





# Acknowledgements

I am grateful to the many wonderful individuals who have contributed to this thesis work, and I couldn't have accomplished it without their support. Firstly, I want to express my profound gratitude to my supervisors, Domenico Borzacchiello and Luisa Silva, who have been constant guiding forces throughout my Ph.D. journey. Both of them have provided me with invaluable insights and direction from the beginning to the end of my thesis. Domenico's ability to explain complex topics in a clear and understandable way is highly appreciated, and I will always be thankful to him for his support and for helping me acquire the knowledge and research skills required to complete this work. Luisa's attention to detail and invaluable input helped me improve my work and iterate until it reached its optimal form. I admire both of them for their wisdom, extensive knowledge of the subject, patience, understanding, and trust in me, and will always treasure their guidance.

I am also deeply appreciative of my CSI committee members, Patrice Laure and Gaetano D'Avino, for their valuable discussions and suggestions, which helped me improve my work and broaden the scope for future research. Gaetano, in particular, co-authored an article with me, and his unwavering support throughout my work is highly appreciated. I extend my sincere thanks to my jury committee members, Marianne Beringhier, Julien Férec, and Christophe Binetruy, for serving on my thesis committee and for their thoughtful criticism and comments regarding this thesis.

A big shout-out to Hughes Dignonnet, with whom I collaborated for a portion of my thesis, and also assisted him with some teaching activities together. I sincerely learned so much from him, and his engaging and helpful discussions are highly valued. I would also like to thank Françoise Foucher for entrusting me with teaching opportunities throughout my Ph.D. and for her unwavering belief in me. Furthermore, I extend my appreciation to Lucas Lestandi, with whom I had the privilege of co-authoring another article. His insightful comments and discussions have made a significant contribution to my work, and his constant support is highly valued.

I owe a debt of gratitude to Pierre Emmanuel for all the technical assistance throughout my Ph.D. journey. For all the times that I almost broke my computer, he was there to help. I would like to express my appreciation to all my friends in and out of the lab for their constant support through both tough times and good times. The list is long, and I appreciate each and every one of them. I would like to give a special shoutout to my longtime friend Kiran Kollepara, who was also a former Ph.D. from the same lab. His insights and discussions have been instrumental to my work, and I am deeply grateful for his help.

Finally, I want to express my love and appreciation to my family, my parents: Dig B. Chetry and Saraswati Chetry, and my sister Manna Massey, for their unwavering love, support, and trust in me. Without them, I would not have been able to pursue my dreams. To my partner, Dhruv Jain, you were my support system during my Ph.D., and I cannot thank you enough for being there for me. This work is dedicated to them. Thank you all from the bottom of my heart for being a part of my journey.

# Contents

<b>Acknowledgements</b>	<b>i</b>
<b>Contents</b>	<b>ii</b>
<b>Abstract</b>	<b>iv</b>
<b>List of Figures</b>	<b>v</b>
<b>List of symbols</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Review of dimensionality reduction techniques</b>	<b>7</b>
2.1 General problem definition . . . . .	7
2.1.1 Discretization space . . . . .	8
2.1.2 Affine parameter dependence . . . . .	9
2.2 Model order reduction . . . . .	9
2.2.1 Main features . . . . .	9
2.2.2 Reduced basis generation via classical POD . . . . .	12
2.2.3 Modified POD/SVD techniques . . . . .	13
2.2.4 Greedy RBM . . . . .	15
2.3 Reduction of non-linear terms . . . . .	16
2.3.1 Interpolation based approach . . . . .	17
2.3.2 Integral approximation approach . . . . .	18
2.3.3 Reduced quadrature rule based on sparse recovery technique: non- negative FOCUSS algorithm . . . . .	20
<b>3 Reduced order modeling framework for viscoelastic fluid flows</b>	<b>21</b>
3.1 Part I: Fundamentals of rheology . . . . .	21
3.1.1 Introduction . . . . .	21
3.1.2 Non-Newtonian fluids . . . . .	22
3.1.3 Viscoelastic fluids . . . . .	23
3.2 Part II: Reduced order modeling for viscoelastic flows . . . . .	26
3.2.1 Motivation and background . . . . .	26
3.2.2 Problem setting of a viscoelastic fluid flow . . . . .	28
3.2.3 Model order reduction methodology . . . . .	30
3.3 Numerical analysis of two benchmark problems. . . . .	35
3.3.1 Posteriori error metrics . . . . .	36
3.3.2 Description of the problems . . . . .	36
3.3.3 Discussions . . . . .	38
3.4 Conclusions . . . . .	43

<b>4</b>	<b>Iterative multi-fidelity modeling for multi-dimensional input parametric PDE systems</b>	<b>45</b>
4.1	<b>Introduction</b>	45
4.1.1	Motivation and background	45
4.1.2	Overview of the idea: Iterative multi-fidelity model order reduction	46
4.2	Iterative multi-fidelity modeling (IMF) for building PODG-ROM	48
4.2.1	Construction of the initial low-fidelity model	50
4.2.2	Parametric point selection	50
4.2.3	Recovery of the reduced basis functions	52
4.2.4	Updating low-fidelity model	52
4.2.5	Error metrics	53
4.3	Numerical example of a heat conduction problem	54
4.3.1	Results and discussions	55
4.4	Numerical example of an advection-diffusion problem	57
4.4.1	Problem setting	58
4.4.2	Results and discussions	60
4.5	Numerical example of flow past a sphere problem	63
4.6	Conclusions	66
<b>5</b>	<b>POD-ROM for spatially adapted snapshots</b>	<b>69</b>
5.1	Introduction	69
5.2	Problem description	70
5.2.1	Numerical analysis	71
5.3	Mesh adaptation	73
5.3.1	Mesh topology	73
5.3.2	Mesh optimization	74
5.3.3	Extension to parametric field adaptation	76
5.4	Part 1: Iterative multi-fidelity modeling (IMF) for building PODG-ROM for spatially adapted snapshots	76
5.5	Part 2: Recovery of the adaptive basis functions	78
5.6	Results and discussions	79
5.6.1	Numerical analysis on mesh adaptation	79
5.6.2	Numerical analysis on iterative multi-fidelity reduced-order modeling	83
5.6.3	Numerical analysis on adaptive reduced basis	84
5.7	Conclusions	85
<b>6</b>	<b>Conclusions and perspectives</b>	<b>89</b>
<b>A</b>	<b>Lower bound estimation of the coercivity constant</b>	<b>93</b>
	<b>Bibliography</b>	<b>97</b>

## Abstract

The subject of this thesis concerns the model order reduction of parameterized non-Newtonian fluid flows. For the purpose of simulating complex flows, high-performance computing (HPC) is a widely used predictive tool. However, HPC systems are challenging to use for real-time interactive scientific computing because of limitations on simulation time and heavy resource usage. The methods that are frequently used to accelerate calculations are always founded on the presumptions such as the existence of low-dimensional representation bases, which result in low-level numerical algorithms. These machine learning approaches, known as reduced order models (ROM), have been shown to be reliable and robust. However, due to the "invasive" nature of their implementation, they continue to have significant downsides.

The main objective of the thesis is to deploy model order reduction (MOR) methods that are less intrusive to solve complex problems such as parameterized viscoelastic fluid flows that have significant industrial applications. Traditional MOR methods restrict the computational performance of such highly nonlinear problems due to their dependency on the original dimension of the system, so we propose a state-of-the-art hyper-reduction technique based on a sparse approximation to tackle the evaluation of nonlinear terms at very reduced complexity and which is also less intrusive. We also provide an offline stabilization for the constitutive model in the reduced order model framework that is less expensive to compute online while maintaining the accuracy of the full order model. Combining the two significantly lowers the CPU cost as compared to FOM evaluation, which inevitably boosts the MOR performance. We validate the suggested approach on two flow benchmark problems. In addition, an adaptive parametric sampling strategy for multi-dimensional parametric PDE systems is presented in this work which is achieved by leveraging multi-fidelity model approximation, and further extending to the viscoelastic fluid flow problem. This technique avoids the need for prerequisite residual-based error estimators.

Towards the end of the thesis, we address another issue that is typically observed for cases when adaptive finite element meshes are deployed. In such cases, MOR methods fail to produce a low-dimensional representation since the snapshots are not vectors of the same length. We, therefore, suggest an alternate method that can generate reduced basis functions for a database of space-adapted snapshots, and demonstrate the technique using a numerical example of a diffusion-reaction problem by altering the reaction coefficient over an extensive range of values.

**Keywords:** *Viscoelastic flows, reduced-order modeling, hyper-reduction, stabilization, adaptive parameter sampling, multi-fidelity modeling, space-adapted snapshots.*

# List of Figures

1.1	Everyday viscoelastic materials can be found on a spectrum, with the majority falling somewhere between fully viscous and entirely elastic (image source: google).	1
1.2	Homogenization of a fiber reinforced composite material.	2
2.1	Low-dimensional manifold in which the field variable resides and approximation of the solution at $\mu_{new}$ by the linear combination of the precomputed solutions.	10
3.1	A qualitative curve representation of typical Newtonian and non-Newtonian fluids [1].	22
3.2	Schematic representation of a polymer chain at rest and under the influence of shear stress [1].	24
3.3	A polymer chain represented by a random walk of step length ‘ $r_i$ ’ in an elastic dumbbell model.	25
3.4	( <i>left</i> )Schematics of the flow past a sphere benchmark, ( <i>bottom</i> ) computational domain.	36
3.5	Mesh representation	37
3.6	Schematics of the step flow benchmark.	37
3.7	( <i>Top</i> ) Computational domain, ( <i>bottom</i> ) mesh representation	38
3.8	( <b>a</b> ) 2D velocity magnitude profile , and ( <b>b</b> ) log conformation tensor $\mathbf{S}_{zz}$ profile in axial direction for $\lambda = 0.5667$ for flow past the sphere benchmark.	38
3.9	( <b>a</b> ) 2D Velocity magnitude profile and ( <b>b</b> ) log conformation tensor $\mathbf{S}_{zz}$ profile in axial direction for $\lambda = 0.5667$ for 4:1-step flow benchmark.	38
3.10	ROM error of the solution $\mathbf{S}$ for the benchmark cases: flow past a sphere ( <i>left</i> ), step flow with sharp corner ( <i>middle</i> ) and step flow with rounded corner ( <i>right</i> ).	39
3.11	ROM error of the solution $\mathbf{S}$ for different rank of $\tau_e$ ( <i>left</i> ), location of DEIM points and its nature of corresponding weights ( <i>right</i> ).	40
3.12	( <i>left</i> ) ROM error of the solution $\mathbf{S}$ using reduced quadrature rule for varying $\gamma$ values of Benchmark 1, ( <i>right</i> ) location of reduced quadrature points.	41
3.13	( <i>left</i> ) ROM error of the solution $\mathbf{S}$ using reduced quadrature rule for varying $\gamma$ values of Benchmark 2, ( <i>right</i> ) location of reduced quadrature points.	42
3.14	POD projection error and ROM error of $\mathbf{S}$ with offline stabilization only (POD-PG) with and without hyper reduction for both the benchmark cases.	42
3.15	Computational speed ups obtained without hyper reduction using POD-PG ROM and POD-SUPG ROM, and POD-PG ROM with hyper reduction for both the benchmark cases.	42
4.1	Multi-fidelity for outer loop applications	47
4.2	Flow of the iterative multi-fidelity modeling approach.	49
4.3	Geometry of heat conduction problem	54
4.4	Two different representative solutions for the parameterized conductivity problem.	55
4.5	Error $\epsilon_{train}$ and $\epsilon_{val}$ between FOM and reconstructed ROM solution	55

4.6	Parametric modes represented by blue and red curves along with its residual represented by black dotted lines, and DEIM points obtained during each loop of the proposed method. . . . .	57
4.7	(a) a coarse discretized mesh for an initial low-fidelity model, and (b) a fine discretized mesh for high-fidelity model approximations. . . . .	58
4.8	Sampled parametric points using two sketch models (a), Gaussian distribution of sampling points over 10 trials (b). . . . .	58
4.9	Error $\epsilon_{\text{POD}}$ of the classical POD projection, posteriori error bound by greedy RBM and error $\epsilon_{\text{ROM}}$ of the solution of the reduced order model obtained by the proposed method (a), and sampled parametric points using the proposed method, classical POD and greedy RBM (b). . . . .	59
4.10	(left) Geometrical set up of advection-diffusion problem in a 9 block system and (right) advective flow field. . . . .	59
4.11	Three different representative solutions for the advection-diffusion problem shown (below) by varying nine different combinations of diffusion coefficient $\mathbb{K}(\mu)$ in all the three domains (top). . . . .	61
4.12	(a,b) Mean of error $\epsilon_{\text{train}}$ and $\epsilon_{\text{val}}$ between FOM and ROM solution obtained using different random sketch models over 10 trials and its std. deviation. (c) Error $\epsilon_{\text{train}}$ and $\epsilon_{\text{val}}$ between FOM and ROM solution for one trial. . . . .	62
4.13	(a) Coarse discretized mesh for initial low-fidelity model, and (b) fine discretized mesh for high-fidelity model approximations. . . . .	63
4.14	Error $\epsilon_{\text{train}}$ and $\epsilon_{\text{val}}$ between FOM and ROM solution using coarse sketch model (a). Error $\epsilon_{\text{POD}}$ of the classical POD projection, and error $\epsilon_{\text{ROM}}$ of the solution of the reduced order model obtained by the proposed method (b). . . . .	63
4.15	Error $\epsilon_{\text{train}}$ and $\epsilon_{\text{val}}$ between FOM and reconstructed ROM solution. . . . .	64
4.16	(a) a coarse discretized mesh for an initial low-fidelity model, and (b) a fine discretized mesh for high-fidelity model approximations. . . . .	64
4.17	Sampled parametric points in each iteration. . . . .	65
4.18	POD modes obtained by classical SVD (a) and by proposed IMF approach (b). . . . .	66
5.1	Uniform fine mesh representation ( $\#Nodes = 316714$ ) . . . . .	72
5.2	Solution profile for different values of $\lambda$ . . . . .	72
5.3	Illustration of the edge vector $\mathbf{x}_{ij}$ of the edge joining nodes i and j, and the edge solution, $\mathbf{u}_{ij}$ joining the solution at the nodes i and j. . . . .	74
5.4	Flow of the proposed method . . . . .	77
5.5	Representative solution with adaptive meshes . . . . .	80
5.8	Zoom in on the edge of the mesh representing $\lambda = 1000$ . . . . .	82
5.9	(a) Illustration of commonly adapted mesh ( $\#Nodes = 18598$ ) and (b) zoom in on the left edge of the mesh. . . . .	82
5.10	Stretch factor of the mesh elements of the commonly adapted mesh. . . . .	83
5.11	Illustration of collocation points generated at every iteration of the proposed method. . . . .	84
5.12	(a) Sampled parametric points and (b) Error $\epsilon_{\text{POD}}$ of the POD projection, and error $\epsilon_{\text{ROM}}$ of the solution of the reduced order model without space adaptation (classical) and with space adaptation (IMF). . . . .	85
5.13	Illustration of commonly adapted finite element meshes that are representative of a linear combination of 4 different adaptive snapshots. . . . .	86
5.14	Mode 1 obtained by classical POD (a), and (b) by iterative multi-fidelity approach. (c) represents spatially adaptive mode 1. . . . .	86
5.15	Mode 2 obtained by classical POD (a), and (b) by iterative multi-fidelity approach. (c) represents spatially adaptive mode 2. . . . .	87

