{pmatrix}}_{\mathbf{M}_f(\mathbf{x},t;\boldsymbol{\mu})},$$

and the equality constraints can be written using the $4 \times (N_{pod}^{\phi} + N_{pod}^{\psi})$ system

$$\underbrace{\begin{pmatrix} \left\langle \Phi_1^{\phi}, 1 \right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}^{\phi}}^{\phi}, 1 \right\rangle_{\boldsymbol{\Theta}} & 0 & \cdots & 0 \\ \left\langle \Phi_1^{\phi}, \xi_u \right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}^{\phi}}^{\phi}, \xi_u \right\rangle_{\boldsymbol{\Theta}} & 0 & \cdots & 0 \\ \left\langle \Phi_1^{\phi}, \xi_v \right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}^{\phi}}^{\phi}, \xi_v \right\rangle_{\boldsymbol{\Theta}} & 0 & \cdots & 0 \\ \left\langle \Phi_1^{\phi}, \frac{\|\boldsymbol{\xi}_2\|_2^2}{2} \right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}^{\phi}}^{\phi}, \frac{\|\boldsymbol{\xi}_2\|_2^2}{2} \right\rangle_{\boldsymbol{\Theta}} & \left\langle \Phi_1^{\psi}, 1 \right\rangle_{\boldsymbol{\Theta}} & \cdots & \left\langle \Phi_{N_{pod}^{\psi}}^{\psi}, 1 \right\rangle_{\boldsymbol{\Theta}} \end{pmatrix}}_{\boldsymbol{\Psi}} \underbrace{\begin{pmatrix} a_1^{M_\phi}(\mathbf{x},t;\boldsymbol{\mu}) \\ \vdots \\ a_{N_{pod}^{\phi}}^{M_\phi}(\mathbf{x},t;\boldsymbol{\mu}) \\ a_1^{M_\psi}(\mathbf{x},t;\boldsymbol{\mu}) \\ \vdots \\ a_{N_{pod}^{\psi}}^{M_\psi}(\mathbf{x},t;\boldsymbol{\mu}) \end{pmatrix}}_{\mathbf{a}^{M_f}(\mathbf{x},t;\boldsymbol{\mu})} = \underbrace{\begin{pmatrix} \rho(\mathbf{x},t;\boldsymbol{\mu}) \\ \rho(\mathbf{x},t;\boldsymbol{\mu})u(\mathbf{x},t;\boldsymbol{\mu}) \\ \rho(\mathbf{x},t;\boldsymbol{\mu})v(\mathbf{x},t;\boldsymbol{\mu}) \\ E(\mathbf{x},t;\boldsymbol{\mu}) \end{pmatrix}}_{\boldsymbol{\rho}(\mathbf{x},t;\boldsymbol{\mu})}.$$

Bibliography

[1] E. Abbate, A. Iollo, and G. Puppo. An all-speed relaxation scheme for gases and compressible materials. *Journal of Computational Physics*, 351:1–24, 2017.

[2] R. Abgrall, D. Amsallem, and R. Crisovan. Robust model reduction by L^1-norm minimization and approximation via dictionaries: application to nonlinear hyperbolic problems. *Advanced Modeling and Simulation in Engineering Sciences*, 3(1):1, 2016.

[3] L. Ambrosio. Lecture notes on optimal transport problems. In *Mathematical aspects of evolving interfaces*, pages 1–52. Springer, 2003.

[4] D. Amsallem and C. Farhat. Stabilization of projection-based reduced-order models. *International Journal for Numerical Methods in Engineering*, 91(4):358–377, 2012.

[5] D. Amsallem, M. J. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92(10):891–916, 2012.

[6] S. S. An, T. Kim, and D. L. James. Optimizing cubature for efficient integration of subspace deformations. *ACM transactions on graphics (TOG)*, 27(5):1–10, 2008.

[7] P. F. Antonietti, P. Pacciarini, and A. Quarteroni. A discontinuous Galerkin reduced basis element method for elliptic problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 50(2):337–360, 2016.

[8] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM journal on numerical analysis*, 19(4):742–760, 1982.

[9] D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[10] U. M. Ascher, S. J. Ruuth, and R. J. Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2-3):151–167, 1997.

[11] M. F. Barone, I. Kalashnikova, D. J. Segalman, and H. K. Thornquist. Stable Galerkin reduced order models for linearized compressible flow. *Journal of Computational Physics*, 228(6):1932–1946, 2009.

[12] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.

[13] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of computational physics*, 131(2):267–279, 1997.

[14] C. E. Baumann and J. T. Oden. A discontinuous hp finite element method for convection–diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 175(3-4):311–341, 1999.

[15] J.-D. Benamou. Numerical resolution of an ?unbalanced? mass transport problem. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 37(5):851–868, 2003.

[16] J.-D. Benamou and Y. Brenier. A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

[17] J.-D. Benamou, B. D. Froese, and A. M. Oberman. Numerical solution of the optimal transportation problem using the Monge–Ampère equation. *Journal of Computational Physics*, 260:107–126, 2014.

[18] M. Bergmann, C.-H. Bruneau, and A. Iollo. Enablers for robust POD models. *Journal of Computational Physics*, 228(2):516–538, 2009.

[19] M. Bergmann, A. Ferrero, A. Iollo, E. Lombardi, A. Scardigli, and H. Telib. A zonal Galerkin-free POD model for incompressible flows. *Journal of Computational Physics*, 352:301–325, 2018.

[20] G. Berkooz, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.

[21] F. Bernard. *Efficient Asymptotic Preserving Schemes for BGK and ES-BGK models on Cartesian grids*. PhD thesis, Bordeaux, 2015.

[22] F. Bernard, A. Iollo, and S. Riffaud. Reduced-order model for the BGK equation based on POD and optimal transport. *Journal of Computational Physics*, 373:545–570, 2018.

[23] P. L. Bhatnagar, E. P. Gross, and M. Krook. A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Physical review*, 94(3):511, 1954.

[24] G. Bird. Molecular gas dynamics and the direct simulation monte carlo of gas flows. *Clarendon, Oxford*, 508:128, 1994.

[25] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[26] L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

[27] Y. Brenier. Décomposition polaire et réarrangement monotone des champs de vecteurs. *CR Acad. Sci. Paris Sér. I Math.*, 305:805–808, 1987.

[28] Y. Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on pure and applied mathematics*, 44(4):375–417, 1991.

[29] M. Buffoni, H. Telib, and A. Iollo. Iterative methods for model reduction by domain decomposition. *Computers & Fluids*, 38(6):1160–1167, 2009.

[30] J. Burkardt, M. Gunzburger, and H.-C. Lee. POD and CVT-based reduced-order modeling of Navier–Stokes flows. *Computer methods in applied mechanics and engineering*, 196(1-3):337–355, 2006.

[31] H. Cabannes, R. Gatignol, and L. S. Luo. The discrete Boltzmann equation. *Lecture Notes at University of California, Berkley*, pages 1–65, 1980.

[32] K. Carlberg, M. Barone, and H. Antil. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. *Journal of Computational Physics*, 330:693–734, 2017.

[33] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.

[34] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242:623–647, 2013.

[35] C. Cercignani. The boltzmann equation. In *The Boltzmann equation and its applications*, pages 40–103. Springer, 1988.

[36] S. Chapman, T. G. Cowling, and D. Burnett. *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*. Cambridge university press, 1990.

[37] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[38] F. Chinesta, A. Leygue, F. Bordeu, J. V. Aguado, E. Cueto, D. González, I. Alfaro, A. Ammar, and A. Huerta. PGD-based computational vademecum for efficient design, optimization and control. *Archives of Computational Methods in Engineering*, 20(1):31–59, 2013.

[39] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard. Scaling algorithms for unbalanced optimal transport problems. *Mathematics of Computation*, 87(314):2563–2609, 2018.

[40] C. Chu. Kinetic-theoretic description of the formation of a shock wave. *The Physics of Fluids*, 8(1):12–22, 1965.

[41] B. Cockburn. Discontinuous galerkin methods. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics*, 83(11):731–754, 2003.

[42] B. Cockburn and C.-W. Shu. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998.

[43] B. Cockburn and C.-W. Shu. The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. *Journal of Computational Physics*, 141(2):199–224, 1998.

[44] M. Couplet, C. Basdevant, and P. Sagaut. Calibrated reduced-order POD-Galerkin system for fluid flow modelling. *Journal of Computational Physics*, 207(1):192–220, 2005.

[45] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.

[46] M. Cuturi and A. Doucet. Fast computation of Wasserstein barycenters. In *International Conference on Machine Learning*, pages 685–693, 2014.

[47] M. Dubiner. Spectral methods on triangles and other domains. *Journal of Scientific Computing*, 6(4):345–390, 1991.

[48] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[49] R. Everson and L. Sirovich. Karhunen–Loeve procedure for gappy data. *JOSA A*, 12(8):1657–1664, 1995.

[50] C. Farhat, P. Avery, T. Chapman, and J. Cortial. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *International Journal for Numerical Methods in Engineering*, 98(9):625–662, 2014.

[51] C. Farhat, T. Chapman, and P. Avery. Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models. *International Journal for Numerical Methods in Engineering*, 102(5):1077–1110, 2015.

[52] G. Forsythe, M. Malcolm, and C. Moler. Computer Methods for Mathematical Computation Prentice-Hall. *Englewood Cliffs, New Jersey*, 1977.

[53] W. Gangbo and R. J. McCann. The geometry of optimal transportation. Technical report, SCAN-9604031, 1996.

[54] S. Gérald. *Méthode de Galerkin Discontinue et intégrations explicites-implicites en temps basées sur un découplage des degrés de liberté. Applications au système des équations de Navier-Stokes.* PhD thesis, Université Pierre et Marie Curie - Paris VI, 2013.

[55] S. Grimberg, C. Farhat, R. Tezaur, and C. Bou-Mosleh. Mesh sampling and weighting for the hyperreduction of nonlinear Petrov-Galerkin reduced-order models with local reduced-order bases, 2020.

[56] S. Grimberg, C. Farhat, and N. Youkilis. On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows. *Journal of Computational Physics*, 419:109681, 2020.

[57] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 42(2):277–302, 2008.

[58] A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of computational physics*, 135(2):260–278, 1997.

[59] A. Harten, P. D. Lax, and B. v. Leer. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM review*, 25(1):35–61, 1983.

[60] J. A. Hernandez, M. A. Caicedo, and A. Ferrer. Dimensional hyper-reduction of nonlinear finite element models via empirical cubature. *Computer methods in applied mechanics and engineering*, 313:687–722, 2017.

[61] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.

[62] A. Iollo, A. Dervieux, J.-A. Désidéri, and S. Lanteri. Two stable POD-based approximations to the Navier–Stokes equations. *Computing and visualization in science*, 3(1-2):61–66, 2000.

[63] A. Iollo, S. Lanteri, and J.-A. Désidéri. Stability properties of POD–Galerkin approximations for the compressible Navier–Stokes equations. *Theoretical and Computational Fluid Dynamics*, 13(6):377–396, 2000.

[64] A. Iollo and D. Lombardi. Advection modes by optimal mass transfer. *Physical Review E*, 89(2):022923, 2014.

[65] E. Kaiser, B. R. Noack, L. Cordier, A. Spohn, M. Segond, M. Abel, G. Daviller, J. Östh, S. Krajnović, and R. K. Niven. Cluster-based reduced-order modelling of a mixing layer. *Journal of Fluid Mechanics*, 754:365–414, 2014.

[66] L. V. Kantorovich. On a problem of Monge. In *CR (Doklady) Acad. Sci. URSS (NS)*, volume 3, pages 225–226, 1948.

[67] C. A. Kennedy and M. H. Carpenter. Additive Runge–Kutta schemes for convection–diffusion–reaction equations. *Applied numerical mathematics*, 44(1-2):139–181, 2003.

[68] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[69] D. Kuzmin. A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods. *Journal of computational and applied mathematics*, 233(12):3077–3085, 2010.

[70] C. L. Lawson and R. J. Hanson. *Solving least squares problems*, volume 15. Siam, 1995.

[71] K. Lee and K. T. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020.

[72] P. LeGresley and J. Alonso. Dynamic domain decomposition and error correction for reduced order models. In *41st Aerospace Sciences Meeting and Exhibit*, page 250, 2003.

[73] P. A. LeGresley. *Application of proper orthogonal decomposition (POD) to design decomposition methods*. Stanford University, 2006.

[74] S. Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[75] D. Lombardi and E. Maitre. Eulerian models and algorithms for unbalanced optimal transport. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1717–1744, 2015.

[76] D. Lucia, P. King, M. Oxley, and P. Beran. Reduced order modeling for a one-dimensional nozzle flow with moving shocks. In *15th AIAA computational fluid dynamics conference*, page 2602, 2001.

[77] A. Mendible, S. L. Brunton, A. Y. Aravkin, W. Lowrie, and J. N. Kutz. Dimensionality Reduction and Reduced Order Modeling for Traveling Wave Physics. *arXiv preprint arXiv:1911.00565*, 2019.

[78] L. Mieussens. Discrete-velocity models and numerical schemes for the Boltzmann-BGK equation in plane and axisymmetric geometries. *Journal of Computational Physics*, 162(2):429–466, 2000.

[79] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *The quarterly journal of mathematics*, 11(1):50–59, 1960.

[80] G. Monge. Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

[81] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE transactions on automatic control*, 26(1):17–32, 1981.

[82] N. J. Nair and M. Balajewicz. Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter dependent shocks. *arXiv preprint arXiv:1712.09144*, 2017.

[83] M. Nonino, F. Ballarin, G. Rozza, and Y. Maday. Overcoming slowly decaying Kolmogorov n-width by transport maps: application to model order reduction of fluid dynamics and fluid–structure interaction problems. *arXiv preprint arXiv:1911.06598*, 2019.

[84] A. M. Oberman. Wide stencil finite difference schemes for the elliptic Monge-Ampère equation and functions of the eigenvalues of the Hessian. *Discrete Contin. Dyn. Syst. Ser. B*, 10(1):221–238, 2008.

[85] N. Papadakis, G. Peyré, and E. Oudet. Optimal transport with proximal splitting. *SIAM Journal on Imaging Sciences*, 7(1):212–238, 2014.

[86] L. Pareschi and G. Russo. Implicit–explicit Runge–Kutta schemes and applications to hyperbolic systems with relaxation. *Journal of Scientific computing*, 25(1):129–155, 2005.

[87] K. Pearson. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[88] J. Peraire and P.-O. Persson. The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM Journal on Scientific Computing*, 30(4):1806–1824, 2008.

[89] M. Rathinam and L. R. Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893–1925, 2003.

[90] W. H. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Lab., N. Mex.(USA), 1973.

[91] J. Reiss. Optimization-based modal decomposition for systems with multiple transports. *arXiv preprint arXiv:2002.11789*, 2020.

[92] S. Riffaud, M. Bergmann, C. Farhat, S. Grimberg, and A. Iollo. The DGDD method for reduced-order modeling of conservation laws. Manuscript submitted for publication, 2020.

[93] P. L. Roe. Characteristic-based schemes for the Euler equations. *Annual review of fluid mechanics*, 18(1):337–365, 1986.

[94] C. W. Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. In *Modeling And Computations In Dynamical Systems: In Commemoration of the 100th Anniversary of the Birth of John von Neumann*, pages 301–317. World Scientific, 2006.

[95] C. W. Rowley, T. Colonius, and R. M. Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D: Nonlinear Phenomena*, 189(1-2):115–129, 2004.

[96] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):1, 2008.

[97] F. Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55(58-63):94, 2015.

[98] E. Schmidt. Zur Theorie der linearen und nichtlinearen Integralgleichungen. In *Integralgleichungen und Gleichungen mit unendlich vielen Unbekannten*, pages 190–233. Springer, 1989.

[99] B. Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481, 2019.

[100] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of computational physics*, 77(2):439–471, 1988.

[101] R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, 35(2):876–879, 1964.

[102] R. Sinkhorn and P. Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[103] L. Sirovich. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quarterly of applied mathematics*, 45(3):561–571, 1987.

[104] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of computational physics*, 27(1):1–31, 1978.

[105] J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG)*, 34(4):66, 2015.

[106] H. Steinhaus. Sur la division des corp materiels en parties. *Bull. Acad. Polon. Sci*, 1(804):801, 1956.

[107] T. Taddei and L. Zhang. Space-time registration-based model reduction of parameterized one-dimensional hyperbolic PDEs. *arXiv preprint arXiv:2004.06693*, 2020.

[108] B. Van Leer and S. Nomura. Discontinuous Galerkin for diffusion. In *17th AIAA Computational Fluid Dynamics Conference*, page 5108, 2005.

[109] K. Veroy and A. Patera. Certified real-time solution of the parametrized steady incompressible Navier–Stokes equations: rigorous reduced-basis a posteriori error bounds. *International Journal for Numerical Methods in Fluids*, 47(8-9):773–788, 2005.

[110] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, 2007.

[111] C. Villani. *Topics in Optimal Transportation*. Graduate studies in mathematics. American Mathematical Society, 2003.

[112] C. Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[113] G. Welper. Interpolation of functions with parameter dependent jumps by transformed snapshots. *SIAM Journal on Scientific Computing*, 39(4):A1225–A1250, 2017.

[114] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA journal*, 40(11):2323–2330, 2002.