5.16	Mode 3 obtained by classical POD <b>(a)</b> , and <b>(b)</b> by iterative multi-fidelity approach. <b>(c)</b> represents spatially adaptive mode 3. . . . .	87
5.17	Mode 4 obtained by classical POD <b>(a)</b> , and <b>(b)</b> by iterative multi-fidelity approach. <b>(c)</b> represents spatially adaptive mode 4. . . . .	87
A.1	Illustration of the estimated lower bounds for the coercivity constant, for example presented in section 4.3. . . . .	94



# List of symbols

## Mathematical spaces

$\mathbb{R}$	Real space
$\mathcal{D}$	Parametric Space
$\mathcal{N}, \mathcal{T}$	Mesh topology
$\mathcal{M}, \mathbb{M}$	Manifold, metric tensor
$\mathcal{V}, \mathcal{V}, \mathcal{P}, \mathcal{G}, \mathcal{I}$	Functional spaces

## Physical quantities

$\mathbf{u}$	Velocity field
$p$	Hydrostatic pressure
$\boldsymbol{\sigma}$	Cauchy stress tensor
$\mathbf{D}$	Strain rate tensor
$\dot{\gamma}$	Equivalent shear rate
$\mathbf{G}$	Velocity gradient tensor
$\boldsymbol{\tau}_e$	Viscoelastic stress tensor
$\mathbf{C}$	Conformation tensor
$\mathbf{S}$	Log-conformation tensor
$\lambda$	Relaxation time of viscoelastic fluid
$\mathbf{b}$	Advection flow field
$\sigma$	reaction term

## Physics constants

$\eta_s$	Solvent viscosity
$\eta_p$	Polymer viscosity
$\alpha$	Giesekus fluid mobility parameter
$\epsilon, \xi$	PTT fluid extensibility, slip parameters
$\mathbb{K}$	Diffusivity coefficient

$\kappa$	Thermal conductivity
$\lambda$	Reaction coefficient
<b>Other symbols</b>	
$\mathbf{x}$	Spatial coordinates
$t$	Time
$\Omega$	Domain
$\mu$	Input parameter
$N, N_{hf}$	Number of snapshots
$\alpha$	Coercivity constant
$\Xi_{train}$	Training set
$\Xi_{val}$	Validation set
$u, \phi$	A field variable and its POD mode
$\lambda, \sigma$	Eigenvalue, singular value
$\Delta^R$	Posteriori error indicator
$\delta_g, \epsilon$	Truncation tolerance, stopping threshold
$Wi$	Weissenberg Number
$De$	Deborah Number
$Pe$	Péclet Number
$\beta$	Dimensionless constant
$\phi$	POD modes of velocity field
$z$	Reduced coefficient of velocity field
$\Theta$	POD modes of log-conformation tensor
$w$	Reduced coefficient of log-conformation tensor
$\xi$	POD modes of viscoelastic stress tensor
$\Psi$	POD modes of test function of the constitutive law
$r, r'$	Truncated order of the POD modes of the velocity field, log-conformation tensor
$m$	Interpolative coefficients
$\mathbb{Q}, \mathbb{E}$	Interpolative matrix
$\mathcal{N}_q, \omega$	Gaussian quadrature points and its corresponding weights
$\gamma$	Relaxation parameter

The same symbol may be used in different parts with different meanings.

## General Notations

- The scalars are represented by italic letters such as  $x$ .
- The vectors are represented by bold small letters such as  $\mathbf{x}, \mathbf{x}$ .
- The tensors of order 2 are represented by bold capital letters such as  $\mathbf{X}$  \*

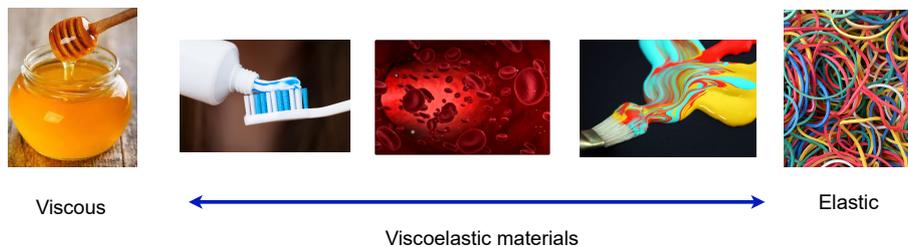
---

\*An exception in chapter 1: tensor of order 2 is represented by a small bold letter and tensor of order 4 by capital bold letter.

# Chapter 1

## Introduction

Significant enhancements in computer hardware and advances in numerical modeling over the past few decades have led to a monumental impact on resolving complex physical and engineering problems, thereby rapidly increasing the growth of industrial projects. However, many problems in computational mechanics still remain intractable despite remarkable development in high-performance computing (HPC) and numerical modeling. One such example is the numerical simulation of particle suspensions in viscoelastic fluid flows [2-4]. Studies of the rheology of such fluids have seen a significant scientific contribution given its wide industrial applications ranging from pharmaceutical to drug delivery in medical science [5], personal care and cosmetics [6], composite manufacturing [7], food products [8], to name a few. Figure 1.1 shows some regular viscoelastic materials that combine attributes of both elastic deformation and a dissipative viscous component. Often, industrial applications require multi-query predictive simulation for both process and product design, but also for monitoring and control applications. Indeed, the exploration of the parametric space is a fundamental task in applications like optimization, model-based control, system identification, uncertainty quantification, and contingency analysis.

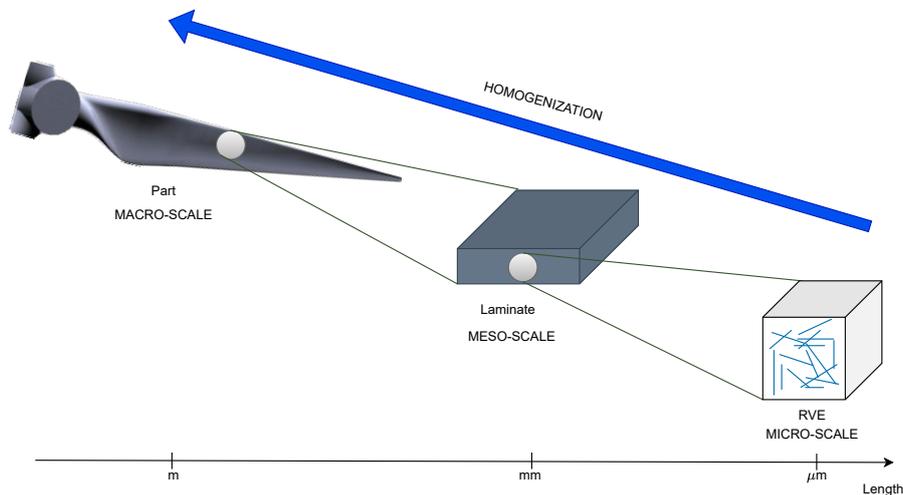


**Figure 1.1:** Everyday viscoelastic materials can be found on a spectrum, with the majority falling somewhere between fully viscous and entirely elastic (image source: google).

The computational burden of such complex physics simulations is potentially very high, which will be discussed shortly, and even the usage of HPC and massively parallel computing resources might impose severe restrictions on the use of high-fidelity models in this framework. On the other hand, surrogate modeling has proven to be an effective approach to alleviating the intrinsic complexity of multi-parametric analysis by providing quick and inexpensive parametric estimators compared to high fidelity models [9, 10]. In this scenario, model order reduction (MOR) provides a robust and reliable approach to building parametric surrogate models from a high-fidelity model by reducing the complexity of the full order model (FOM) without significantly compromising on its accuracy, thus potentially bridging the gap between HPC and surrogate modeling.

Particle-laden fluids involve multi-scale level modeling: The microscopic scale modeling

is related to the kinematic motion of the individual particles, in the mesoscopic scale a cluster of particles at a material point is studied, and finally, the macroscopic scale is related to the forming process and the final part itself [11]. As early as 1922, the motion of a single ellipsoid particle immersed in an incompressible Newtonian fluid was well-predicted thanks to Jeffrey [12]. Since then, many theoretical models have been developed to study dilute and semi-dilute suspensions on Newtonian fluid, a survey of which can be found in [13]. One can also find in the literature on direct numerical simulation (DNS) to study the particle kinematics [14–19]. However, due to the continuous evolution of the particle position and orientation and also due to its particle-particle interactions and interaction with fluids, DNS is not currently tractable even with the most powerful computing resources. As it requires full-scale simulation of the flow around the particle at each time step, most of the research based on DNS remains qualitative and conducted on a small representative volume as shown in figure 1.2 and far from actual configurations. The bulk properties of the system are then predicted by some numerical homogenization [20] or volume averaging [21] up-scaling techniques. In DNS, the hydrodynamic forces and torques on the particles are not modeled, but determined in the post-processing from the computed motion of the fluid to calculate the particle trajectories.



**Figure 1.2:** Homogenization of a fiber reinforced composite material.

This calls for less detailed mesoscopic models such as those of kinetic theory that results from coarsening of microscopic descriptions [22, 23]. It involves a probability distribution function (PDF),  $\psi(\mathbf{x}, \mathbf{p}, t)$ , that allows describing more widely the state of orientation of a population of particles at a material point since it is not feasible to study the motion of each particle due to the aforementioned reasons. The PDF presents at a time and position the probability of finding the particles oriented in the direction  $\mathbf{p}$  and is governed by the Fokker-Planck equation,

$$\frac{D}{Dt}\psi = -\frac{\partial}{\partial \mathbf{p}} \cdot (\psi \dot{\mathbf{p}}), \quad (1.1)$$

where  $D/Dt$  is the material time derivative and  $\dot{\mathbf{p}}$  is the time evolution for the particles. For concentrated suspension where hydrodynamic interactions and contacts take place between neighboring particles, Folgar and Tucker [24] gave the following expression for  $\dot{\mathbf{p}}$  where,

$$\dot{\mathbf{p}} = -\frac{1}{2}\boldsymbol{\omega} \cdot \mathbf{p} + \frac{1}{2}\lambda(\dot{\boldsymbol{\gamma}} \cdot \mathbf{p} - \dot{\boldsymbol{\gamma}} : \mathbf{p} \otimes \mathbf{p} \otimes \mathbf{p}) - \frac{D_r}{\psi} \frac{\partial \psi}{\partial \mathbf{p}}, \quad (1.2)$$

where  $\boldsymbol{\omega}$  and  $\dot{\boldsymbol{\gamma}}$  are the vorticity and deformation rate tensors respectively, and  $\lambda$  is related to the aspect ratio of the particle. The last term in the right-hand side of (1.2) with  $D_r$  is a rotary diffusivity term, an addition to the original work of Jeffery [12] that takes into account the effect of particle interactions.

This mesoscopic approach is a reasonable compromise between the detailed microscale modeling and the coarser macroscopic approach discussed below. However, the Fokker-Planck equation (1.1) involves a high dimensional approximation space as it is solved in both the physical and conformation spaces and in time, therefore suffers from what is known as the curse of dimensionality and hence, cannot be solved by means of standard mesh-based approximation techniques. Numerical strategies based on the use of particles have been extensively developed by many authors [25–27]. Based on the separated representation of variables within the reduced basis approximations have been presented for directly solving the Fokker-Planck equation in conformation spaces of high dimension [26]. On the other hand, there are few works on the solution of the Fokker-Planck equation with standard discretization techniques that can be referred to in the articles [28–30].

For the efficiency of the computation, orientation tensors [31,32] which resort to closure approximations have been widely accepted, which enables a compact representation of the orientation state in the macroscopic scale,

$$\mathbf{a} = \oint \mathbf{p} \otimes \mathbf{p} \psi(\mathbf{p}) d\mathbf{p} \quad (1.3)$$

$$\mathbf{A} = \oint \mathbf{p} \otimes \mathbf{p} \otimes \mathbf{p} \otimes \mathbf{p} \psi(\mathbf{p}) d\mathbf{p} \quad (1.4)$$

$\mathbf{a}$  and  $\mathbf{A}$  are second and fourth-order orientation tensors, respectively. The equations governing the time evolution of  $\mathbf{a}$  are obtained by taking the time derivative of eq. (1.3), expressing  $\dot{\psi}$  from eq. (1.1) and then integrating by parts. This procedure yields,

$$\dot{\mathbf{a}} = \oint (\dot{\mathbf{p}} \otimes \mathbf{p} + \mathbf{p} \otimes \dot{\mathbf{p}}) \psi(\mathbf{p}) d\mathbf{p} \quad (1.5)$$

which leads to,

$$\dot{\mathbf{a}} = \nabla \mathbf{u} \cdot \mathbf{a} + \mathbf{a} \cdot (\nabla \mathbf{u})^T - 2\mathbf{A} : \nabla \mathbf{u} \quad (1.6)$$

It can be seen that the fourth-order orientation tensor appears in the equation (1.6). So the evolution of the fourth-order tensor in the Folgar and Tucker equation is a function of the sixth-order tensor and so forth. Thus, a closure approximation is required where a low even order particle orientation tensor depends on the tensor of the next higher even order. Different closure approximations have been proposed in the literature [31,33,34] but often result in inaccuracy and loss of information [35], and hence care must be taken while working with these models. However, this method delivers a significant decrease in the necessary computing expenses, which is appreciated in engineering applications.

The vast majority of the work for multi-scale level modeling for orientation prediction in the literature is based on the Jeffery theory that describes the motion of an ellipsoidal particle immersed in a Newtonian fluid, and there is no existing close analytical form to predict  $\dot{\mathbf{p}}$ , a general Jeffery counterpart available for suspending fluids of second order at least in macroscopic scale taking into account hydrodynamic interactions. The complexity of particle-filled fluids manifolds when the suspending medium is a viscoelastic fluid, which is the case often for industrial applications. The rheology of viscoelastic fluids is complex in nature with the governing equations being strongly coupled, hyperbolic, and having stability issues, for which additional degrees of freedom are added to an already high dimensional problem. The fluid displays viscoelastic properties which cause extra stress tensor, along with shear thinning or thickening behavior, so essentially they are

constitutively highly nonlinear in a stress-strain relationship. This can further increase the complexity of multi-scale modeling using DNS.

Few authors have explored the effects of the suspending fluid’s non-Newtonian rheology on the motion of a single particle under simple shear flow [17, 36–42]. In the late 1970s, Leal [43] and Brunn [44] derived the equations governing the motion of rods and transversely isotropic particles respectively that could be viewed as the counterparts of the Jeffery equation for the second-order viscoelastic fluids in the limit of low Weissenberg numbers. Domenico et al. [45] have addressed modeling of short fiber suspensions in second-order fluids at various description scales, from microscopic to macro-scale. Based on Brunn’s model [44], the authors demonstrated a macroscopic description for viscoelastic fluids, however, it turns out that there are sixth and eighth-order tensors, for which there isn’t currently a good closure approximation. Accordingly, the authors proposed a condensed modeling framework that makes use of an effective velocity gradient that is created by expanding the Giesekus constitutive equation asymptotically for low Weissenberg numbers. This effective velocity gradient is then injected into the equation of the ellipsoid motion, resulting in the extended Jeffery model and its macroscopic equivalent, the Folgar and Tucker model.

However, it is very important to precisely simulate the rheological characteristics of such fluid flows in order to forecast an efficient and reliable outcome on an industrial scale. It is vital to maintain the physics of the issue since more complex fluid dynamics are seen in viscoelastic suspensions, particularly as the typical characteristic flow rate rises [46]. For a precise result, it is, therefore, necessary to research the effect of high Weissenberg numbers, focusing on simulating concentrated suspensions in 3D while taking into account all hydrodynamic forces and accounting for contact interactions between the particles. The difficulty in obtaining high-order moments of the distribution function in a macroscopic scale, which necessitates precise closure relations, as well as the enormous computational cost involved in calculating the forces acting on the particle surface and integrating them to derive an explicit expression for the particle kinematics on a microscopic scale, make this problem, particularly challenging. Additionally, second-order conformation tensors that characterize the orientation of the polymer macromolecules are typically used in constitutive models to describe the viscoelastic deformation of polymeric fluids [1]. Such a macromolecule can be idealized by elastic dumbbell models [47], equivalence can be drawn to the elastic dumbbell models used to describe the ellipsoid particles in viscoelastic suspensions [44, 45].

As a result, in addition to the rigorous, repetitive calculations required by full-scale simulation over each particle at the microscopic level, particles suspended in viscoelastic media also suffer from additional, hyperbolic evolution equations of the conformation tensor. In light of this, we would like to suggest reduced-order modeling for viscoelastic flows which has received less attention than a problem involving Newtonian fluids [48, 49] for different ranges of fluid’s relaxation times, while streamlining the computations of nonlinear terms, so that such surrogate models can significantly aid future research on the study of DNS of particle kinematics. The fundamental idea of MOR is to tackle the complexity of multi-parametric problems in two stages. In the first offline phase, training data sets are generated through the FOM for given sampling points in the parametric space. The model reduction then is achieved through some dimensionality reduction techniques such as Proper Orthogonal Decomposition (POD) employed by truncated singular value decomposition (SVD) or Principal Component Analysis (PCA) which are used to create a low-rank basis for the solution of the problem [50]. Then, the governing equations are reduced by using POD-Galerkin (POD-G) projection onto the subspace spanned by the reduced basis. Then, during the second online phase, the ROM can be solved for previously unseen parameters

to produce an output of interest. These are generally classified as posteriori methods, On the other hand, a-priori methods like Proper Generalized Decomposition (PGD) [51, 52] assume a canonical tensor decomposition format for the solution and do not need prior availability of high fidelity solutions. It aims to directly construct a reduced model without using the standard solver. In this thesis, we focus on the posteriori methods for performing model order reduction.

While traditional MOR methods are exceptional in accelerating the computational speed-up for linear systems, it is not quite that effective in reducing the computational cost for nonlinear or non-affine parametric dependent systems. The evaluation cost of the projected nonlinear equation on the low dimensional subspace spanned by the reduced basis still depends on the original dimension of the system or in other words the ROM is not fully independent of the true size of the system Therefore, it is necessary to seek a hyper-reduction approach such that the purpose of the model order reduction is served, and the overall significant gain in the computational performance is achieved which is the main goal of MOR. Thus, the first objective of this thesis is met by implementing a stable hyper-reduction framework in the numerical modeling of viscoelastic flows.

As mentioned previously, industrial applications require multi-query predictive simulation for both process and product design, but also for monitoring and control applications. The exploration of the parametric space is a fundamental task in such applications [51]. This leads us to the second MOR method bottleneck: the input parameter for which the snapshots are generated has a significant impact on how accurately ROM replicates the high-fidelity solver. This suggests that in order to effectively represent the large-scale PDE, the training set must be sufficiently rich to fully cover key areas of the parametric space. On the downside, it involves an enormous offline cost. To address this issue, there is already detailed research in the literature on assessing the accuracy of the ROM, thanks to the availability of rigorous error estimators that is the driving process for efficient parametric sampling [10, 53–55]. However, the construction of error estimators is not simple, especially in a complex coupled problem such as viscoelastic fluid flows. Thus, this leads to the second objective of the thesis which is to propose an original training strategy for a ROM by overcoming the requirement of a prior error estimator which is stable, robust, and efficient to overcome build a stable, robust and accurate ROM at a reduced offline cost while having a no prior error estimator.

Another crucial finding in the numerical simulation of particle suspensions is the dynamic evolution of the mesh, where the mesh topology changes in response to the locations where stress is concentrated and moves as a result of continuous particle movement [17, 18]. This implies that each snapshot would be located in a distinct finite element space. The MOR approaches are only relevant to static snapshots produced by a single, identical finite element mesh, which brings us to our third bottleneck. As a result, the thesis' third and final goal is to be able to address the problem of creating a POD-ROM for snapshots that are produced using adaptive finite elements. This is demonstrated in this study using a numerical example of a diffusion-reaction problem that necessitates the use of an adaptive mesh discretization due to the presence of abrupt gradients in the boundary layer at higher values for the reaction coefficients.

The manuscript is organized in the following manner:

1. In chapter 2, a review of some dimensionality reduction methods is presented for a parametrized PDE problem to introduce notations, followed by a discussion on techniques for the reduction of non-linear terms.
2. In chapter 3, a Non-Newtonian viscoelastic fluid system is formulated. The discretization using the finite elements is outlined, and a reduced-order model framework is

derived. Different stabilization strategies are proposed for obtaining a stable and efficient ROM. A hyper reduction technique based on the reduced quadrature rule is examined for the treatment of the nonlinear terms appearing in the governing equations. The chapter is concluded with some numerical benchmark problems resolved with the proposed methodology.

3. In chapter 4, a novel multi-fidelity modeling in conjunction with projection-based ROM is introduced for the reduction of a multi-dimensional parameterized PDE system. The chapter is devoted to adhoc sampling of parameter points from a given discrete training set. The effectiveness of the proposed methodology is demonstrated and validated on a parametric elliptic PDE problem and extended to the viscoelastic fluid flow benchmark problem outlined in chapter 3.
4. Chapter 5 is finally dedicated to model reduction for snapshots that lie on different finite element spaces. The concept from chapter 4 is expanded upon for this issue, and for the first time, a framework is presented that enables one to quickly sample the right parametric points from spatially adjusted snapshots without building any error estimator and easily construct a ROM by simply interpolating from their own mesh into a fixed reference mesh. Additionally, we present a different approach in which we create adaptable reduced basis functions as linear combinations of the chosen adaptive snapshots.



## Chapter 2

# Review of dimensionality reduction techniques

Model order reduction based on projections on low-dimensional spaces (for PDEs) can be traced back to the late 1960s [56] used for studying coherent structures in turbulent flows with the assumption of continuous velocity fields, which was then extended by Sirovich for a discrete database [50]. This has led researchers to apply POD-ROM to a wide range of physical science problems such as for modeling advection-diffusion equations, Navier–Stokes equations for CFD applications, to turbulence modeling which can be referred for further details in the articles [48, 49, 53, 57–59]. Besides physical modeling, POD-ROM has been extensively used for engineering applications such as for shape optimization, and flow control problems [60–62].

Among projection-based MOR methods, the most established are Proper Orthogonal Decomposition (POD) and reduced basis (RB) methods. Other used popular methods are balanced truncation [63], and Krylov subspace methods [64, 65], although it lies out of the scope of the present manuscript. An excellent survey of projection-based reduction techniques can be found in [66]. Before proceeding to the discussions on model order reduction methods, let us first introduce a general idea about the problem definition in the following section. In order to demonstrate the fundamental concept of reduced order modeling, we will take into consideration an elliptic parameterized PDE for the implication that is well discussed in any of the textbooks on model order reduction [10].

### 2.1 General problem definition

Let  $\Omega$  be a bounded Lipschitz domain of dimension  $\mathbb{R}^d$  for  $d = 1, 2, 3$  with Dirichlet ( $\Gamma_D$ ) and Neumann ( $\Gamma_N$ ) boundary conditions enforced such that  $\partial\Omega = \Gamma_D \cup \Gamma_N$  and  $\Gamma_D \cap \Gamma_N = \emptyset$ . We also introduce  $P$ -dimensional bounded parametric domain  $\mathcal{D} \in \mathbb{R}^P$  with  $P \geq 1$  where it can be any set of input parameters be its physical properties of the system, source terms, boundary, or initial conditions, the geometry of the computation domain and so on. Let us introduce the associated Hilbert space,

$$\mathcal{V}_i = \left\{ v \in (H_0^1(\Omega))^d \mid v|_{\Gamma_D^i} = 0 \right\} \quad i = 1, \dots, d$$

Here,  $H_0^1(\Omega) = \left\{ w \in L^2(\Omega) \mid \nabla w \in (L^2(\Omega))^d \right\}$  where  $L^2(\Omega)$  is the space of square-integrable functions over  $\Omega$ . We equip  $\mathcal{V}$  with inner product  $(w, v)_{\mathcal{V}}$  and the induced norm,  $\|w\|_{\mathcal{V}} = \sqrt{(w, w)_{\mathcal{V}}}$ .

The abstract variational formulation in the weak form reads as: for  $\mu \in \mathcal{D}$ , find  $u(\mu) \in \mathcal{V}$  such that,

$$a(u(\mu), v; \mu) = f(v; \mu) \quad \forall v \in \mathcal{V} \quad (2.1)$$

where,  $a : \mathcal{V} \times \mathcal{V} \times \mathcal{D} \rightarrow \mathbb{R}$  is a parameterized bilinear form where the bilinearity is with respect to the first two variables, and  $f : \mathcal{V} \times \mathcal{D} \rightarrow \mathbb{R}$  is a parameterized linear form where the linearity is with respect to the first variable.

To ensure the problem is well-posed, the bilinear form  $a(\cdot, \cdot; \mu)$  is continuous and coercive for every  $\mu \in \mathcal{D}$ :

$$\begin{aligned} \alpha(\mu) &= \inf_{v \in \mathcal{V}} \frac{a(v, v; \mu)}{\|v\|_{\mathcal{V}}^2} > 0 \quad \forall \mu \in \mathcal{D} \\ \gamma(\mu) &= \sup_{u \in \mathcal{V}} \sup_{v \in \mathcal{V}} \frac{a(u, v; \mu)}{\|u\|_{\mathcal{V}} \|v\|_{\mathcal{V}}} < \infty \quad \forall \mu \in \mathcal{D} \end{aligned} \quad (2.2)$$

Under these assumptions and conditions on  $a$  and  $f$ , there exists a unique solution to the weak form due to the Lax-Milgram Theorem [67].

However, the work produced in this manuscript is extended to a more relevant class of problems called saddle point problems, which do not have bilinear forms satisfying the expression as in (2.2). Let  $\mathcal{V} = (H_0^1(\Omega))^d$  and  $\mathcal{Q} = L^2(\Omega)$  be two Hilbert spaces along with their dual  $\mathcal{V}'$ ,  $\mathcal{Q}'$  respectively. The saddle-point problem is to find a pair of fields  $u$  in  $\mathcal{V}$ ,  $p$  in  $\mathcal{Q}$  for  $\mu \in \mathcal{D}$  such that, for all  $v$  in  $\mathcal{V}$  and  $q$  in  $\mathcal{Q}$ ,

$$\begin{aligned} a(u, v; \mu) + b(v, p; \mu) &= f(v; \mu) \quad \forall v \in \mathcal{V} \\ b(u, q; \mu) &= g(q; \mu) \quad \forall q \in \mathcal{Q} \end{aligned} \quad (2.3)$$

where  $a : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$  and  $b : \mathcal{V} \times \mathcal{Q} \rightarrow \mathbb{R}$  are the bilinear forms and  $f \in \mathcal{V}'$  and  $g \in \mathcal{Q}'$  respectively the linear functionals. There exists a unique solution  $u, p$  of the saddle-point problem provided the inf-sup condition on both the bilinear forms  $a$  and  $b$  are fulfilled over  $\mathcal{V}$  and  $\mathcal{Q}$  given by [68]:

$$\begin{aligned} \alpha &= \inf_{u \in \mathcal{V}} \sup_{v \in \mathcal{V}} \frac{a(u, v)}{\|u\|_{\mathcal{V}} \|v\|_{\mathcal{V}}} \geq \alpha_0 > 0 \quad \forall \mu \in \mathcal{D} \\ \beta &= \inf_{q \in \mathcal{Q}} \sup_{v \in \mathcal{V}} \frac{b(v, q)}{\|v\|_{\mathcal{V}} \|q\|_{\mathcal{Q}}} \geq \beta_0 \quad \forall \mu \in \mathcal{D} \end{aligned} \quad (2.4)$$

This framework is used to formulate a number of problems, including Stokes equations, high-order PDEs, and optimal control problems, to name a few. The research done for this manuscript focuses on second-order non-Newtonian fluid flows as mentioned in the introduction, thus we will be dealing with reduced basis approximation of the parameterized Stokes equation with a nonlinear constitutive law that is covered in chapter 3 in great detail.

### 2.1.1 Discretization space

This section discusses the finite element formulation to the abstract form (2.1). Let us define a discrete approximation space  $\mathcal{V}^{\mathcal{N}} \subset \mathcal{V}$  of dimension  $\mathcal{N}$ . The finite element approximation  $u_h \in \mathcal{V}^{\mathcal{N}}$  to the "exact" problem (2.1) can then be expressed as,

$$a(u_h(\mu), v_h; \mu) = f(v_h; \mu) \quad \forall v_h \in \mathcal{V}^{\mathcal{N}} \quad (2.5)$$

Note that,  $\mathcal{V}^{\mathcal{N}}$  inherits the same inner product and norm from  $\mathcal{V}$ . Similar coercivity and continuity conditions as (2.2) are to be fulfilled for well-posedness of the problem in the discrete from,

$$\begin{aligned}
\alpha^{\mathcal{N}}(\mu) &= \inf_{v_h \in \mathcal{V}^{\mathcal{N}}} \frac{a(v_h, v_h; \mu)}{\|v_h\|_{\mathcal{V}^{\mathcal{N}}}^2} \geq \alpha(\mu) \quad \forall \mu \in \mathcal{D} \\
\gamma^{\mathcal{N}}(\mu) &= \sup_{v_h \in \mathcal{V}^{\mathcal{N}}} \sup_{w_h \in \mathcal{V}^{\mathcal{N}}} \frac{|a(v_h, w_h; \mu)|}{\|v_h\|_{\mathcal{V}^{\mathcal{N}}} \|w_h\|_{\mathcal{V}^{\mathcal{N}}}} \geq \gamma(\mu) \quad \forall \mu \in \mathcal{D}.
\end{aligned} \tag{2.6}$$

The FE solution of the field variable ‘ $u$ ’ can be approximated as:

$$u \approx u_h(\mathbf{x}; \mu) = \sum_{i=1}^{\mathcal{N}} N_i(\mathbf{x})(u(\mu))_i \tag{2.7}$$

where,  $N_i$  are the shape functions of choice and  $(u(\mu))_i$  represent the scalar values of the field  $u_h$  at discretization points  $\mathbf{x}_i \in \mathbb{R}^d, i = 1 \dots \mathcal{N}$ . In practice, these values are stored as a collection of high-fidelity solutions that we call "snapshots" at distinct parameter values  $\mu \in \Xi_{train} \subset \mathcal{D}$ , of cardinality  $|\Xi_{train}| = N$ . This set of snapshots  $\{\mathbf{u}^k\}_{k=1}^N$  are generated in the offline stage by solving the PDE equation (2.5) using a high fidelity solver for varying choice of input parameters.