[115] H. Xiao and Z. Gimbutas. A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Computers & mathematics with applications*, 59(2):663–676, 2010.

[116] M. Yano. Discontinuous Galerkin reduced basis empirical quadrature procedure for model reduction of parametrized nonlinear conservation laws. *Advances in Computational Mathematics*, 45(5-6):2287–2320, 2019.

[117] M. Yano and A. T. Patera. An LP empirical quadrature procedure for reduced basis treatment of parametrized nonlinear PDEs. *Computer Methods in Applied Mechanics and Engineering*, 344:1104–1123, 2019.

Modèles réduits : convergence entre calcul et données pour la mécanique des fluides

Introduction

En simulation numérique, la dynamique d'un fluide est gouvernée par un modèle mathématique impliquant la résolution d'équations aux dérivées partielles (EDP). Ces équations n'admettant généralement pas de solution analytique, le problème continu est discrétisé par des méthodes numériques, conduisant à chaque pas de temps à la résolution d'un système $N \times N$ de grande dimension

$$r_h[u_h](\mathbf{x}, t; \boldsymbol{\mu}) = 0,$$

où $\mathbf{x} \in \Omega$ désigne la variable spatiale, $t \in \mathbb{R}_+^*$ désigne le temps, $\boldsymbol{\mu} \in \mathcal{D}$ représente les paramètres d'entrée, $u_h \in \mathcal{V}_h(\Omega)$ désigne la solution discrète et r_h représente le résidu discret. La complexité de ce système peut poser problème à cause du nombre important de degrés de liberté $N \approx O(10^6, \ldots, 10^9)$ à déterminer. Dans de nombreuses applications industrielles, il est nécessaire de résoudre efficacement ces systèmes, soit en raison de contraintes portant sur le temps d'exécution dans le cas de modèles de très grande dimension, soit en raison du nombre important de simulations à effectuer pour différents paramètres d'entrée $\boldsymbol{\mu}$.

Les modèles réduits ont été développés dans le but de diminuer drastiquement la complexité des simulations. Plutôt que de discrétiser la solution sans aucune connaissance du système dynamique à résoudre, les modèles réduits utilisent de l'information à posteriori afin de réduire significativement le nombre d'inconnues $M \approx O(10^1)$ à déterminer :

$$\widetilde{u}_h(\mathbf{x}, t; \boldsymbol{\mu}) = u_o(\mathbf{x}) + \sum_{n=1}^{M} a_n(t; \boldsymbol{\mu}) \Phi_n(\mathbf{x}),$$

où l'offset u_o et les modes propres de la base réduite Φ_n définissent le sous-espace affine d'approximation $\mathcal{S}_h(\Omega)$, et a_n désignent les coordonnées de la solution approchée $\widetilde{u}_h \in \mathcal{S}_h(\Omega)$ dans cet espace. La construction des modèles réduits est

ensuite similaire à l'approche utilisée en apprentissage automatique pour obtenir la réduction de la dimensionnalité ($M \ll N$). Elle consiste d'abord en une phase d'apprentissage au cours de laquelle des solutions haute-fidélité sont acquises pour différents paramètres d'entrainement $\boldsymbol{\mu}$ afin d'identifier l'espace fonctionnel $\mathcal{V}_h(\Omega)$, où évolue la solution haute-fidélité u_h, et d'en extraire le sous-espace d'approximation $\mathcal{S}_h(\Omega) \subset \mathcal{V}_h(\Omega)$ de faible dimension M représentant de manière optimale $\mathcal{V}_h(\Omega)$. Ensuite, au cours de l'étape de prédiction, la solution approchée est introduite dans le système de grande dimension, qui est lui-même projeté sur le sous-espace test, conduisant à la résolution d'un système $M \times M$ de faible dimension

$$\forall n \in \{1, \dots, M\} \ : \ \int_\Omega r_h[\widetilde{u}_h](\mathbf{x}, t; \boldsymbol{\mu}) \Psi_n(\mathbf{x}) \, \mathrm{d}\mathbf{x} = 0,$$

où les fonctions Ψ_n engendrent le sous-espace test. Le sous-espace d'approximation $\mathcal{S}_h(\Omega)$ (c.-à-d. u_o et Φ_n) étant construit au cours de la phase d'apprentissage, il ne reste plus qu'à déterminer les M coordonnées réduites a_n pendant l'étape de prédiction, permettant ainsi d'obtenir des simulations rapides voire en temps réel pour de nouveaux paramètres d'entrée $\boldsymbol{\mu}$.

Modèles réduits pour les gaz raréfiés

La première contribution de cette thèse concerne la modélisation d'écoulements gazeux dans les régimes hydrodynamique et raréfié. L'objectif est de développer un nouveau modèle réduit [22] pour l'équation de Bathnagar-Gross-Krook (BGK) [23] afin de réduire significativement le temps de calcul associé à la simulation de ces écoulements.

Modèle BGK

Dans ce travail, la dynamique de l'écoulement est gouvernée par l'équation BGK pour $\mathbf{x} \in \Omega_\mathbf{x}$, $\boldsymbol{\xi} = (\xi_u, \xi_v, \xi_w)^T \in \mathbb{R}^3$, $t \in \mathbb{R}_+^*$ et $\boldsymbol{\mu} \in \mathcal{D}$:

$$\frac{\partial f}{\partial t}(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) + \boldsymbol{\xi} \cdot \nabla_\mathbf{x} f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \frac{M_f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) - f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu})}{\tau(\mathbf{x}, t; \boldsymbol{\mu})}. \tag{1}$$

Pour chaque paramètre d'entrée $\boldsymbol{\mu}$, la fonction de distribution $f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu})$ représente l'évolution temporelle de la distribution des particules du gaz au point $\mathbf{x}$ et se déplaçant à la vitesse microscopique $\boldsymbol{\xi}$. De plus, la fonction d'équilibre maxwellienne est définie par

$$M_f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \frac{\rho(\mathbf{x}, t; \boldsymbol{\mu})}{(2\pi T(\mathbf{x}, t; \boldsymbol{\mu}))^{\frac{3}{2}}} \exp\left(-\frac{\|\boldsymbol{\xi} - \mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu})\|_2^2}{2T(\mathbf{x}, t; \boldsymbol{\mu})} \right),$$

où ρ est la densité, $\mathbf{u}$ est la vitesse macroscopique and T est la température du gaz. Ces quantités macroscopiques sont calculées à partir de la fonction de distribution

$f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu})$:

$$\int_{\mathbb{R}^3} f(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) \begin{pmatrix} 1 \\ \boldsymbol{\xi} \\ \frac{\|\boldsymbol{\xi}\|_2^2}{2} \end{pmatrix} \mathrm{d}\boldsymbol{\xi} = \begin{pmatrix} \rho(\mathbf{x}, t; \boldsymbol{\mu}) \\ \rho(\mathbf{x}, t; \boldsymbol{\mu}) \mathbf{u}(\mathbf{x}, t; \boldsymbol{\mu}) \\ E(\mathbf{x}, t; \boldsymbol{\mu}) \end{pmatrix}. \tag{2}$$

Notamment, cette équation connecte le comportement microscopique des particules avec l'état macroscopique du gaz. Elle est vérifiée par toutes les fonctions de distribution (f et M_f) et assure la conservation de la masse, de la quantité de mouvement et de l'énergie du gaz.

Modèle d'ordre réduit

Afin de réduire le nombre de degrés de liberté, les fonctions de distribution sont approximées par une combinaison linéaire de fonctions à variables séparées :

$$\widetilde{f_h}(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \sum_{n=1}^{N_{pod}} a_n^f(\mathbf{x}, t; \boldsymbol{\mu}) \Phi_n(\boldsymbol{\xi}) \quad \text{et} \quad \widetilde{M_{f_h}}(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \sum_{n=1}^{N_{pod}} a_n^{M_f}(\mathbf{x}, t; \boldsymbol{\mu}) \Phi_n(\boldsymbol{\xi}).$$

Les modes propres Φ_n sont construits pendant la phase d'apprentissage par décomposition orthogonale aux valeurs propres (POD), et les coordonnées réduites a_n^f et $a_n^{M_f}$ sont déterminées au cours de l'étape de prédiction par la méthode de Galerkin.