### 2.1.2 Affine parameter dependence

The affine parameter dependence indicates that an operator can be expressed as the sum of the products of parameter-dependent functions and parameter-independent operators. In particular, we say that the parametric bounded bilinear form  $a : \mathcal{V}^{\mathcal{N}} \times \mathcal{V}^{\mathcal{N}} \times \mathcal{D} \rightarrow \mathbb{R}$  and linear form  $f : \mathcal{V}^{\mathcal{N}} \times \mathcal{D} \rightarrow \mathbb{R}$  are affine in the parameter if

$$\begin{aligned}
a(w_h, v_h; \mu) &= \sum_{q=1}^{Q_a} \Theta_a^q(\mu) a^q(w_h, v_h) \\
f(v_h; \mu) &= \sum_{q=1}^{Q_f} \Theta_f^q(\mu) f^q(v_h)
\end{aligned} \tag{2.8}$$

for some finite  $Q_a$  and  $Q_f$ . Here,  $\Theta_a^q, \Theta_f^q : \mathcal{D} \rightarrow \mathbb{R}$  are smooth parameter-dependent functions, and  $a^q(w_h, v_h) : \mathcal{V}^{\mathcal{N}} \times \mathcal{V}^{\mathcal{N}} \rightarrow \mathbb{R}$  and  $f^q(v) : \mathcal{V}^{\mathcal{N}} \rightarrow \mathbb{R}, 1 \leq q \leq Q_a, Q_f$ , are parameter-independent bounded bilinear and linear forms respectively.

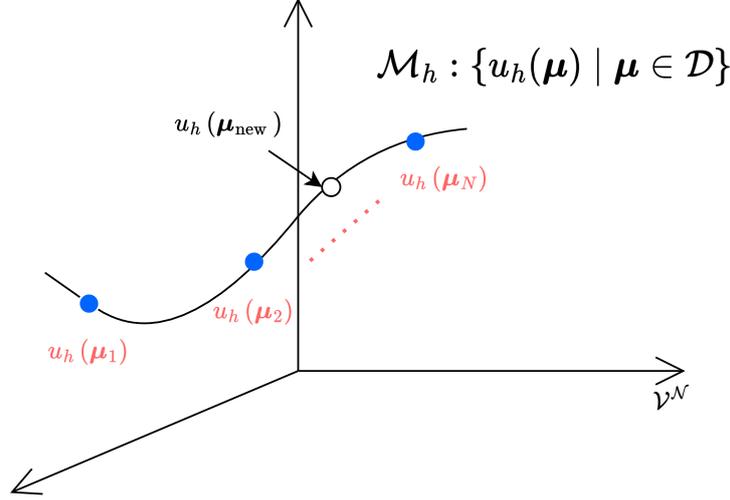
As noted, here  $\mathcal{N}$  represents the number of spatial grid points used for the discretization of the system and it can become enormous for a complex physical system. So solving such a high-fidelity system for any value of  $\mu \in \mathcal{D}$  entails severe computational costs, which can be mitigated by introducing a suitable reduced basis approximation that can represent such high models with similar accuracy. Thus follows our next discussion on model order reduction.

## 2.2 Model order reduction

### 2.2.1 Main features

The reduced basis approximation recognizes that the solution field  $u(\mu)$  (of the corresponding elliptic PDE) theoretically belongs to an infinite-dimensional space  $\mathcal{V}$ , but in fact, resides on a very low-dimensional smooth manifold  $\mathcal{M} \equiv \{u(\mu) \mid \mu \in \mathcal{D}\}$  induced by the parametric dependence. However, the solution manifold  $\mathcal{M}$  is clearly unavailable and we can only compute approximations of a discretized counterpart  $\mathcal{M}_h = \{u_h(\mu) : \mu \in \mathcal{D}\}$ . In the case of a single parametric dimension ( $P = 1$ ), for instance,  $\mathcal{M}_h$  can be described as a one-dimensional filament that winds through  $\mathcal{V}^{\mathcal{N}}$  as illustrated in figure 2.1. All possible

solutions  $u_h(\mu), \forall \mu \in \mathcal{D}$ , lying on this filament, would reside correspondingly on this manifold  $\mathcal{M}_h$  and for any new parameter  $\mu_{\text{new}}$ , the solution  $u_h$  can be approximated as a linear combination of pre-computed solutions. This observation presents a key opportunity where we could obtain a significant dimensionality reduction and therefore save considerable computational efforts.



**Figure 2.1:** Low-dimensional manifold in which the field variable resides and approximation of the solution at  $\mu_{\text{new}}$  by the linear combination of the precomputed solutions.

In reduced order modeling, we aim to exploit a low dimensional structure  $\mathcal{V}^R \subset \mathcal{V}^{\mathcal{N}}$ ,  $R = \dim(\mathcal{V}^R)$  by appropriately choosing a set of orthogonal basis functions  $\{\phi^1, \phi^2, \dots, \phi^R\}$  with  $R \ll \mathcal{N}$  that spans the subspace of the manifold  $\mathcal{M}_h$  and can well represent the manifold with small error. The associated reduced subspace is then given by,

$$\mathcal{V}^R = \text{span}\{\phi^1, \phi^2, \dots, \phi^R\} \subset \mathcal{V}^{\mathcal{N}}$$

The smaller  $R$ , the cheaper the reduced problem to solve. The reduced solution  $u^R \in \mathcal{V}^R$  can then be approximated by a linear combination of these basis functions given by:

$$u^R(\mathbf{x}; \mu) = \sum_{k=1}^R \phi^k(\mathbf{x}) b^k(\mu) \quad (2.9)$$

whose coefficients  $b^k(\mu)$  are calculated thanks to Galerkin projection onto the reduced basis (RB) space. Therefore, the reduced basis approximation for eq. (2.5) can be sought as: for any  $\mu \in \mathcal{D}$ ,

$$a(u_j^R, v; \mu) = f(v; \mu) \quad \forall v \in \mathcal{V}^R \quad (2.10)$$

for  $j = 1 \dots N$ . Recall that,  $R \ll \mathcal{N}$ , so to find the RB solution  $u^R$  we just need to solve an  $R \times R$  linear system which is greatly reduced from solving a  $\mathcal{N} \times \mathcal{N}$  system in the finite element setting. Although, eq. (2.10) is usually a very low dimensional system, it still involves quantities related to the high-fidelity  $\mathcal{N}$ -dimensional space. Using these quantities to assemble the operators for each value of  $\mu$  leaves the cost of the single evaluation input-output  $\mu \rightarrow \mathbf{u}^R(\mu)$  too high. The problem is overcome thanks to the assumption of parameter separability, as discussed in 2.1.2.

Now, the system (2.10) can be expressed as follows,

$$\left( \sum_{q=1}^{Q_a} \Theta_a^q(\mu) \mathbf{A}_R^q \right) \mathbf{u}^R(\mu) = \sum_{q=1}^{Q_f} \Theta_f^q(\mu) \mathbf{f}_R^q, \quad (2.11)$$

where  $(\mathbf{A}_R^q)_{ik} = a^q(\phi^k, \phi^i)$ ,  $(\mathbf{f}_R^q)_i = f^q(\phi^i)$ , for  $1 \leq i, k \leq R$ . Therefore, the computation now requires an expensive offline phase, where all the  $\mu$ -independent structures are computed and stored once for all, and an extremely efficient online phase whose complexity is independent of  $\mathcal{N}$ , to compute for each selected value of  $\mu \in \mathcal{D}$ .

In order to derive a ROM for the saddle point problem as expressed in eq. (2.3), we introduce abstract reduced spaces  $\mathcal{V}^R \subset \mathcal{V}$  and  $\mathcal{Q}^R \subset \mathcal{Q}$  for the field,  $u, p$  respectively. The index  $R$  is related but doesn't necessarily equate to the dimensions of  $\mathcal{V}^R$  and  $\mathcal{Q}^R$ . Thus, by Galerkin projection, find  $u_1^R, u_2^R, \dots, u_N^R \in \mathcal{V}^R$  and  $p_1^R, p_2^R, \dots, p_N^R$  such that,

$$\begin{aligned} a(u_j^R, v; \mu) + b(v, p_j^R; \mu) &= f(v; \mu) & \forall v \in \mathcal{V}^R \\ b(u_j^R, q; \mu) &= g(q; \mu) & \forall q \in \mathcal{Q}^R \end{aligned} \quad (2.12)$$

for all  $j = 1 \dots N$ . The stability of the reduced eq. (2.12) is not guaranteed for all pairs of  $\mathcal{V}^R$  and  $\mathcal{Q}^R$ . It has been extensively discussed in the literature on the construction of ROMs that satisfy the inf-sup condition for problems like the parametrized Stokes equation [48, 59, 69] which is addressed by Galerkin projection of the governing equations onto a POD space for pressure field and a POD space for the velocity field enriched with supremizer functions. On the other hand, the pressure term can be removed while still satisfying the continuity equation by building projection space in the spirit of [70, 71] by ensuring that the resulting velocity POD modes are divergence-free with respect to the pressure reference space.

**Remark:** The Galerkin projection can be very inaccurate for non-coercive or ill-conditioned problems such as convection-diffusion-reaction, wave equations, and heterogeneous problems. In such cases, stability is achieved by choosing a test space different from the approximation space. Various approaches for selecting suitable test spaces have been proposed in the context of reduced basis approximation methods [72–75]. One of the simplest ways is to use minimal residual methods, where the test space is chosen such that the dual norm of the residual error  $\|r(u^R(\mu); \mu)\|_{\mathcal{V}^{\mathcal{N}'}}$  is minimized. More precisely, the minres projection is a Petrov-Galerkin projection defined by,

$$\langle r(u^R(\mu); \mu), v \rangle = 0, \quad \forall v \in \mathcal{W}^R \quad (2.13)$$

The choosing of test space  $\mathcal{W}^R$  can be appropriately referred to in details in [10]. Here  $r(\cdot; \mu) \in \mathcal{V}^{\mathcal{N}'}$  is the residual of the high-fidelity problem computed on the RB solution given by,

$$r(v; \mu) = f(v; \mu) - a(u^R(\mu), v; \mu) \quad \forall v \in \mathcal{W}^R.$$

Note that

$$\langle r(\cdot; \mu), v \rangle = r(v; \mu)$$

The following section is dedicated to the construction of reduced basis functions.

### 2.2.2 Reduced basis generation via classical POD

The proper orthogonal decomposition, also known as Karhunen–Loève expansion and principle components analysis computes a set of basis vectors that capture the dominant structures of the system [50, 56]. Given a set of snapshots  $\mathbf{S} = [\mathbf{u}^1, \mathbf{u}^2 \dots, \mathbf{u}^N] \in \mathbb{R}^{N \times N}$ , a POD basis of dimension ‘ $r$ ’ with  $1 \leq r \leq d = \min(\mathcal{N}, N)$  is a set of orthonormal vectors  $\{\phi^j(x)\}_{j=1}^r \subset \mathbb{R}^{\mathcal{N}}$  whose linear span best approximates the concerned manifold of solutions. The basis set  $\{\phi^j(x)\}_{j=1}^r$  is obtained in such a way that it minimizes the following problem in  $\mathcal{L}^2$ -norm

$$\min_{\{\phi^j\}_{j=1}^r} \sum_{k=1}^N \left\| \mathbf{u}^k - \sum_{j=1}^r \langle \mathbf{u}^k, \phi^j \rangle \phi^j \right\|_{\mathcal{L}^2}^2 \quad (2.14)$$

$$\text{with } \phi_i^T \phi_j = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}, \quad i, j = 1, \dots, r \quad (2.15)$$

Here,  $\langle \cdot, \cdot \rangle$  represents the inner product in  $\mathcal{L}^2$ -norm. It is well known that the solution to eq. (2.14) can be obtained by SVD. The basis vector is provided by the set of the left singular vectors of the snapshot matrix  $\mathbf{S}$ . In particular, suppose that the SVD of  $\mathbf{S}$  is

$$\mathbf{S} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \quad (2.16)$$

where  $\mathbf{U} = [\phi^1, \dots, \phi^r] \in \mathbb{R}^{N \times r}$  and  $\mathbf{V} = [\psi_1, \dots, \psi_r] \in \mathbb{R}^{N \times r}$  are orthonormal left, and right singular vectors of rank  $r$ , and  $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$ .

It is possible to define ‘ $r$ ’ such that the loss of relative information content of the parametric system corresponding to the unselected ‘ $d - r$ ’ modes is less than a certain tolerance  $\delta_g$  using the following relation:

$$I(r) = \frac{\sum_{i=r+1}^d \sigma_i^2}{\sum_{i=1}^d \sigma_i^2} \leq 1 - \delta_g \quad (2.17)$$

**Remark:** It is essential to have a decay of singular values for an efficient POD. If the decay of the singular values is very slow which is typical for high-rank PDEs, either the ROM constructed will be of large dimension or it will contain large approximation errors. For practical reasons, if  $N \ll \mathcal{N}$ , the method of snapshots [50] can be used as a way to efficiently determine the POD modes for large problems, such as those encountered in CFD applications. Instead of computing SVD on the snapshot matrix, we compute eigenvector of the covariance matrix  $\mathbf{S}^T \mathbf{S} \in \mathbb{R}^{N \times N}$  reduced significantly from order  $\mathcal{O}(\mathcal{N} \times N)$ . POD basis is then calculated by post-multiplying the snapshot matrix with these eigenvectors. An eigenvalue problem can be written mathematically as,

$$\mathbf{S}^T \mathbf{S} \psi_j = \lambda_j \psi_j, \quad \text{for } j = 1, \dots, r \quad (2.18)$$

where  $\mathbf{S}^T \mathbf{S}$  is known as the covariance matrix with size  $N \times N$ . Note from eq.(2.16),

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

Comparing (2.18) with (2.16), we see that the right singular vectors of  $\mathbf{S}$  are the same as the eigenvectors of  $\mathbf{S}^T \mathbf{S}$ , and the singular values of  $\mathbf{S}$  are the positive square roots of

the eigenvalues of  $\mathbf{S}^\top \mathbf{S}$ . Now, the  $j^{\text{th}}$  POD basis function  $\phi^j$  is determined by,

$$\phi^j = \frac{1}{\sqrt{\lambda_j}} \sum_{k=1}^N \mathbf{S}_{(:,k)} \psi_k^j, \quad 1 \leq j \leq r,$$

The method of snapshots has been effectively used in both continuous and discrete settings [76, 77] for cases when snapshots lie in different discretization spaces. Some other state-of-the-art POD methods are discussed below which are useful in many applications such as pattern recognition [78], image processing, face reconstruction [79], to deriving reduced order models.

### 2.2.3 Modified POD/SVD techniques

- Randomized algorithms:

Randomness is a powerful concept that has been a hot topic in the scientific community over the past few decades due to its surprising reliability and computational efficiency [80]. In recent times, probabilistic methods are used as an effective strategy for designing better algorithms so that the bottleneck of deterministic algorithms can be overcome. It has been effectively used in matrix approximations. The Monte Carlo algorithm is one of the most well-known and widely used randomized methods [81]. Generally, probabilistic matrix algorithms work by introducing a degree of randomness to derive smaller matrices from a large-scale dataset while retaining the essential features of the matrix. It is crucial to understand randomized matrix algorithms are most effective for low-rank matrices.

For instance, in randomized SVD (RSVD) we can compute singular vectors from a smaller-sized matrix  $\mathbf{B} \in \mathbb{R}^{k \times N}$ ,  $k \ll N$  instead of the full-sized snapshot matrix  $\mathbf{S}$ . The smaller matrix  $\mathbf{B}$  is constructed by projecting the full snapshot matrix  $\mathbf{S}$  onto a basis matrix computed using a randomized scheme  $\mathbf{Q} \in \mathbb{R}^{N \times k}$ , i.e.  $\mathbf{B} = \mathbf{Q}^\top \mathbf{S}$ . Now full SVD is computed on  $\mathbf{B}$  matrix rather than  $\mathbf{S}$ ,

$$\mathbf{B} = \tilde{\mathbf{U}} \mathbf{\Sigma} \mathbf{V}^\top \quad (2.19)$$

The right singular vectors  $\mathbf{V} \in \mathbb{R}^{k \times N}$  as well as the corresponding singular values are recovered. The true left singular vectors  $\mathbf{U} \in \mathbb{R}^{N \times k}$  can be recovered from the approximate left singular vectors,  $\tilde{\mathbf{U}} \in \mathbb{R}^{k \times k}$  as,

$$\mathbf{S} \approx \mathbf{Q} \mathbf{Q}^\top \mathbf{S} = \mathbf{Q} \mathbf{B} = \mathbf{Q} \tilde{\mathbf{U}} \mathbf{\Sigma} \mathbf{V}^\top \quad (2.20)$$

where,

$$\mathbf{U} \approx \mathbf{Q} \tilde{\mathbf{U}} \quad (2.21)$$

Detailed work on randomized SVD can be found in [82, 83]. The computational complexity of this approximation in time is  $\mathcal{O}(N N r)$  (assuming dense matrix) in contrast to  $\mathcal{O}(N^2)$  for full SVD. Wolfe et al. [84] provided a computationally more effective alternative for structured random test matrices, bringing down the costs to  $N \log(r)$ . In the PhD thesis of Balabanov [85], a randomized sketching technique was developed for random projections of high dimensional data into a low dimensional space initially, and after the reduced model is then efficiently constructed which is also numerically stable from the projections of the reduced approximation space and the spaces of associated residuals rather than the original high dimensional space.

- **Incremental POD/SVD:** Different methods on incremental SVD have been proposed for “long-thin” matrices in [86, 87] which is a good assumption for many MOR problems. These algorithms incrementally compute the SVD of a matrix when new rows (columns) are added to the matrix. In the article [86], the authors implemented a surrogate POD model by a rank-1 update to a 2D airfoil simulation, as proposed by Brand [88]. The method follows as: when a new snapshot is available  $\mathbf{u}^k$ , and POD basis of the updated snapshot matrix  $[\mathbf{S}_k, \mathbf{u}^{k+1}] \in \mathbb{R}^{N \times (k+1)}$  has to be computed that can be obtained by rank-1 update of the SVD,

$$\mathbf{S}_{k+1} = [\mathbf{S}_k, \mathbf{u}^{k+1}] = \left[ \mathbf{U}_k \frac{\phi}{\|\phi\|} \right] \mathbf{D}_{k+1} \begin{bmatrix} \mathbf{V}_k^T & \mathbf{0} \\ \mathbf{0}^T & \mathbf{1} \end{bmatrix} \quad (2.22)$$

where,

$$\mathbf{D}_{k+1} \stackrel{\text{def}}{=} \begin{bmatrix} \Sigma_k & \mathbf{U}_k^T \mathbf{u}^{k+1} \\ \mathbf{0}^T & \|\phi\| \end{bmatrix}$$

where  $\phi$  is the basis obtained by Gram Schmidt orthogonalization of the new snapshot  $\mathbf{u}^{k+1}$  against the columns of  $\mathbf{S}_k$ . By SVD,  $\mathbf{D}_{k+1} = \tilde{\mathbf{U}} \Sigma_{k+1} \tilde{\mathbf{V}}^T$ , we can rewrite (2.22),

$$\mathbf{S}_{k+1} = \left[ \mathbf{U}_k \frac{\mathbf{q}}{\|\mathbf{q}\|} \right] \tilde{\mathbf{U}} \Sigma_{k+1} \tilde{\mathbf{V}}^T \begin{bmatrix} \mathbf{V}_k^T & \mathbf{0} \\ \mathbf{0}^T & \mathbf{1} \end{bmatrix} = \mathbf{U}_{k+1} \Sigma_{k+1} \mathbf{V}_{k+1}^T$$

In [89], the authors proposed low-rank updates to the SVD of dense matrices. One can read the work of [90], where the authors have implemented online adaptivity to the ROM by low-rank updates to both the POD basis and the reduced operators, while avoiding expensive rebuilding of the FOM operators online. In [86], the authors combined incremental SVD with a self-adaptive (adaptation in the parametric space) strategy to estimate and increase the accuracy of the POD/SVD surrogate model and successfully implemented the procedure in a 2D airfoil simulation. This strategy is based on the leave-one-out algorithm (LOOA), which is extensively used in machine learning algorithms for cross-validation when data samples are limited. Detecting the regions with large errors, the parametric domain is resampled using quad-tree data structures.

- **Gappy POD:** Gappy POD is an extension to classical POD that allows the consideration of incomplete data sets. It uses a POD basis to reconstruct the missing or “gappy” data. Gappy POD was developed by Everson and Sirovich in the context of reconstruction of images, such as human faces, from partial data [91] and has been extended to flow problems where only partial experimental data might be available [92–94]. The method follows: suppose we are given the first ‘ $r$ ’ POD basis  $\phi_r$  of the snapshot matrix  $\mathbf{S}$  where all snapshots are known, and let  $\mathbf{g}(\mu)$  be another solution vector whose some spatial elements are missing. Then, by gappy-POD approximation, we can reconstruct the full vector  $\tilde{\mathbf{g}}(\mu)$  from the incomplete data using the existing POD basis functions as,

$$\tilde{\mathbf{g}} = \sum_{i=1}^r \phi_i b_i \quad (2.23)$$

The coefficients  $b_i$  are computed by minimizing the error between the original missing data and repaired solution vector,

$$\min_b \|g - \tilde{g}\|_n^2 \quad (2.24)$$

using the gappy norm so that only the elements present in the gappy data are compared.

**Definition:**  $\|\mathbf{v}^k\|_n$  is called the gappy norm of  $\mathbf{v}^k$  which is defined as the following,

$$\|\mathbf{v}^k\|_n^2 := (\mathbf{v}^k, \mathbf{v}^k)_n = (\mathbf{n}^k \odot \mathbf{v}^k, \mathbf{n}^k \odot \mathbf{v}^k)_{\mathcal{L}^2} = \|\mathbf{n}^k \odot \mathbf{v}^k\|_{\mathcal{L}^2}^2 \quad (2.25)$$

where ‘ $\odot$ ’ denotes a point-wise multiplication and  $\mathbf{n}^k \in \mathbb{R}^N$  is a mask vector that describes where the data are missing and available on the particular observation vector  $\mathbf{v}_j$ . Hence,  $\mathbf{n}_i$  for an observation, the data set  $\mathbf{v}_i$  is defined in the following way:

$$n_i^k = \begin{cases} 1, & \text{if } v_i^k \text{ is known} \\ 0, & \text{otherwise} \end{cases}$$

- Partitioned method of snapshots: If ‘ $p$ ’ multiple processors are used for generating high-fidelity snapshots, the partitioned method of snapshots is a suitable choice for computing basis functions. The idea of partitioned SVD was developed by Beattie et al. [95] and further developments on the domain decomposition approach to SVD can be found in the article [96]. The method follows: assume  $\mathbf{S}_i \in \mathbb{R}^{n_i \times N}$  is the local data on each processor, with ‘ $n$ ’ being the local degrees of freedom,  $N$  the total number of snapshots and  $p$  is the number of processors. Due to the parallelism of the simulation, one can first compute  $\mathbf{D}_i = \mathbf{S}_i^T \mathbf{S}_i$  locally on each processor, then transfer  $\mathbf{D}_i$  to a single processor which can be obtained by simple addition,  $\mathbf{D} = \sum_{i=1}^p \mathbf{D}_i$  followed by solution of the eigenvalue problem  $[\mathbf{V}, \mathbf{\Sigma}] = \text{eigs}(\mathbf{D})$  where `eigs` is a MATLAB function. Then the components of POD basis functions can be computed by vector multiplication of  $\mathbf{V}$  and  $\mathbf{S}_i$  locally on each processor,

$$\phi_i^j = \frac{1}{\sqrt{\lambda_j}} \sum_{k=1}^N \mathbf{S}_{i(\cdot, k)} \psi_k^j, \quad 1 \leq j \leq r, 1 \leq i \leq p.$$

## 2.2.4 Greedy RBM

The objective of the greedy algorithm in the context of the reduced basis method is to adaptively enrich the reduced subspace with linearly independent basis functions [97]. By doing so, the evaluation of high-fidelity snapshots for all the training parameters (as done in the classical POD) can be avoided in the offline step, thereby reducing the offline cost and improving the efficiency of MOR. It is based on the idea to select the parameter representing a local optimum in relation to an opportune error indicator iteratively,

$$\mu_{n+1} = \arg \max_{\mu \in \Xi_{train}} \Delta^R(\mu) \quad (2.26)$$

which means in the  $(n+1)^{th}$  step, basically the sample point whose error metric  $\Delta^R(\mu)$  indicates to be worst approximated among all the parameters  $\mu \in \Xi_{train}$  by the solution of the current reduced model  $\mathcal{V}_{n+1}^R$  is selected as the next sample point. At the sampled point, the high-fidelity snapshot is generated using the finite element model, followed by an enrichment of the reduced basis. This is repeated until the error estimator reaches a

prescribed tolerance. The algorithm is given below,

**Input:**  $\delta_g, \mu^1, N_{\max}$   
**Output:** A reduced space  $\mathcal{V}^R$   
 $n = 1$   
 $S_n = \{\mu^1\}$   
Evaluate  $u_h(\mu^1)$ ;  
 $\mathcal{V}_n^R = \text{span}\{u_h(\mu^1)\}$   
compute  $\Delta_n^R(\mu^1)$ ;  
**while**  $\Delta^R(\mu) > \delta_g$  and  $n \leq N_{\max}$  **do**  
 $n \leftarrow n + 1$   
compute  $\Delta_n^R(\mu) \quad \forall \mu \in \Xi_{\text{train}}$   
 $\mu^n := \text{argmax}_{\mu \in \Xi_{\text{train}}} \Delta_{n-1}^R(\mu)$   
 $S_n \leftarrow S_{n-1} \cup \{\mu^n\}$   
compute  $u_h(\mu^n)$   
 $\mathcal{V}_n^R \leftarrow \mathcal{V}_{n-1}^R \cup \text{span}\{u_h(\mu^n)\}$   
**end**

It is essential for a good a posteriori error estimator to be sharp, and rigorous for ensuring the reliability of the RBM. The ideal error indicator would be,  $\Delta_{n-1}^R = \|u_h(\mu) - \Pi u_h(\mu)\|_{\mathcal{V}^{\mathcal{N}}}$  where  $\Pi u_h(\mu)$  is the orthogonal projection of  $u_h(\mu)$  onto  $\mathcal{V}_{n-1}^R$ . However, such an error indicator is expensive to compute due to the evaluation of FOM solve for each parameter in the training set in every step of the algorithm. So, to obtain an efficient performance of the greedy sampling instead a surrogate error indicator is used which is given by dual norm of the residual such that,

$$\|u_h(\mu) - u^R(\mu)\|_{\mathcal{V}^{\mathcal{N}}} \leq \Delta^R(\mu) = \frac{\|r(u^R; \mu)\|_{\mathcal{V}^{\mathcal{N}'}}}{\alpha_{LB}^{\mathcal{N}}} \quad (2.27)$$

where  $r(u^R(\mu); \mu) : \mathcal{V}^{\mathcal{N}} \rightarrow \mathbb{R}$  is the dual norm of the residual w.r.t the reduced solution given by,

$$\langle r(u^R(\mu); \mu), v_h \rangle_{\mathcal{V}^{\mathcal{N}'}} = f(v_h; \mu) - a(u^R, v_h; \mu), \quad \forall v_h \in \mathcal{V}^{\mathcal{N}} \quad (2.28)$$

and  $\alpha_{LB}^{\mathcal{N}} : \mathcal{D} \rightarrow \mathbb{R}$  is the lower bound for the coercivity constant  $\alpha^{\mathcal{N}}$  such that:

$$0 < \alpha_{LB}^{\mathcal{N}}(\mu) \leq \alpha^{\mathcal{N}}(\mu) \quad \forall \mu \in \mathcal{D}$$

The computational cost to evaluate  $\mu \rightarrow \alpha_{LB}^{\mathcal{N}}(\mu)$  is independent of  $\mathcal{N}$ . To get this lower bound, we can resort to theta methods for simple PDEs ([9]) explained in the appendix A or by successive constraint method (SCM) for general PDEs [98–100].

## 2.3 Reduction of non-linear terms

The online efficiency of the classical projection-based MOR methods usually depends on parameter separability, as shown in eq.(2.8). So far, we considered the parametric dependence of the operator  $a(w, v; \mu)$  is affine in relationship to  $\mu$  and linear w.r.t  $u(\mathbf{x}; \mu)$

which is indeed a very restrictive hypothesis. Typically, all operators do not admit such a trivial affine decomposition (2.8) and without this assumption, we lose the possibility of offline-online decoupling and our problem still depends on the original dimension of the system.

To fix this problem, the most popular techniques involve the use of a hyper reduction stage, first coined by Ryckelynck [101] and can be broadly classified into interpolation approaches such as (discrete) empirical interpolation method (EIM) [55, 102], the best point interpolation method (BPIM) [103] and integral approaches based on empirical cubature rule [104, 105]. Some other commonly used methods are the missing point estimation method [106], Gauss-Newton Approximated Tensors (GNAT) [107], Petrov–Galerkin reduced integration [101]. The idea is to find a reduced integration domain (RID) for doing fast numerical integration of the nonlinear terms in the online stage. In this chapter, we will be focusing only on DEIM and empirical quadrature rule by sparse approximation.

### 2.3.1 Interpolation based approach

Let  $F(u_h(\mathbf{x}; \mu); \mu) \in \mathbb{R}^N$  denote a nonlinear operator that is nonlinear in the field variable and does not admit an affine structure as in (2.8). The corresponding projection onto a reduced order space is given by  $F^R(\mathbf{x}; \mu) = \phi^\top F(u_h(\mathbf{x}; \mu); \mu) \in \mathbb{R}^r$ . In DEIM, the idea is to approximate the non-linear function  $F(u_h(\mathbf{x}; \mu); \mu)$  using an interpolative approach in order to construct an affine decomposition for an online efficient ROM,

$$F(\mathbf{x}; \mu) \approx \hat{F}(\mathbf{x}; \mu) = \sum_{j=1}^{r'} q^j(\mathbf{x}) m_j(\mu) \quad (2.29)$$

where  $q^j(\mathbf{x}) \in \mathbb{R}^{N \times r'}$  are the basis functions of dimension ‘ $r'$ ’ such that  $r' \leq N$  and  $m_j$  denotes the interpolative coefficients. The basis functions are obtained by performing POD on the set of snapshots of the nonlinear term evaluated in the offline stage  $\{F^1, F^2 \dots F^N\} \in \mathbb{R}^N$ . The interpolation coefficients are determined in such a way that the approximation is exact at certain points  $\mathbf{x}_p, p = 1 \dots r'$

$$F(\mathbf{x}_p; \mu) = \hat{F}^{DEIM}(\mathbf{x}_p; \mu) = \sum_{j=1}^{r'} q^j(\mathbf{x}_p) m_j(\mu) \quad \forall p \quad (2.30)$$

The above equation is now a  $r' \times r'$  system and is solved to determine the interpolation coefficients. The above equation can be re-written in algebraic form as:

$$\hat{F}^{DEIM} = \mathbb{Q}m \quad (2.31)$$

where,

$$[\hat{F}^{DEIM}(\mu)]_p = \hat{F}^{DEIM}(\mathbf{x}_p; \mu) \quad (\mathbb{Q}_p)_j = q^j(\mathbf{x}_p) \quad [m(\mu)]_p = m_p(\mu)$$

Now eq. (2.29) can be written as,

$$F(\mathbf{x}; \mu) \approx \hat{F}(\mathbf{x}; \mu) = \sum_{j=1}^{r'} q^j(\mathbf{x}) [\mathbb{Q}^{-1} \hat{F}^{DEIM}(\mu)]_j \quad (2.32)$$

The interpolation points can be determined through the DEIM algorithm using the greedy approach as explained in detail in [108]. Since the input basis functions are linearly independent and arranged according to the dominant singular values, the DEIM algorithm ensures the indices of the interpolation points selected are non-repetitive and hierarchical

in nature. Once the interpolation points are found, the interpolative basis functions are determined as the inverse of the operator  $\mathbb{Q}$ , whose rows correspond to the evaluation of the basis functions,  $\{q^j\}_{j=1}^{r'}$  at points  $\mathbf{x}_p$ . The user can choose to apply such an algorithm either on the nodes of the grid or on the set of quadrature points.