Phase d'apprentissage

Lors de la phase d'apprentissage, des instantanés (snapshots en anglais) des fonctions de distribution sont collectées afin d'identifier le sous-espace d'approximation. Soit $s_l^f(\boldsymbol{\xi}) = f_h(\mathbf{x}_{i(l)}, \boldsymbol{\xi}, t_{k(l)}; \boldsymbol{\mu}_{j(l)})$ (resp. $s_l^M(\boldsymbol{\xi}) = M_{f_h}(\mathbf{x}_{i(l)}, \boldsymbol{\xi}, t_{k(l)}; \boldsymbol{\mu}_{j(l)})$) un instantané de la fonction de distribution haute-fidélité f_h (resp. M_{f_h}) pris au point $\mathbf{x}_{i(l)}$, au temps $t_{k(l)}$ et pour le paramètre d'entrée $\boldsymbol{\mu}_{j(l)}$, l'échantillonnage de la solution haute-fidélité conduit à la création de la base de données

$$S = \left\{ s_l^f(\boldsymbol{\xi}) \right\}_{l=1}^K \bigcup \left\{ s_l^M(\boldsymbol{\xi}) \right\}_{l=1}^K.$$

Les modes propres sont ensuite construits par POD [103] afin d'extraire le sous-espace d'approximation de faible dimension qui est optimal au sens des moindres carrés pour représenter les fonctions de distribution contenues dans S :

$$\begin{cases} \underset{\Phi_1(\boldsymbol{\xi}), \ldots, \Phi_{N_{pod}}(\boldsymbol{\xi})}{\text{minimiser}} & \displaystyle\sum_{l=1}^{2K} \int_{\mathbb{R}^3} \left(s_l(\boldsymbol{\xi}) - \widehat{s}_l(\boldsymbol{\xi}) \right)^2 \mathrm{d}\boldsymbol{\xi} \\ \text{tel que} & \displaystyle\int_{\mathbb{R}^3} \Phi_n(\boldsymbol{\xi}) \Phi_m(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi} = \delta_{n,m}, \end{cases}$$

où s_l désigne un instantané (s_l^f ou s_l^M) et $\widehat{s}_l$ représente la projection orthogonale de s_l sur l'espace d'approximation, c.-à-d. $\widehat{s}_l(\boldsymbol{\xi}) = \sum_{n=1}^{N_{pod}} \left(\int_{\mathbb{R}^3} s_l(\boldsymbol{\xi}') \Phi_n(\boldsymbol{\xi}') \, \mathrm{d}\boldsymbol{\xi}' \right) \Phi_n(\boldsymbol{\xi})$.

Étape de prédiction

Une fois les modes propres construits, les fonctions de distribution approchées ne dépendent plus que des coordonnées réduites. Celles-ci sont déterminées à faible coût par la méthode de Galerkin au cours de l'étape de prédiction. Dans cette approche, la solution approchée est introduite dans l'équation BGK (1), qui est ensuite projetée sur les modes propres, conduisant à la résolution du système d'EDP pour $n \in \{1, \dots, N_{pod}\}$:

$$\frac{\partial a_n^f}{\partial t} + \sum_{m=1}^{N_{pod}} \left(A_{n,m} \frac{\partial a_m^f}{\partial x} + B_{n,m} \frac{\partial a_m^f}{\partial y} + C_{n,m} \frac{\partial a_m^f}{\partial z} \right) = \frac{a_n^{M_f} - a_n^f}{\tau}, \tag{3}$$

où $A_{n,m} = \int_{\mathbb{R}^3} \boldsymbol{\xi}_u \Phi_n \Phi_m \, \mathrm{d}\boldsymbol{\xi}$, $B_{n,m} = \int_{\mathbb{R}^3} \boldsymbol{\xi}_v \Phi_n \Phi_m \, \mathrm{d}\boldsymbol{\xi}$ et $C_{n,m} = \int_{\mathbb{R}^3} \boldsymbol{\xi}_w \Phi_n \Phi_m \, \mathrm{d}\boldsymbol{\xi}$. Ce système est hyperbolique par construction et est résolu par la méthode des volumes finis en espace et un schéma de Runge-Kutta implicite-explicite en temps. De plus, le système (3) est modifié afin de conserver la masse, la quantité de mouvement et l'énergie du gaz. Pour cela, la fonction maxwellienne approchée est déterminée de manière à respecter l'équation (2) tout en étant la plus proche possible au sens des moindres carrés de la fonction d'équilibre maxwellienne :

$$\begin{cases} \underset{a_1^{M_f}(\mathbf{x},t;\boldsymbol{\mu}),\dots,a_{N_{pod}}^{M_f}(\mathbf{x},t;\boldsymbol{\mu})}{\text{minimiser}} & \displaystyle\int_{\mathbb{R}^3} \left(\widetilde{M}_{f_h}(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}) - M_f(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}) \right)^2 \mathrm{d}\boldsymbol{\xi} \\ \text{tel que} & \displaystyle\int_{\mathbb{R}^3} \widetilde{M}_{f_h}(\mathbf{x},\boldsymbol{\xi},t;\boldsymbol{\mu}) \begin{pmatrix} 1 \\ \boldsymbol{\xi} \\ \frac{\|\boldsymbol{\xi}\|_2^2}{2} \end{pmatrix} \mathrm{d}\boldsymbol{\xi} = \begin{pmatrix} \rho(\mathbf{x},t;\boldsymbol{\mu}) \\ \rho(\mathbf{x},t;\boldsymbol{\mu})\mathbf{u}(\mathbf{x},t;\boldsymbol{\mu}) \\ E(\mathbf{x},t;\boldsymbol{\mu}) \end{pmatrix}. \end{cases}$$

Prédiction d'un vortex

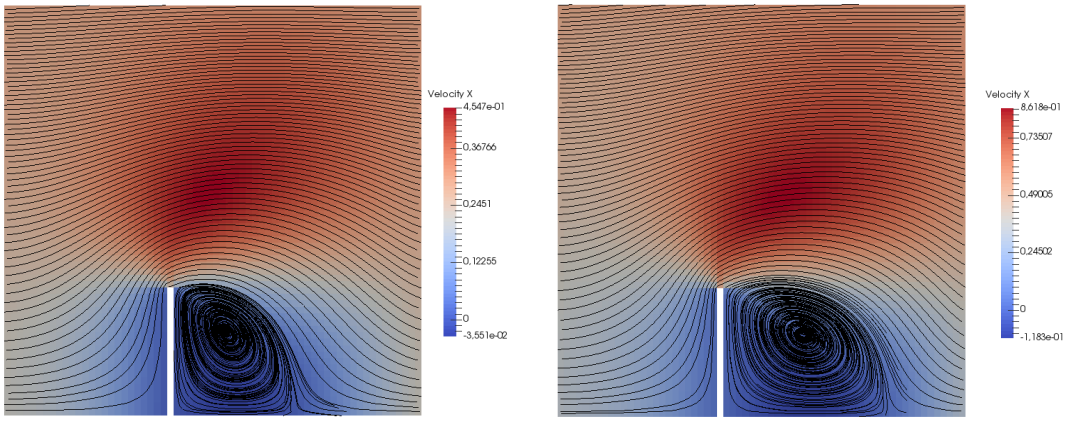
Le modèle réduit est évalué sur sa capacité à prédire un écoulement à $Kn = 0.0345$ pour différents nombres de Mach en entrée $\mu \in [0.23, 0.63]$. La condition initiale est un écoulement uniforme à Mach μ

$$\forall \mathbf{x} \in \Omega_{\mathbf{x}} \ : \ \rho_0(\mathbf{x};\mu) = 1, \ u_0(\mathbf{x};\mu) = \mu, \ v_0(\mathbf{x};\mu) = 0, \ T_0(\mathbf{x};\mu) = 1,$$

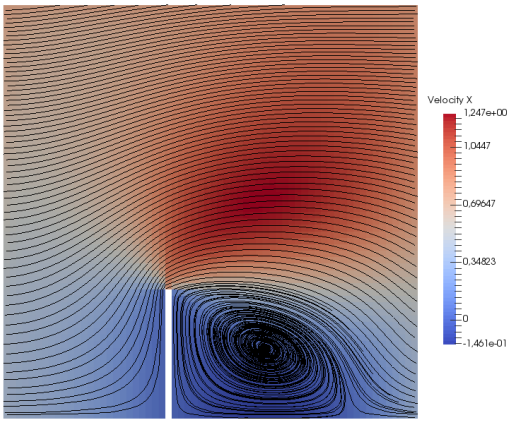
et le temps final est $t_{max} = 5.3332$. De plus, un écoulement uniforme à Mach μ est imposé au bord du domaine ($x = -1.33$, $x = 2$ et $y = 3.33$), et une réflection spéculaire est appliquée sur le mur ($\mathbf{x} = \{0\} \times]0, 1[$) et au bord ($y = 0$).

Le modèle réduit est entrainé à partir de la base de données S constituée d'instantanés collectés au cours de la simulation haute-fidélité correspondante au paramètre d'entrée $\mu = 0.63$.

Les lignes de courant de la vitesse macroscopique du gaz prédites par le modèle réduit sont affichées sur la Figure 1. Sur la Figure 2, les performances du modèle réduit sont évaluées pour différentes prédictions correspondantes aux paramètres d'entrée $\mu \in \{0.23, 0.33, 0.43, 0.53, 0.63\}$.



(a) $\mu = 0.23$ (b) $\mu = 0.43$



(c) $\mu = 0.63$

Figure 1: Lignes de courant $\mathbf{u}$ pour la prédiction de vortex avec $N_{pod} = 20$.

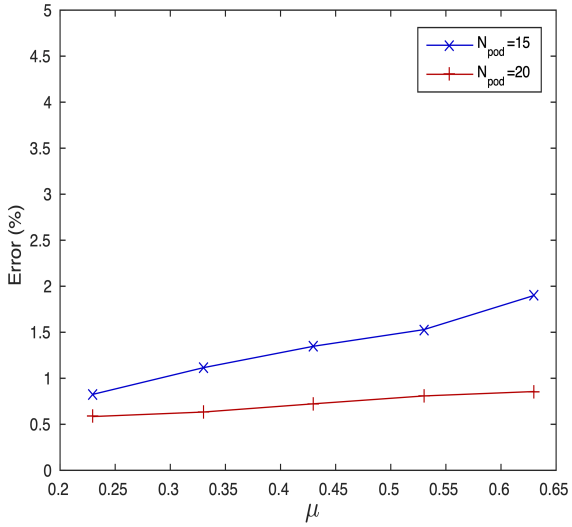


Figure 2: Précision du modèle réduit pour la prédiction de vortex.

Pour $\mu \in [0.23, 0.63]$, le modèle réduit est capable de prédire précisément les nouvelles fonctions de distribution, bien que celles-ci ne soient pas présentes dans la base de données S. En particulier, avec $N_{pod} = 20$ modes propres, l'erreur est inférieure à 1% pour tous les tests de prédiction, et le temps de calcul est divisé par environ 45 par rapport aux simulations haute-fidélité.

Transport optimal pour la réduction de modèle

Après avoir développé un modèle réduit pour l'équation BGK, deux améliorations pour ce modèle sont proposées. Celles-ci sont basées sur le problème de transport optimal, qui permet d'analyser de manière pertinente les fonctions de distribution.

Problème de transport optimal

Soient deux fonctions de distribution $f_1, f_2 : \mathbb{R}^d \to \mathbb{R}_+$ ayant la même masse totale (c.-à-d. $\int_{\mathbb{R}^d} f_1 \, \mathrm{d}\mathbf{x} = \int_{\mathbb{R}^d} f_2 \, \mathrm{d}\mathbf{x}$) et un coût de transport $c(\mathbf{x}, \mathbf{y})$ associé au déplacement d'une unité de masse de $\mathbf{x}$ vers $\mathbf{y}$. Le problème de transport optimal [80, 66] consiste à trouver le plan de transport $\pi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ minimisant

$$\min_{\pi \in \Pi(f_1, f_2)} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} c(\mathbf{x}, \mathbf{y}) \pi(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{y}, \tag{4}$$

où $\pi(\mathbf{x}, \mathbf{y})$ représente la quantité de masse déplacée de $\mathbf{x}$ vers $\mathbf{y}$ et $\Pi(f_1, f_2)$ désigne l'ensemble des plans de transport vérifiant $f_1(\mathbf{x}) = \int_{\mathbb{R}^d} \pi(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{y}$ et $f_2(\mathbf{y}) = \int_{\mathbb{R}^d} \pi(\mathbf{x}, \mathbf{y}) \, \mathrm{d}\mathbf{x}$. En particulier, lorsque le coût de transport est associé à la norme L^2 (c.-à-d. $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2^2$), le coût de transport total (4) correspond au carré de la distance L^2 de Wasserstein $\mathcal{W}_2(f_1, f_2)$ entre les functions f_1 et f_2. Cette distance offre notamment une manière naturelle de comparer et manipuler les fonctions de distribution, comme illustré à la Figure 3.

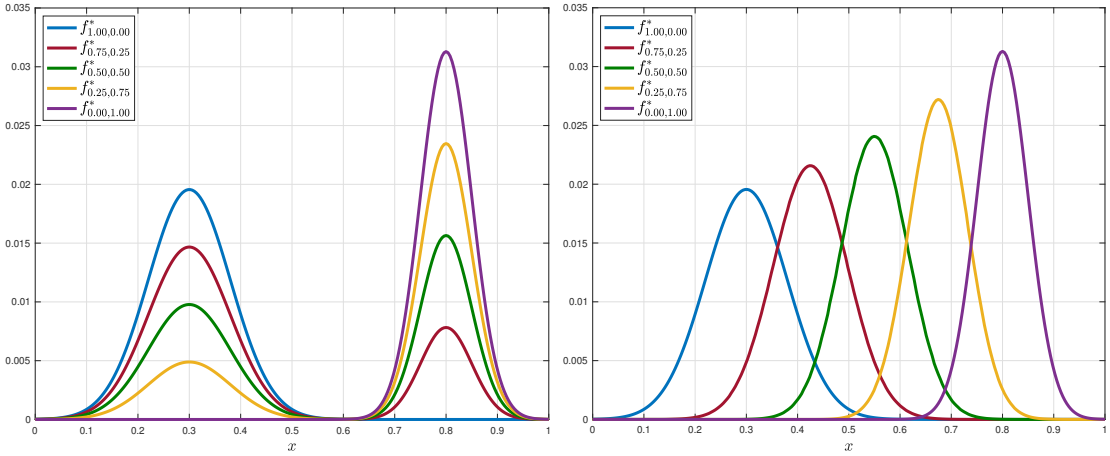


Figure 3: Comparaison de 5 interpolations barycentriques définies à partir de la norme L^2 (gauche) et de la distance L^2 de Wasserstein (droite).

Application à l'enrichissement de données

La première application du transport optimal concerne l'interpolation des fonctions de distribution. Le sous-espace d'approximation étant construit de manière à approcher les instantanés de la solution, la précision et la fiabilité du modèle réduit dépendent de la base de données S. Cependant, le nombre de simulations haute-fidélité disponibles pour la construction de la base de données est limité à cause du coût de calcul élevé associé à ces simulations. Pour cette raison, nous proposons d'enrichir la base de données S avec des nouveaux instantanés générés par transport optimal [82, 113, 22]. Ces instantanés artificiels s^* sont définis comme les barycentres de Wasserstein des instantanés haute-fidélité $(s_1, \ldots, s_K)$ aux coordonnées barycentriques $(\lambda_1, \ldots, \lambda_K)$:

$$s^* = \arg\min_s \sum_{l=1}^{K} \lambda_l \mathcal{W}_2(s_l, s)^2,$$

où $\sum_{l=1}^{K} \lambda_l = 1$ et $\lambda_l \geqslant 0$ pour $l \in \{1, \ldots, K\}$. Le modèle réduit est ensuite le même que celui présenté dans la section précédente, à l'exception de la base de données qui contient aussi les instantanés artificiels.

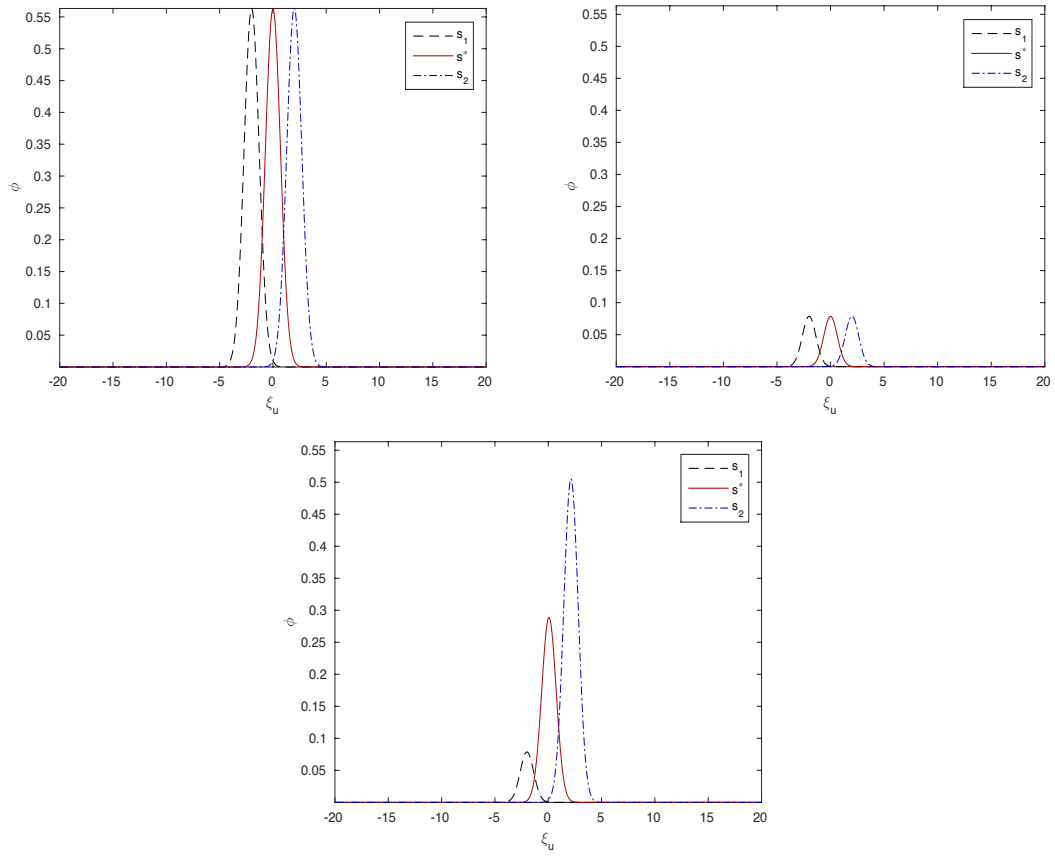


Figure 4: Examples d'instantanés artificiels (rouge) créés à partir des instantanés haute-fidélité correspondant aux simulations $\mu = -2$ (noir) et $\mu = 2$ (bleu).