The DEIM algorithm selects the indices of the interpolation points from the basis functions sequentially by finding the max norm of the residual of the current approximation.

---

**Algorithm 1** DEIM algorithm

---

- 1: Input: POD basis of the non-linear terms  $\{q^i\}_{i=1}^{r'} \in \mathbb{R}^{\mathcal{N}}$
  - 2: Output: DEIM points,  $\mathbf{p}$
  - 3:  $p_1 \leftarrow \arg \underbrace{\max}_{j=1 \dots \mathcal{N}} |q^1(x_j)|$
  - 4:  $\mathbf{Q} = [q^1], \mathbf{p} = [p_1]$  and  $\mathbb{Q} = \mathbf{Q}_{(x_{p_1}, \cdot)}$
  - 5: **for**  $l = 2 : r'$  **do**
  - 6:     Solve  $\mathbb{Q}\mathbf{m} = q^l(x_{\mathbf{p}})$
  - 7:      $\mathbf{r} \leftarrow q^l - \mathbb{Q}\mathbf{m}$
  - 8:      $p_l = \arg \underbrace{\max}_{j=1 \dots \mathcal{N}} |\mathbf{r}(x_j)|$
  - 9:      $\mathbf{Q} = [\mathbf{Q}, q^l], \mathbf{p} = [\mathbf{p}, p_l]$  and  $\mathbb{Q} = \mathbf{Q}_{(x_{\mathbf{p}}, \cdot)}$
  - 10: **end for**
- 

### 2.3.2 Integral approximation approach

In the context of numerical integration by quadrature rule, the essence of this approach is to apply a low-cost quadrature rule during the online phase, instead of computing an affine decomposition in the offline phase [109]. In the reduced order model,  $F^R(\mathbf{x}; \mu)$  can also be obtained as a result of integrating over the concerned domain,

$$F^R(\mathbf{x}; \mu) = \Phi^\top \int_{\Omega} f_h d\Omega = \int_{\Omega} f d\Omega$$

As done in a standard finite element model, we can now express the approximation of the nonlinear term using the standard Gaussian quadrature rule as,

$$\int_{\Omega} f(\mathbf{x}; \mu) d\Omega \approx \sum_{i=1}^{\mathcal{N}_q} f(\mathbf{x}_i; \mu) \omega_i \quad (2.33)$$

with additional constraints,

$$\begin{aligned} \sum_{i=1}^{\mathcal{N}_q} \omega_i &= |\Omega| \\ \omega_i &\geq 0 \end{aligned} \quad (2.34)$$

where  $\mathcal{N}_q$  is the number of quadrature points of the finite element domain and  $\omega_i$  are its corresponding weights. The idea is to approximate the integral at an optimal set of points ‘ $p$ ’ from among the original set of integration points  $p \subset \mathcal{N}_q$  using the data from high fidelity simulations as,

$$\begin{aligned} \int_{\Omega} f_{\text{HR}}^j d\Omega &= \sum_{k=1}^p \omega_k^{\text{HR}} f^j(\mathbf{x}_k) \quad j = 1 \dots N, p \ll \mathcal{N}_q \\ \text{s.t. } \int_{\Omega} f^j(\mathbf{x}; \mu) d\Omega &\approx \int_{\Omega} f_{\text{HR}}^j d\Omega \end{aligned} \quad (2.35)$$

But  $p$  is not known a priori, so instead a sparse representation of ‘ $\omega_k^{\text{HR}}$ ’ is to be found such that we approximate,

$$\int f_{\text{HR}}^j d\Omega \approx \sum_{k=1}^{\mathcal{N}_q} \omega_k^{\text{HR}} f^j(\mathbf{x}_k) \quad j = 1 \dots N, \quad (2.36)$$

This leads to the following optimization problem,

$$\min_{\mathbf{y} \in \mathbb{R}^{\mathcal{N}_q}} \|\mathbf{y}\|_{\ell_p} \quad (2.37a)$$

$$\text{subject to } \|\mathbf{A}\mathbf{y} - \mathbf{b}\|_{\ell_2} \leq \epsilon \quad (2.37b)$$

$$\sum_{k=1}^{\mathcal{N}_q} y_k = |\Omega|, y_k \geq 0 \quad \forall k \quad (2.37c)$$

where  $\|\cdot\|_p$  is the classical  $\ell_p$ -norm associated with vectors and the value  $p$  depends on the algorithm chosen for sparse approximation.  $\mathbf{A}$  and  $\mathbf{b}$  are given by,

$$\mathbf{A} := \begin{bmatrix} f^1(\mathbf{x}_1) & f^1(\mathbf{x}_2) & \dots & f^1(\mathbf{x}_{\mathcal{N}_q}) \\ f^2(\mathbf{x}_1) & f^2(\mathbf{x}_2) & \dots & f^2(\mathbf{x}_{\mathcal{N}_q}) \\ \dots & \dots & \vdots & \dots \\ f^N(\mathbf{x}_1) & f^N(\mathbf{x}_2) & \dots & f^N(\mathbf{x}_{\mathcal{N}_q}) \end{bmatrix}, \quad \mathbf{b} := \begin{bmatrix} \int_{\Omega} f^1 d\Omega \\ \int_{\Omega} f^2 d\Omega \\ \vdots \\ \int_{\Omega} f^N d\Omega \end{bmatrix}$$

Notice that the  $k^{\text{th}}$  column of  $\mathbf{A} \in \mathbb{R}^{N \times \mathcal{N}_q}$  contains the value of the integrand at the  $k^{\text{th}}$  integration point for all the parameters in the training set, the vector  $\mathbf{b} \in \mathbb{R}^N$ , on the other hand, is formed by arranging the exact integral of each function in a single column and  $\mathbf{y} \in \mathbb{R}^{\mathcal{N}_q}$  contains the modified integration weights. This is an under-determined system since  $N \ll \mathcal{N}_q$ .

**Proposition:** One particular solution of the undetermined system (2.37), assuming  $\mathbf{A}$  full rank so that  $\mathbf{A}\mathbf{A}^T$  is invertible, is

$$\mathbf{y} = \mathbf{A}^\dagger \mathbf{b},$$

where  $\mathbf{A}^\dagger = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1}$  denotes the Moore-Penrose inverse [110]. This solution is the one that minimizes  $\|\mathbf{y}\|_{\ell_2}$ , i.e.  $\mathbf{y}$  is the solution of the optimization problem (2.37)

$$\begin{aligned} & \min_{\mathbf{y} \in \mathbb{R}^{\mathcal{N}_q}} \|\mathbf{y}\|_{\ell_2} \\ & \text{subject to } \|\mathbf{A}\mathbf{y} - \mathbf{b}\|_{\ell_2} \leq \epsilon \end{aligned} \quad (2.38)$$

Unfortunately, the minimum norm solution does not provide a sparse solution. Rather, it has the tendency to spread the energy among all entries of,  $\mathbf{y}$  instead of focussing all the energy into just a few entries. If the sparse solutions are obtained using  $\ell_0$  norm, then one needs to find the number of non-zero components of  $\mathbf{y}$  but solving in  $\ell_0$  norm is an NP-hard problem and needs appropriate formulations [111]. A common strategy in sparse regression is to replace  $\ell_0$  by  $\ell_1$  norm [112], also known as Lasso regression which naturally provides a sparse solution, however, this choice is not consistent in this scenario as it conflicts with the constraints (2.37c) expressed in the optimization problem. Since the sum of the reduced quadrature weights,  $\sum_k y_k$  is exactly equivalent to the  $\ell_1$ - norm of  $y$  (as the quadrature weights are non-negative), and therefore cannot be minimized.

To address these issues while respecting all the constraints, Manucci et al. in the article [113] have developed a modified version of the FOCUSS algorithm originally

introduced for signal processing [114] into an efficient cubature scheme that can exactly retain the Gauss Legendre Quadrature rule. The idea is to search for a solution that minimizes the  $\ell_p$  norm with  $0 < p < 1$ . There are also different concepts proposed in sparse approximation and one can read [104] where sparsity is obtained by a heuristic sequential point selection process but at the cost of solving a non-negative least squares problem. In order to treat the nonlinear terms appearing in the governing equations of viscoelastic fluid flows in this manuscript, the hyper reduction method is adapted from the work of [113].

### 2.3.3 Reduced quadrature rule based on sparse recovery technique: non-negative FOCUSS algorithm

The problem is solved using a non-negative FOCUSS algorithm [113]. It is an iterative fixed-point algorithm, that achieves the  $\ell_p$  quasi norm minimum by solving a sequence of weighted  $\ell_2$  norm optimization problems, i.e.  $\|\mathbf{W}^{-1}\mathbf{y}\|_{\ell_p}$  where  $\mathbf{W}$  is a square invertible matrix with  $0 < p < 1$  for which we have a unique solution. At iteration  $n + 1$ , the solution reads as,

$$\mathbf{y}^{n+1} = \mathbf{W}^n (\mathbf{A}\mathbf{W}^n)^\dagger \mathbf{b} \quad (2.39)$$

where  $\mathbf{W}^n = \text{diag}(|\mathbf{y}^n|^{1-\frac{p}{2}})$ ,  $\text{diag}$  is a diagonalization operator in MATLAB and  $(\mathbf{A}\mathbf{W}^n)^\dagger$  denotes the Moore-Penrose inverse.

The authors extrapolate the idea of the FOCUSS algorithm into an efficient quadrature scheme that can exactly retain the Gauss Legendre quadrature rule. To adapt into an efficient quadrature rule, it needs to ensure the non-negativity of the integration weights. This can be obtained by introducing a relaxation step to check at every iteration if each entry is positive:

$$\mathbf{y}^{n+1} \leftarrow \alpha_n \mathbf{y}^{n+1} + (1 - \alpha_n) \mathbf{y}^n \quad \text{if } \mathbf{y}^{n+1} \not\geq \mathbf{0} \quad (2.40)$$

The algorithm begins with an initial guess  $\mathbf{y}^0$  with all positive entries, for instance, the full order quadrature rule, and then at every iteration onwards ( $n + 1$ ), first, it checks if each entry of the weights  $\mathbf{y}^{n+1}$  are non-negative. If not, the new solution of step ( $n + 1$ ) is determined according to the condition expressed in (2.40).

Besides, the condition on the measure of the domain  $\sum_{k=1}^{N_q} y_k = |\Omega|$  is enforced by adding to the last row of the matrix  $\mathbf{A}$ , a uniform vector of one corresponding to a constant function and the last entry of  $\mathbf{b}$  corresponds to  $|\Omega|$ . This step ensures the well-posedness of the minimization problem, otherwise ill-posed for cases in which the integral function is zero for all values of the input parameters, i.e. which possess trivial solution [104]. As we seek for the sparsest solution, the algorithm is considered to reach convergence when  $\|\mathbf{y}_{k+1} - \mathbf{y}_k\|_{\ell_2} / \|\mathbf{y}_k\|_{\ell_2} < \epsilon$ , where  $\epsilon$  is a user-defined tolerance.

By introducing a regularization parameter  $\gamma$ , sparsity can be tuned as per the user's requirements while always controlling the residual at each step. Finally, the objective function can be replaced with:

$$\min_{\mathbf{y} \in \mathbb{R}^{N_q}} \left[ \|\mathbf{A}\mathbf{y} - \mathbf{b}\|_{\ell_2} + \gamma \|\mathbf{W}^{-1}\mathbf{y}\|_{\ell_2} \right] \quad (2.41)$$

The solution of this minimization problem is given by:

$$\mathbf{W}^{-1}\mathbf{y}^{k+1} = \mathbf{W}^k \mathbf{A}^T \left[ \mathbf{A}\mathbf{W}^k (\mathbf{A}\mathbf{W}^k)^T + \gamma \mathbf{I} \right]^{-1} \mathbf{b} \quad (2.42)$$

By setting to a higher value of  $\gamma$ , we tend to obtain the solution of the minimization problem with more zero entries in the set. For details on the algorithm, please refer to [113].



# Chapter 3

## Reduced order modeling framework for viscoelastic fluid flows

### Abstract

This chapter is divided into two parts: Part I is a brief review on the rheology of fluids with a focus on viscoelastic fluids referred from the books [1, 115] and, Part II is dedicated to a computational framework for reduced order modeling of viscoelastic fluid flows, particularly focused on the stabilization method and the approximation of nonlinear terms appearing in the governing equations. Three different stabilization approaches are proposed, two based on offline stabilization only and one that uses offline-online stabilization. Two main techniques for treating nonlinear terms are compared with each other: the discrete empirical interpolation method (DEIM) and a reduced quadrature method based on a sparse recovery technique. A number of numerical experiments are conducted on two benchmark flows: flow past a sphere and the 4:1 contraction flow problem. Significant computational speedup manifests with the proposed offline stabilization together with hyper-reduction as compared to the original formulation. Discussions about potential accuracy losses of the reduced order model (ROM) that could result from mesh-related issues of the full order model (FOM) is carried out, and possible remedies are proposed.

### 3.1 Part I: Fundamentals of rheology

#### 3.1.1 Introduction

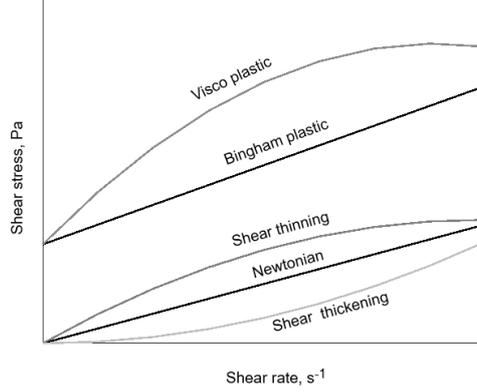
It is well known that the mathematical models describing small deformations for fluid substances under the assumption of continuum modeling is often explained by the Cauchy stress tensor. For a general 3D case, there are 6 shearing components and 3 normal components of the stress tensor which can be split into two parts given by:

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau} \quad (3.1)$$

where  $p$  is the isotropic part,  $\mathbf{I}$  is the identity operator, and  $\boldsymbol{\tau}$  is the deviatoric part of the stress tensor. In Newtonian fluids, the deviatoric stress tensor  $\boldsymbol{\tau}$  can be modeled with a constitutive law which is quite simple and is proportional to the rate of deformation tensor  $\mathbf{D}$  given by:

$$\boldsymbol{\tau} = 2\eta_s\mathbf{D} \quad (3.2)$$

where  $\eta_s$  is the dynamic viscosity and  $\mathbf{D} = \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^\top)$ . However, most of the materials which have industrial significance such as modeling of multi-phase fluids which includes polymer suspensions, emulsions, foams, composites, and slurries, to study complex biological systems, or soft matter, with classical equations of fluid mechanics cannot capture the intricacies of the deformation or flow behavior of such fluids as a result of which the deviatoric stress tensor evolves non-linearly in the rate of strain and the system of equations to describe the evolution state of the fluid may be nonlinear and strongly coupled. A brief description of different types of non-Newtonian behavior is explained in the following subsections.