L'enrichissement de la base de données est évalué pour prédire une solution qui est très différente des instantanés utilisés pour entrainer le modèle réduit. Comme illustré sur la Figure 4, le transport optimal est employé pour interpoler les instantanés haute-fidélité et enrichir la base de données. Nous comparons ensuite deux modèles réduits : le premier modèle est construit à partir des instantanés haute-fidélité, tandis que le second modèle est construit à partir des instantanés haute-fidélité et des instantanés artificiels. D'après la Figure 5, les instantanés artificiels permettent d'améliorer la fiabilité du modèle réduit pour $\mu \in [-1.5, 1.5]$. Pour $\mu \in \{-2, 2\}$, les prédictions sont légèrement moins précises car les instantanés artificiels n'apportent pas de nouvelle information utile pour approcher la solution.

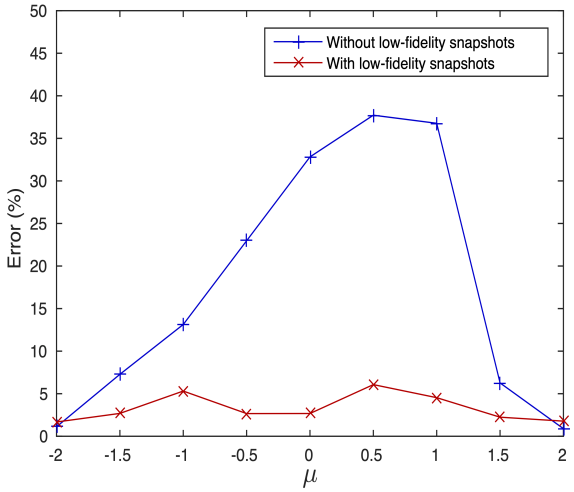


Figure 5: Précision des modèles réduits pour la prédiction d'une onde de choc avec $N_{pod} = 9$ modes propres.

Application au partitionnement de données

Le transport optimal est utilisé dans une seconde application pour comparer les fonctions de distribution. Dans le modèle réduit développé dans la section précédente, les fonctions de base sont les mêmes dans tout le domaine $\Omega_{\mathbf{x}}$, mais différentes bases réduites peuvent aussi être employées en chaque point $\mathbf{x}$ afin d'améliorer la précision de l'approximation des fonctions de distribution [5]. Cependant, la mémoire requise pour stocker $N_{\mathbf{x}}$ bases réduites peut être prohibitive à cause du grand nombre de points. Pour cette raison, nous proposons d'employer $N_c \in \{1, \ldots, N_{\mathbf{x}}\}$ bases réduites locales en fonction de la quantité de mémoire disponible. Le domaine $\Omega_{\mathbf{x}}$ est ensuite décomposé en N_c sous-domaines $\Omega_l \subseteq \Omega_{\mathbf{x}}$, et la solution est approchées dans chaque sous-domaine par la base réduite correspondante, c.-à-d. $\forall \mathbf{x} \in \Omega_l$:

$$\widetilde{f}(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \sum_{n=1}^{N_{pod}^l} a_n^f(\mathbf{x}, t; \boldsymbol{\mu}) \Phi_n^l(\boldsymbol{\xi}) \quad \text{et} \quad \widetilde{M_f}(\mathbf{x}, \boldsymbol{\xi}, t; \boldsymbol{\mu}) = \sum_{n=1}^{N_{pod}^l} a_n^M(\mathbf{x}, t; \boldsymbol{\mu}) \Phi_n^l(\boldsymbol{\xi}).$$

Cette partition du domaine est déterminée par une méthode de classification non-supervisée à partir de la base de données S. L'objectif est d'identifier les régions où le comportement de la solution est similaire pour décomposer le domaine. De plus, la mesure de similarité entre les fonctions de distribution est basée sur la distance L^2 de Wasserstein plutôt que sur la norme usuelle L^2. Le problème de classification qui en résulte est résolu par l'algorithme des k-moyennes.

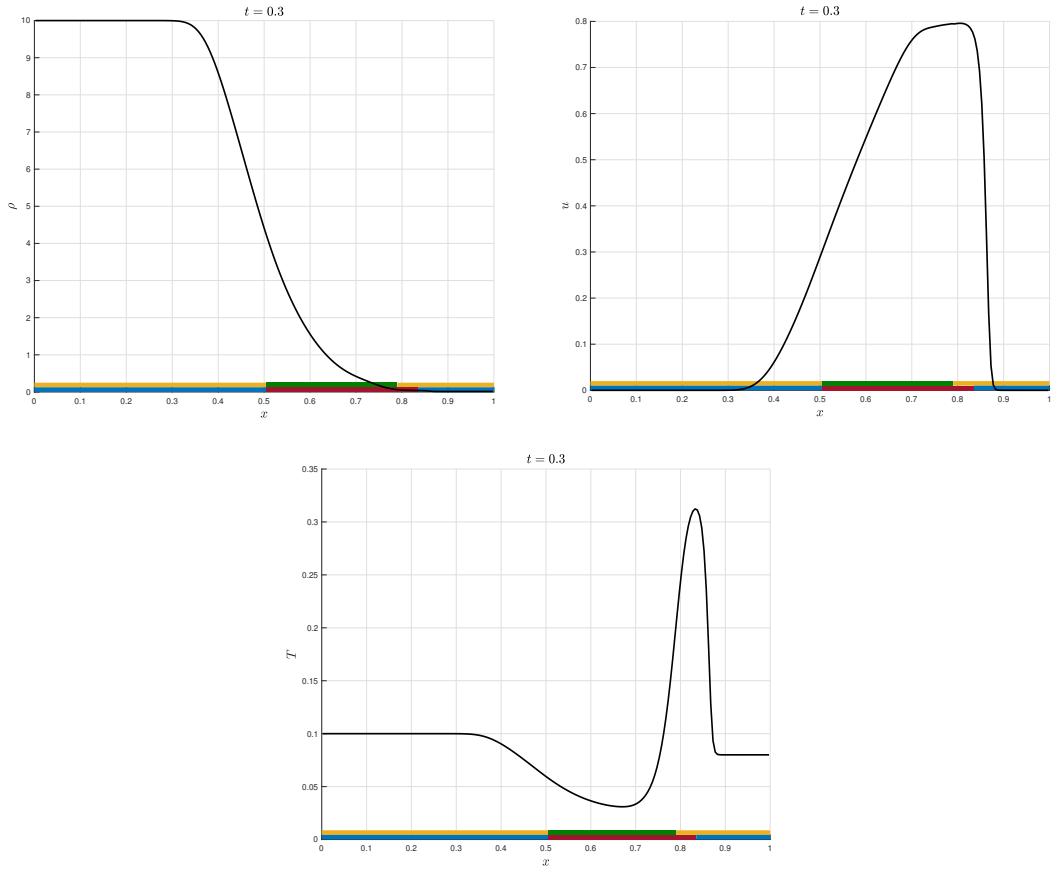


Figure 6: Décomposition du domaine. Les régions jaune et verte (resp. bleue et rouge) représentent les deux clusters pour ϕ (resp. ψ). La courbe noire représente la solution au temps final du modèle réduit local avec $N_{pod} = 15$ modes propres.

Cette modification est évaluée sur un test de reproduction d'une onde de choc à $Kn = 10^{-5}$. Sur la Figure 6, le domaine est décomposé en deux sous-domaines ($N_c = 2$). Comme attendu, dans le premier sous-domaine, l'onde de choc n'est que très peu voire pas du tout présente au cours du temps, et la solution peut être approchée par un faible nombre de modes propres ; tandis que dans le second sous-domaine, la réduction de la dimensionnalité est beaucoup plus limitée à cause de l'onde de choc qui se déplace. Sur la Figure 7, la précision des modèles réduits global ($N_c = 1$) et local ($N_c = 2$) est comparée en fonction du nombre de modes propres. Le modèle réduit local est plus précis que le modèle réduit global car les

modes propres locaux sont mieux adaptés pour approcher localement la solution.