**Figure 3.1:** A qualitative curve representation of typical Newtonian and non-Newtonian fluids [1].

### 3.1.2 Non-Newtonian fluids

This class of fluids shows non-Newtonian behavior in the sense that the applied shear stress deviates from the linear response with respect to the deformation rate or, simply put, the apparent viscosity is not a constant but a function of  $\sigma$  or  $\dot{\gamma}$ . The total stress tensor can then be expressed as,

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\eta_s(\dot{\gamma})\mathbf{D} \quad (3.3)$$

where  $\dot{\gamma}$  is the equivalent shear rate, computed as  $\dot{\gamma} = \sqrt{2\mathbf{D} : \mathbf{D}}$ . When the viscosity of the fluid is influenced only by the instantaneous shear rate, these fluids are termed generalized Newtonian fluids or purely viscous fluids. These fluids have no dependency on the deformation history, or in other words, no memory of past stresses.

Depending on the deformation rate or applied shear stress, these fluids can be classified as:

- Shear thinning fluids  $\frac{\partial\eta_s}{\partial\dot{\gamma}} < 0$
- Shear thickening  $\frac{\partial\eta_s}{\partial\dot{\gamma}} > 0$
- Viscoplastic fluids with or without shear thickening

The typical response of  $\tau - \dot{\gamma}$  is shown in the graph 3.1. Shear thinning is one of the most commonly seen fluidic behavior in engineering practice such as in food or cosmetic products like ketchup, honey, whipped creams, shampoos, etc. In such fluids, the apparent viscosity decreases with the increase in shear rate. On the other hand, when apparent viscosity increases with an increase in shear rate, these types of fluids are referred to as

shear thickening fluids. Typical examples include cornstarch water, thick suspensions, and so on. Another behavior observed in this class of fluids is the linear or non-linear response of  $\boldsymbol{\tau} - \dot{\boldsymbol{\gamma}}$  and it deforms only upon loading beyond yield stress. When the applied load is below a yield stress limit, the fluid acts as an elastic solid. These fluids are categorized as viscoplastic and when the flow curve is linear, the fluids are called Bingham plastic fluids. Examples include toothpaste, blood, yogurt, emulsions, foams, etc. The numerical modeling of the above-classified fluids has been extensively studied in the literature [116–119]. Most industrialized products display fluid characteristics where the apparent viscosity is not only a function of shear rate or applied stress, but also of the duration of the applied loading and their past kinematic histories. Such behavior can be observed commonly, for example in cement paste which happened to be sheared at a constant rate, followed by a long period of rest which eventually leads to the breakage of internal microstructures. This causes a decrease in viscosity until the rate of change of viscosity ultimately approaches zero. Depending on the response of deformation with time, this fluid can be further classified mainly into two types: thixotropic and rheotropic fluids.

### 3.1.3 Viscoelastic fluids

There is another category of fluids belonging to the non-Newtonian class also known as viscoelastic fluids and these are of primary concern to this thesis. These fluids exhibit a blend of both viscous and elastic responses to the applied forces. A familiar example of this phenomenon is the silly putty [120]. When the putty is applied a sudden deformation, it bounces like a perfectly elastic solid because its response time to deform is very large as compared to the time of observation, whereas if the putty is kept at rest for a long time without any external disturbances, it will flow and dissipate on a time scale governed by its viscosity. In this scenario, the flow is considered rheologically slow, as the time of observation is quite large compared to its response time for deformation. Such characteristic can be measured by what is known as the relaxation time of the fluid or its counterpart dimensionless parameter also known as Deborah number \*

$$De = \frac{\lambda}{t_c} \rightarrow \frac{\text{Relaxation time of the fluid}}{\text{Characteristic flow time scale}} \quad (3.4)$$

When  $De \rightarrow 0$ , it represents a purely viscous response, and when  $De \rightarrow \infty$  represents a purely elastic solid, and most engineering applications occur between these two extremes. In the prior discussions on non-Newtonian fluids where the apparent viscosity is shear dependent or time-dependent, the structure of the stress tensor is in a general sense like Newtonian fluids for a particular fluid flow and the velocity gradient acts instantaneously as stress changes and there is no presence of additional stresses in the fluid flow. While for such viscoelastic fluids also known as complex fluids, the structure of the stress tensor is quite different, and they do not behave like Newtonian counterparts. One consequence of this type of fluid behavior is the rise of extra stress tensors due to elastic components which are given by the normal stress differences in the direction normal to the applied shear stress, which is absent in generalized Newtonian fluids subjected to the same deformation history [115]. The normal stress differences  $\mathbf{N}_1$  and  $\mathbf{N}_2$  are also proportional to the shear rate and can be expressed in terms of normal coefficients as:

$$\begin{aligned} \psi_1 &= \frac{\mathbf{N}_1}{(\dot{\boldsymbol{\gamma}})^2} \\ \psi_2 &= \frac{\mathbf{N}_2}{(\dot{\boldsymbol{\gamma}})^2} \end{aligned} \quad (3.5)$$

---

\*Markus Reiner chose the name Deborah number inspired by a verse in the Bible, stating "The mountains flowed before the Lord" in a song by the prophetess Deborah in the Book of Judges [121].

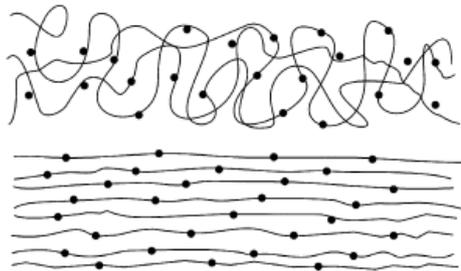
where,  $N_1 = \sigma_{xx} - \sigma_{yy}$  and  $N_2 = \sigma_{yy} - \sigma_{zz}$ . To quantify the viscoelastic effects, another dimensionless parameter is often used, which is given by the Weissenberg number. It is expressed as the ratio of elastic forces to viscous forces,

$$Wi = \frac{U\lambda}{l_c} = \frac{\lambda}{l_c/U} \quad (3.6)$$

Here, ' $l_c$ ' represents the characteristic length, and ' $U$ ' the characteristic velocity scale. For  $Wi < 1$ , the viscous forces are too weak, and so the fluid acts as inelastic, and for  $Wi > 1$ , the fluid starts to display viscoelastic effects.

**Remark:** Both the quantities Deborah number and Weissenberg number have been used interchangeably by rheologists quite often. This has been possible because the complex fluids are characterized by the relaxation time, and a single characteristic time and length and velocity scales do result in identical definitions [122]. It is to be noted that all these assumptions are made based on the unidirectional shear flows and depending on kinematic histories and different types of model flow like elongational or oscillatory motion, the viscoelastic behavior may vary. One can reverse engineer to change the structure of the fluid to impart desired rheological properties to a product that has industrial demand.

Polymer solutions, polymer melts, colloidal, suspensions, or worm micellar substances are some of the viscoelastic fluids [2, 123, 124]. At the molecular level, they are quite larger than the molecules of a Newtonian fluid, and also the relaxation time of such fluids is of a much higher order than regular Newtonian fluids, hence depending on the action of forces, even at very low shearing rates, micro-structural changes are observed in a time scale which can be captured in real-time. However, the continuum hypothesis is kept true while modeling the flow as long as the size of the micro-structures is much smaller than the characteristic length scales. For example, in figure 3.2, it can be seen that at very low shearing rates or at rest conditions, the micro-structures are oriented randomly and are coiled together corresponding to their minimum energy levels. This represents qualitatively that at low shearing rates, the polymer molecules resist deformation by exhibiting higher yield stress and this can be represented as an elastic response of the polymer. But upon high shearing rates, the tangled coiled-up polymer chains start detangling and straighten up and orient along the direction of streamlines, representing a purely viscous response. This polymer chain, on the other hand, influences the flow field and produces extra stress which is the viscoelastic stress tensor as an additional term in the governing equations.

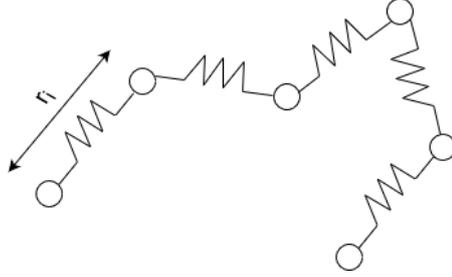


**Figure 3.2:** Schematic representation of a polymer chain at rest and under the influence of shear stress [1].

One of the simplest models of a polymer chain is an elastic dumbbell model consisting of two beads connected by a spring mass which is used for approximating a single polymer molecule shown in figure 3.3 and the total contribution of a single polymer chain is given by the sum of each individual molecules [125]. The orientation and elongation of the

dumbbell is represented by an end-to-end vector ' $\mathbf{r}$ ' with a fixed length  $r_i = |\mathbf{r}|$  and  $r_{eq}$  is the equilibrium length. The conformation tensor is a measure of the micro-structural changes in the state of the fluid which basically provides information about the state of stretch of the polymers, and is given by the statistical average of the dyadic product of the end-to-end vectors normalized with the equilibrium length squared,

$$\mathbf{C} = \frac{\langle \mathbf{r} \otimes \mathbf{r} \rangle}{r_{eq}^2} \quad (3.7)$$



**Figure 3.3:** A polymer chain represented by a random walk of step length ' $r_i$ ' in an elastic dumbbell model.

In the dumbbell model, the stress due to the polymers ( $\boldsymbol{\tau}_e$ ) is expressed in terms of the second-order conformation tensor and is linearly related  $\boldsymbol{\tau}_e = \frac{\eta_p}{\lambda}(\mathbf{C} - \mathbf{I})$ . There are many constitutive models in the literature [126] that appropriately relate the extra stress tensor ( $\boldsymbol{\tau}_e$ ) (or conformation tensor) with shear rate  $\dot{\boldsymbol{\gamma}}$ . One of the simplest models is given by the upper convected Maxwell model (UCM) also known as the Oldroyd-B model which can be expressed as:

$$\lambda \overset{\nabla}{\mathbf{C}} + \mathbf{C} - \mathbf{I} = 0 \quad (3.8)$$

where  $\eta_p$  is the polymer viscosity and  $\lambda$  is the relaxation time of the fluid. The symbol  $(\nabla)$  denotes the upper convected time derivative

$$\overset{\nabla}{\mathbf{C}} = \frac{\partial \mathbf{C}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{C} - (\nabla \mathbf{u})^\top \cdot \mathbf{C} - \mathbf{C} \cdot \nabla \mathbf{u} \quad (3.9)$$

where  $\nabla \mathbf{u} = \frac{\partial u_j}{\partial x_i}$ . Due to the hyperbolic nature of the equation (3.9), even efficient numerical solvers are severely limited by what is known as the high Weissenberg number problem [46]. The numerical evidence of this breakdown at moderately large Weissenberg numbers is due to the polynomial-based approximation to represent the stress (or conformation) tensor profiles, which are exponential in the regions of geometrical singularities or of large stress gradients. The exponential stress growth can be remedied by changing the variable that scales logarithmically with the conformation tensor  $\mathbf{S} = \log(\mathbf{C})$  [46]. Since the conformation tensor can be recomputed  $\mathbf{C} = \exp(\mathbf{S})$ , it retains the positive definiteness of the tensor even if  $\mathbf{S}$  contains large numerical errors. The constitutive equation (3.8) reformulated in terms of  $\mathbf{S}$  as:

$$\lambda \overset{\nabla}{\mathbf{S}} + \mathbf{S} - \mathbf{I} = 0 \quad (3.10)$$

Oldroyd-B model can only predict the first normal stress difference and is not able to capture the transverse normal stress in a shear flow, i.e.  $\sigma_{yy} = \sigma_{zz} = 0$  and then in a trivial sense normal coefficient  $\psi_2 = 0$ . In order to capture both the normal stress differences,

there is a need to switch to non-linear models which have higher-order derivatives. Some common second-order non-linear constitutive models that are able to quantitatively describe the behavior of viscoelastic fluids are finite extensible nonlinear elastic dumbbell with a Peterlin closure approximation (FENE-P) [127], Giesekus [128], and Phan-Thien Tanner (PTT) [129] models. The PTT model is derived from network theories of concentrated polymer solutions and melts, where the stress tensor (or log conformation tensor)  $\mathbf{S}$  satisfies the following equation [129]

$$\begin{aligned} \lambda \overset{\square}{\mathbf{S}} + Y(\text{tr } \mathbf{S})(\mathbf{S} - \mathbf{I}) &= 0 \\ Y(\text{tr } \mathbf{S}) &= \begin{cases} 1 + \epsilon(\text{tr } \mathbf{S} - 3) & \text{linear} \\ \exp[\epsilon(\text{tr } \mathbf{S} - 3)] & \text{exponential} \end{cases} \end{aligned} \quad (3.11)$$

$\overset{\square}{\mathbf{S}}$  denotes the following Gordon-Schowalter convected derivative operator, given by,

$$\overset{\square}{\mathbf{S}} = \frac{\partial \mathbf{S}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{S} - (\nabla \mathbf{u} - \xi \mathbf{D}) \cdot \mathbf{S} - \mathbf{S} \cdot (\nabla \mathbf{u} - \xi \mathbf{D})^T \quad (3.12)$$

In the PTT model, the dimensionless parameter  $\epsilon$  controls how fast the effective polymeric viscosity and the relaxation time decrease with the stress, and  $\xi$  is the slip parameter that is related to the second normal stress difference. When  $\epsilon = 0$ , the UCM model is recovered. In the Giesekus model,  $\mathbf{S}$  satisfies the following equation [128],

$$\lambda \overset{\nabla}{\mathbf{S}} + \mathbf{S} - \mathbf{I} + \alpha (\mathbf{S} - \mathbf{I})^2 = 0$$

Here,  $\alpha$  is the mobility parameter that modulates shear thinning and  $\overset{\nabla}{\mathbf{S}}$  denotes the Oldroyd's upper convected derivative operator (eq. 3.9). Giesekus model can predict both the normal stress difference and the second normal stress coefficient is always negative. If  $\alpha = 0$ , the Giesekus model simplifies to the UCM model.

## 3.2 Part II: Reduced order modeling for viscoelastic flows

### 3.2.1 Motivation and background

This part of the chapter is devoted to the surrogate modeling of viscoelastic fluid flows of second order via the POD reduced order model. In particular, the choice of the stabilization technique is discussed, as the approximation of the nonlinear terms in the reduced model framework. These two are closely related, as it is shown that the improper treatment of nonlinearity can affect the global numerical stability of the ROM.

One can find in the literature, a number of articles on the use of POD-ROM as well as greedy RBM methods on incompressible viscous fluid flow for CFD applications to turbulence modeling [48, 48, 49, 53, 57, 57–59]. Besides physical modeling, POD-ROMs have been extensively used for engineering applications such as for shape optimization, and flow control problems [60–62]. However, there are relatively few works in the literature addressing MOR for viscoelastic flows. Recent studies on modeling viscoelastic fluid flows can be found in the literature [130, 131]. In the article [130], the authors have established a POD-Galerkin approach for modeling isotropic turbulence flow of FENE-P fluid by DNS and have demonstrated well the efficiency, stability, and robustness of the ROM for computing the flow fields, while in [131], the authors applied POD-ROM to study a thermal flow problem governed by Boussinesq approximation for an Oldroyd-B model with intrinsic stabilized closure models. Both papers concentrated on the turbulence mechanism, however, our area of interest is in studying the flow behavior of purely polymeric fluids for

moderately high Weissenberg numbers. Also, for the first time non-linearity is addressed in this manuscript for second-order viscoelastic fluids, which were not addressed in the earlier works.