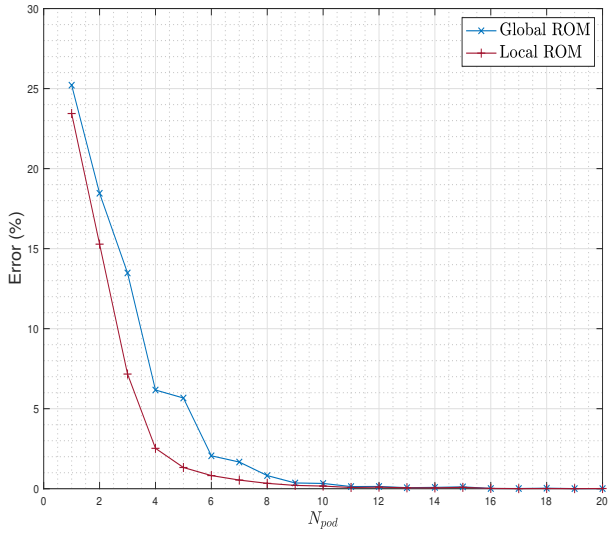


Figure 7: Comparaison de la précision des modèles réduits global et local.

Décomposition de domaine via la méthode de Galerkin discontinue

La dernière contribution de cette thèse [92] concerne le développement d'une méthode de décomposition de domaine [76, 72] basée sur la méthode de Galerkin discontinue [62, 7, 116] pour la modélisation d'ordre réduite. Dans cette approche, le modèle haute-fidélité résout le système dynamique où un certain degré de précision est requis, tandis que le modèle réduit est utilisé dans le reste du domaine.

Équations d'Euler

Dans ce travail, nous considérons l'écoulement de fluides compressibles et non-visqueux gouverné par les équations d'Euler :

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) = 0,$$

où $\mathbf{x} \in \Omega \subset \mathbb{R}^2$, $t \in \mathbb{R}^*_+$ et $\boldsymbol{\mu} \in \mathcal{D}$. Ici, $\mathbf{q} \in \mathbb{R}^4$ désigne la variable conservative et $\mathbf{F} = (\mathbf{f}, \mathbf{g})$ représente les flux :

$$\mathbf{q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix},$$

où ρ est la densité, $\mathbf{u} = (u, v)^T$ est la vitesse, E est l'énergie et p est la pression du fluide.

Modèle réduit

Dans le modèle réduit, chaque variable conservative q_i (c.-à-d. la densité, la quantité de mouvement et l'énergie) est approchée en espace par un faible nombre M_i de modes propres

$$\forall i \in \{1, \dots, 4\} \ : \ \widetilde{q}_i(\mathbf{x}, t; \boldsymbol{\mu}) = q_o^i(\mathbf{x}) + \sum_{n=1}^{M_i} a_n^i(t; \boldsymbol{\mu}) \Phi_n^i(\mathbf{x}),$$

où l'offset q_o^i et les modes propres Φ_n^i sont construits pendant la phase d'apprentissage, et les coordonnées réduites a_n^i sont déterminées au cours de l'étape de prédiction par la méthode de Galerkin discontinue.

Phase d'apprentissage

Lors de la phase d'apprentissage, des instantanés $s_l^i(\mathbf{x}) = q_i(\mathbf{x}, t_{k(l)}; \boldsymbol{\mu}_{j(l)})$ de la variable conservative q_i sont collectés à différents temps $t_{k(l)}$ et paramètres d'entrée $\boldsymbol{\mu}_{j(l)}$ afin de construire la base de données. L'offset q_o^i est ensuite définie comme la moyenne des instantanés :

$$q_o^i(\mathbf{x}) = \frac{1}{K} \sum_{l=1}^{K} s_l^i(\mathbf{x}),$$

et les modes propres Φ_n^i sont construits par POD à partir de la base de données :

$$\begin{cases} \underset{\Phi_1^i(\mathbf{x}), \dots, \Phi_{M_i}^i(\mathbf{x})}{\text{minimiser}} & \displaystyle\sum_{l=1}^{K} \int_\Omega \left(s_l^i(\mathbf{x}) - \widehat{s}_l^i(\mathbf{x}) \right)^2 \mathrm{d}\mathbf{x} \\ \text{tel que} & \displaystyle\int_\Omega \Phi_n^i(\mathbf{x}) \Phi_m^i(\mathbf{x}) \, \mathrm{d}\mathbf{x} = \delta_{n,m}, \end{cases}$$

où $\widehat{s}_l^i(\mathbf{x}) = q_o^i(\mathbf{x}) + \sum_{n=1}^{M_i} \left(\int_\Omega (s_l^i(\mathbf{y}) - q_o^i(\mathbf{y})) \Phi_n(\mathbf{y}) \, \mathrm{d}\mathbf{y} \right) \Phi_n(\mathbf{x})$. Finalement, les modes propres Φ_n^i sont dérivés de manière analytique afin d'obtenir leurs gradients $\nabla \Phi_n^i$, qui sont aussi requis dans la formulation de Galerkin discontinue.

Étape de prédiction

Comparées à la méthode de Galerkin discontinue classique, les fonctions de base polynomiales sont ici remplaçées par les modes propres POD, conduisant au système d'EDO suivant pour $i \in \{1, \dots, 4\}$ et $n \in \{1, \dots, M_i\}$:

$$\frac{\mathrm{d}a_n^i}{\mathrm{d}t} = \sum_{K \in \Omega} \left(\int_K F_i(\widetilde{\mathbf{q}}) \cdot \nabla \Phi_n^i \, \mathrm{d}\mathbf{x} - \int_{\partial K} \widehat{F_i}(\widetilde{\mathbf{q}}^-, \widetilde{\mathbf{q}}^+, \mathbf{n}) \Phi_n^i \, \mathrm{d}\boldsymbol{\sigma} \right),$$

où $\widehat{F_i}(\tilde{\mathbf{q}}^-, \tilde{\mathbf{q}}^+, \mathbf{n})$ désigne le flux numériques avec $\tilde{\mathbf{q}}^+$ et $\tilde{\mathbf{q}}^-$, la trace positive et négative de $\tilde{\mathbf{q}}$, respectivement, $\mathbf{n}$ désigne la normale sortante, et $\tilde{\mathbf{q}}^+ = \mathbf{q}_{bc}$ au bord. Par rapport à la méthode de Galerkin classique, les intégrales aux faces supplémentaires permettent, d'une part, d'imposer les conditions aux bords dans un sens faible et, d'autre part, d'introduire de la diffusion/dissipation numérique à travers le flux numérique pour stabiliser le modèle réduit. De plus, afin de réduire le coût de calcul des intégrales, celles-ci sont évaluées par la méthode d'hyper-réduction ECSW, qui définit une méthode d'intégration numérique empirique où l'intégrande n'a besoin d'être évalué qu'en un faible nombre de points $\mathbf{x}$ ou $\boldsymbol{\sigma}$.

Décomposition de domaine

Le modèle réduit développé précédemment offre une manière simple de mettre en oeuvre la décomposition de domaine. Dans cette approche, le domaine est décomposé en micro- et macro-cellules comme illustré sur la Figure 8. Le modèle haute-fidélité décrit la dynamique du fluide dans les micro-cellules K_j, tandis que le modèle réduit approxime la solution dans les macro-cellules Ω_j. La restriction de la solution sur chaque macro-cellule Ω_j est approchée par

$$\forall i \in \{1, \dots, 4\},\ \forall \mathbf{x} \in \Omega_j\ :\ \tilde{q}_i(\mathbf{x}, t; \boldsymbol{\mu}) = q_o^{i,j}(\mathbf{x}) + \sum_{n=1}^{M_{i,j}} a_n^{i,j}(t; \boldsymbol{\mu}) \Phi_n^{i,j}(\mathbf{x}),$$

où l'offset $q_o^{i,j}$ et les modes propres $\Phi_n^{i,j}$ sont construits de la même manière que précédemment. Les coordonnées réduites vérifient maintenant le système d'ODE

$$\frac{\mathrm{d}a_n^{i,j}}{\mathrm{d}t} = \sum_{K \in \Omega_j} \left(\int_K F_i(\tilde{\mathbf{q}}) \cdot \nabla \Phi_n^i \, \mathrm{d}\mathbf{x} - \int_{\partial K} \widehat{F_i}(\tilde{\mathbf{q}}^-, \tilde{\mathbf{q}}^+, \mathbf{n}) \Phi_n^i \, \mathrm{d}\boldsymbol{\sigma} \right),$$

où les intégrales sont calculées par la méthode ECSW afin de réduire le coût de calcul des intégrales. De cette manière, la solution globale est reconstruite en raccordant les solutions locales à travers les flux numériques à l'interface des micro- et macro-cellules.

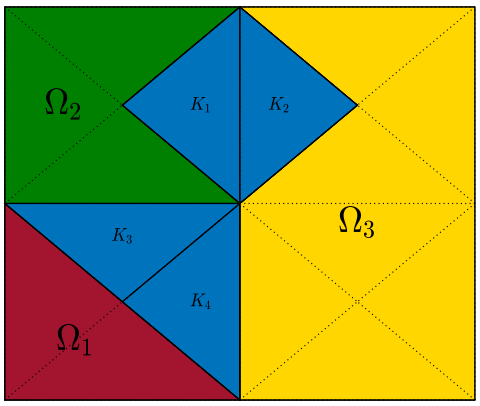


Figure 8: Exemple de décomposition de domaine en 4 micro-cellules et 3 macro-cellules.

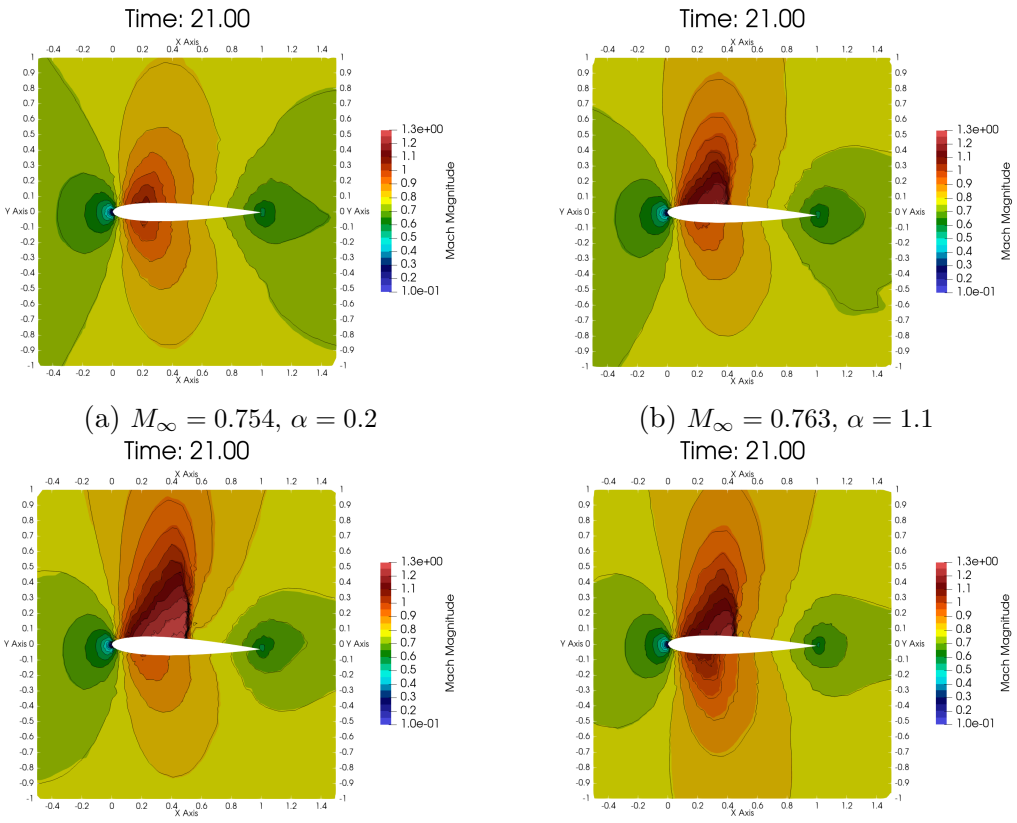
Prédiction d'un écoulement transsonique

La décomposition de domaine basée sur la méthode de Galerkin discontinue est évaluée sur sa capacité à prédire un écoulement transsonique autour d'un profil d'aile NACA 0012. La solution dépend des paramètres d'entrée $\boldsymbol{\mu} = (M_\infty, \alpha)$ correspondant à différents nombres de Mach en entrée M_∞ et angles d'attaque α. Un écoulement uniforme à Mach M_∞ est imposé sur les bords du domaine

$$\forall \boldsymbol{\sigma} \in \partial\Omega \ : \ \rho_{bc}(\boldsymbol{\sigma};\boldsymbol{\mu}) = 1, \ u_{bc}(\boldsymbol{\sigma};\boldsymbol{\mu}) = M_\infty, \ v_{bc}(\boldsymbol{\sigma};\boldsymbol{\mu}) = 0, \ T_{bc}(\boldsymbol{\sigma};\boldsymbol{\mu}) = 1,$$

et une condition de glissement est appliquée sur le profil d'aile. Le domaine Ω est décomposé en deux régions : le modèle haute-fidélité est utilisé autour de l'aile afin de représenter précisément l'onde de choc, tandis que le modèle réduit est employé dans le reste du domaine pour approximer la solution.

Les solutions prédites à l'état d'équilibre sont présentées sur la Figure 9, et les performances de la décomposition de domaine sont données à la Figure 10. Lorsque $M = 7$, l'erreur de prédiction est inférieure à 1% pour tous les tests, et le temps de calcul est réduit de 78% par rapport aux simulations haute-fidélité.



(a) $M_\infty = 0.754$, $\alpha = 0.2$ (b) $M_\infty = 0.763$, $\alpha = 1.1$

(c) $M_\infty = 0.776$, $\alpha = 1.8$ (d) $M_\infty = 0.784$, $\alpha = 0.6$

Figure 9: Nombre de Mach de la solution approchée correspondante à $M = 16$ pour la prédiction d'un écoulement autour d'un profil d'aile NACA 0012.

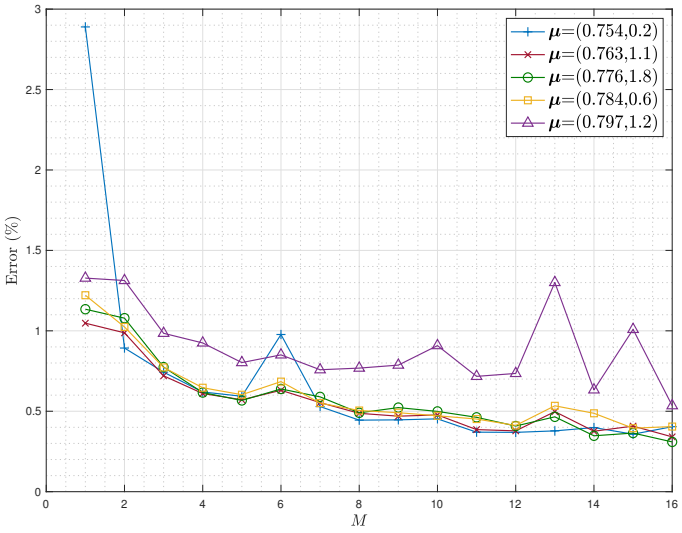


Figure 10: Précision de la méthode DGDD pour la prédiction d'un écoulement autour d'un profil d'aile NACA 0012 en fonction de la taille de la base réduite.

Conclusion

Dans cette thèse, nous avons tout d'abord développé un modèle réduit pour la simulation d'écoulements gazeux dans les régimes raréfié et hydrodynamique. Les tests ont démontré la précision et l'efficacité du modèle réduit, avec une erreur inférieure à 1% et un temps de calcul divisé par environ 45 par rapport aux simulations haute-fidélité en utilisant 20 modes propres.

Ensuite, deux améliorations pour le modèle réduit précédent, basées sur le problème de transport optimal, ont été proposées. La première amélioration porte sur l'enrichissement de la base de données avec des nouveaux instantanés artificiels interpolés par transport optimal. Les tests ont démontré que ces instantanés artificiels amélioraient la fiabilité du modèle réduit dans le cas d'un sous-échantillonnage de la solution à prédire. La seconde amélioration consiste à partitionner le domaine par une méthode de classification non-supervisé couplée à la distance de Wasserstein, puis à approximer la solution dans chaque sous-domaine par différentes bases réduites locales. Les tests ont montré que cette modification améliorait la précision du modèle réduit.

La dernière contribution visait à développer une méthode de décomposition de domaine basée sur la méthode de Galerkin discontinue pour la modélisation d'ordre réduite. Dans cette approche, le modèle haute-fidélité résout le système d'équations où un certain degré de précision est requis, tandis que le modèle réduit est utilisé dans le reste du domaine. Les tests ont démontré les performances de la décomposition de domaine, avec une erreur inférieure à 1% et un temps de calcul réduit de 78% par rapport aux simulations haute-fidélité en utilisant 7 modes propres.