The problem of modeling complex fluids is indeed quite challenging because of the nature of the governing equations which are hyperbolic, highly nonlinear, non-affine in parametric dependence, and strongly coupled. In the finite element (FE) formulation, stabilization is required in both the momentum balance and the constitutive model equations.

In Newtonian cases, the discretization of velocity and pressure field satisfies the LBB condition required for the stability of the numerical simulation. However, in viscoelastic fluids, due to the dependency of the viscoelastic stress tensor on velocity gradients, there arises some compatibility issues in the discretization space between the velocity field and the viscoelastic stress tensor. There are many improvements in the numerical techniques to treat and maintain the elliptic nature of the momentum equation for small solvent viscosities such as elastic viscous split stress(EVSS/FEM) developed by Rajagopalan et al. [132], discrete elastic viscous split stress-Galerkin(DEVSS-G) by Guénette and Fortin [133], and discontinuous Galerkin method (DG) by Fortin and Fortin [134]. Such numerical treatment allows using the same discretization spaces for the elastic stress tensor and the additional variable. In this work, the DEVSS-G method has been implemented followed by the work of D’Avino et al. [135] for smooth interpolation of velocity gradient in the constitutive equation by considering velocity gradient as an additional dependent variable. By doing so the number of scalar degrees of freedom for 3D models adds to 19: 3 for velocity, 1 for pressure, 6 for stress tensor, and 9 for velocity gradient (or equivalently, 10 in 2D). Therefore, it is clear that viscoelastic fluid flows is a high dimensional problem and that, despite the use of powerful computers, invoking the full-order model for the evaluation of all solutions in a vast parametric space for multi-query problems is still time-constrained.

Similarly, to introduce the elliptic component in the constitutive equation in case of smaller solvent viscosities, artificial diffusion in the upwind direction such as streamline upwind diffusion (SUPG) [136] needs to be applied. In addition, due to the hyperbolic nature of the constitutive equation, even efficient numerical solvers are severely limited by what is known as the high Weissenberg number problem [46]. The Weissenberg number is a dimensionless parameter, which is a ratio between the elastic relaxation time and a time corresponding to the local rate of deformation. A high Weissenberg number means that the history dependence of viscoelastic stress tensor is evident. The numerical evidence of codes to breakdown at moderately high Weissenberg numbers is due to the polynomial-based approximation to represent the viscoelastic stress tensor (or conformation tensor) profiles which are exponential in the regions of large stress gradients, near stagnation points, or geometrical singularities (example re-entrant corners) [137, 138]. However, by operating instead on the logarithm of the conformation tensor, which may be referred to in the works [46], the exponential stress growth can be effectively handled. This adds additional complexity to an already high-dimensional nonlinear problem.

It is widely known that projection to a lower dimensional space alone is insufficient for nonlinear problems to achieve computational efficiency, thus numerous hyper reduction strategies have been created [139–142] to specifically address the cost of assessing nonlinear terms, typically by empirical interpolation or empirical quadrature, in order to get around this bottleneck. Unfortunately, the interpolation-based method focuses primarily on the approximation of the nonlinear terms and very little attention has been paid to preserving the properties of the resulting reduced models such as preservation of symmetry (for the FOM operators) or numerical stability [143] which can be reflected in our numerical analysis of the benchmark problems. Hence, we concentrate on the empirical quadrature procedure, which has demonstrated efficacy for finite element-based systems. By only

requiring assembly over a tiny area of the mesh, as specified by an extremely sparse vector of element weights generated as the solution of an optimization problem, this method returns computing efficiency [109].

Our goal in this chapter is to present a reduced-order modeling framework to deal with all the above-mentioned points. First, we deal with the reduction of degrees of freedom per spatial node in the ROM framework. By assuming divergence-free velocity space and by static condensation of the unknown velocity gradient variable, we can easily eliminate the pressure and the auxiliary velocity gradient dofs from the ROM. Second, we focus on the stability of the ROM. The hypothesis that the solutions generated from a stabilized high-fidelity solver will recover stable basis functions for a ROM construction, may not always be true. Residual-based stabilization such as Galerkin Least Squares (GLS), Petrov Galerkin (PG), and SUPG techniques have been widely used to tackle the stability issues for convection-dominated problems in both FOM level and ROM level. The authors in reference [144] have adopted both offline-only and offline-online strategies for advection-dominated PDEs, parabolic PDEs, and even stochastic-based problems. In the offline-online strategy, the same stabilized bilinear form is used in both stages and offline only consists of stabilizing only in the offline phase, followed by classical Galerkin projection in the online stage. In the article [145], the authors have extended such stabilization strategies to Navier Stokes flow problems. In this work, we observe that an offline-only strategy, may not always produce satisfactory results and can cause numerical noise along the boundary layers, especially if there is complexity in the geometry or the presence of singularity. On the other hand, the offline-online strategy aims to work in the same order of accuracy as FOM which is expected but at the cost of the original system. Thus, we propose an alternate offline strategy where the objective of establishing a stable system in the reduced framework is achieved at a significantly lower cost than offline-online stabilization. Hence, with hyper reduction and an offline stabilization strategy, a very stable ROM at an exceptional computational speed up within a certain accuracy is achieved and is demonstrated in this chapter.

Finally, the efficiency of the non-linear term computations in the online stage is addressed by some hyper reduction techniques based on DEIM and a reduced quadrature rule based on the FOCUSS algorithm [113].

We begin by presenting the FOM framework for viscoelastic fluid flows, followed by a discussion on the computation of basis functions and construction of the surrogate model. Finally, we demonstrate the proposed ROM methodology by solving two benchmark problems: Flow past a sphere and 4:1 contract flow problem, followed by a discussion on the numerical analysis.

### 3.2.2 Problem setting of a viscoelastic fluid flow

#### Governing Equations

The flow behavior of an inertialess viscoelastic fluid is governed by the momentum balance and continuity equations given by (3.13). In this work, the flow is considered to be incompressible and with no body forces.

$$\begin{aligned}\nabla \cdot (2\eta_s \mathbf{D} + \boldsymbol{\tau}_e(\mathbf{S}) - p\mathbf{I}) &= 0 \\ \nabla \cdot \mathbf{u} &= 0\end{aligned}\tag{3.13}$$

where  $\mathbf{u}$  represents the velocity vector field,  $p$  is the scalar pressure field,  $\mathbf{D}$  is the strain rate tensor, and  $\eta_s$  is the solvent viscosity.  $\boldsymbol{\tau}_e(\mathbf{S})$  needs to be specified by choosing appropriate constitutive models which can relate  $\boldsymbol{\tau}_e$  with the velocity gradient or deformation

history. In this work, the second-order Giesekus fluid is adopted to model the constitutive equation, following the work of D'Avino et al. in [135]. The expression is given by,

$$\begin{aligned} \frac{\partial \mathbf{S}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{S} - (\nabla \mathbf{u})^\top \cdot \mathbf{S} - \mathbf{S} \cdot \nabla \mathbf{u} - \mathbf{g}(\mathbf{S}) &= 0 \\ \mathbf{g}(\mathbf{S}) &= \frac{\mathbf{S} - \mathbf{I}}{\lambda} + \frac{\alpha(\mathbf{S} - \mathbf{I})^2}{\lambda} \end{aligned} \quad (3.14)$$

Nonetheless, this formulation can accommodate other constitutive models as well such as Oldroyd-B, FENE-P, PTT, Pom-Pom model, and so on. Finally, the governing equations (3.13) combined with constitutive model (3.14) forms a well-posed system in the unknowns  $\mathbf{u}$ ,  $p$  and  $\mathbf{S}$  specified with appropriate boundary conditions.

The boundary conditions are similar to the Stokes system

$$\begin{aligned} \mathbf{u} &= 0 \text{ on } \Gamma_{D_o} \\ \mathbf{u} &= \mathbf{u}_D \text{ on } \Gamma_{D_g} \\ -p\mathbf{n} + 2\eta_s \mathbf{D} \cdot \mathbf{n} + \boldsymbol{\tau}_e \cdot \mathbf{n} &= \mathbf{t}_N \text{ on } \Gamma_N \end{aligned}$$

where the boundary  $\Gamma = \Gamma_{D_o} \cup \Gamma_{D_g} \cup \Gamma_N$  has been split into a homogeneous Dirichlet, non-homogeneous Dirichlet and a Neumann part. Due to the hyperbolic nature of the constitutive equation, an additional boundary condition is needed for  $\mathbf{S}$  at the inflow boundary:

$$\mathbf{S} = \mathbf{S}_{\text{in}} \text{ on } \Gamma_{\text{in}}$$

where  $\Gamma_{\text{in}}$  is the inflow boundary, i.e. the part of  $\Gamma$  where  $\mathbf{u} \cdot \mathbf{n} < 0$ , with  $\mathbf{n}$  the outwardly directed unit normal. Due to the assumption of an inertialess flow, only an initial condition for  $\mathbf{S}$  is required.

### Finite element formulation

To obtain the variational equivalent of the strong form equations (3.13) and (3.14) in 2D spatial dimension, we set the following finite element spaces:  $\mathcal{T}^{\mathcal{N}'} = (L^2(\Omega))^{2 \times 2}$ ,  $\mathcal{V}^{\mathcal{N}} = (H_0^1(\Omega))^2$ ,  $\mathcal{P}^{\mathcal{N}'} = L^2(\Omega)$ ,  $\mathcal{G}^{\mathcal{N}'} = (L^2(\Omega))^{2 \times 2}$ . Recall:  $L^2(\Omega) = \{q, \int_\Omega q^2 d\Omega < \infty\}$  and  $H_0^1(\Omega) = \{q \in L^2(\Omega), \nabla q \in (L^2(\Omega))^2 | q|_{\partial\Omega} = 0\}$ .

The weak forms for the FE formulation read as: find  $(\mathbf{u}_h, p_h, \mathbf{S}_h) \in \mathcal{V}^{\mathcal{N}} \times \mathcal{P}^{\mathcal{N}'} \times \mathcal{T}^{\mathcal{N}'}$  such that:

$$-\langle \nabla \cdot \mathbf{v}_h, p_h \rangle + \langle (\nabla \mathbf{v}_h), 2\eta_s \mathbf{D}_h \rangle = \langle \mathbf{v}_h, \nabla \cdot \boldsymbol{\tau}_{he} \rangle + \langle \mathbf{v}_h, \mathbf{t}_N \rangle_{\Gamma_N} \quad \forall \mathbf{v}_h \in \mathcal{V}^{\mathcal{N}} \quad (3.15)$$

$$\langle q_h, \nabla \cdot \mathbf{u}_h \rangle = 0 \quad \forall q_h \in \mathcal{P}^{\mathcal{N}'} \quad (3.16)$$

$$\langle \mathbf{E}_h, \frac{\partial \mathbf{S}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{S} - (\nabla \mathbf{u})^\top \cdot \mathbf{S} - \mathbf{S} \cdot \nabla \mathbf{u} - \mathbf{g}(\mathbf{S}) \rangle = 0 \quad \forall \mathbf{E}_h \in \mathcal{T}^{\mathcal{N}'} \quad (3.17)$$

But experience shows that Galerkin FEM techniques are not very useful for viscoelastic flows, as there is a compatibility issue between the velocity field and extra stress tensor approximation spaces. So, techniques such as DEVSS-G proposed by Bogaerds et al. [146] are used to stabilize the momentum equations in this work. An additional auxiliary variable, velocity gradient tensor,  $\mathbf{G}_h = \nabla \mathbf{u}_h^\top$  is added to the momentum equation which adds real diffusion and stabilizes the term. For simple notation, we drop the 'h' from the superscript. The stabilized weak form of the momentum equation is then given by,

$$-\langle \nabla \mathbf{v}, \eta_a (\nabla \mathbf{u} - \mathbf{G}^\top) \rangle - \langle \nabla \cdot \mathbf{v}, p \rangle + \langle \nabla \mathbf{v}, 2\eta_s \nabla \mathbf{D} \rangle = \langle \mathbf{v}, \nabla \cdot \boldsymbol{\tau}_e \rangle + \langle \mathbf{v}, \mathbf{t}_N \rangle_{\Gamma_N} \quad \forall \mathbf{v} \in \mathcal{V}^{\mathcal{N}} \quad (3.18)$$

$$\langle \mathbf{H}, -\nabla \mathbf{u} + \mathbf{G}^\top \rangle = 0 \quad \forall \mathbf{H} \in \mathcal{G}^{\mathcal{N}'} \quad (3.19)$$

**Remark:**  $\eta_a$  is an artificial viscosity added to achieve numerical stability so that the system matrix obtained is symmetric, and also the value is kept constant over the domain. Here, in this work,  $\eta_a = \eta_p$  is chosen.

The presence of the convection term in the constitutive equation causes the numerical scheme to be unstable in convection-dominated cases. So Galerkin projection cannot produce good results, and stabilization techniques like SUPG are required where the test function is modified [147] by adding artificial diffusion in the upwind direction. The modified test function in the constitutive equation can be expressed as:

$$\langle \mathbf{E} + \tau \mathbf{u} \cdot \nabla \mathbf{E}, \frac{\partial \mathbf{S}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{S} - (\nabla \mathbf{u})^\top \cdot \mathbf{S} - \mathbf{S} \cdot \nabla \mathbf{u} - \mathbf{g}(\mathbf{S}) \rangle = 0 \quad \forall \mathbf{E} \in \mathcal{F}^{\mathcal{N}'} \quad (3.20)$$

Here,  $\tau$  in eq. (3.20) is a scalar field of dimension time given by  $\tau = \frac{\beta h}{2\|\mathbf{u}\|}$  where  $\beta$  is a dimensionless constant that varies between (0,1) and is dependent on the local Péclet number,  $h$  is a typical finite element mesh size in the direction of the velocity and  $\|\mathbf{u}\|$  is the characteristic flow velocity.

**Discretization spaces:** For the numerical tests presented in this work, the flow domain  $\Omega$  is discretized utilizing a certain triangulation  $\mathcal{T}$ . Mixed FE discretization spaces are employed for the fields  $(\mathbf{u}, p, \mathbf{G}, \mathbf{S})$ . Taylor Hood  $\mathbb{P}2 - \mathbb{P}1$  (continuous) elements have been used for discretizing velocity and pressure fields respectively, which are Ladyzhenskaya-Babuska-Brezzi (LBB) stable [148]. Now for the other pair  $(\mathbf{G}, \mathbf{S})$ , the same discretization space as used for the pressure field is used for discretizing the scalar components of  $\mathbf{S}$  and  $\mathbf{G}$  which is  $\mathbb{P}1 - \mathbb{P}1$  (continuous) elements. The time integration of the governing equations is performed using a variable step variable order (VSVO) Adams-Bashforth-Moulton PECE solver, with orders ranging from 1 to 13 [149, 150].

### 3.2.3 Model order reduction methodology

#### Parametrization of the complex fluid

In this section, a projection-based POD approach is discussed for the construction of the ROM. In the POD formulation, a set of basis functions are constructed from a collection of snapshots taken at different time instances or for any parameter from a discrete training set,  $\mu \in \Xi_{train} \in \mathbb{R}^N$  of the FOM. Assume that an already enriched dataset for velocity field ( $\mathbf{u}$ ), pressure field ( $p$ ), velocity gradient tensor ( $\mathbf{G}$ ), viscoelastic stress tensor ( $\boldsymbol{\tau}_e$ ) and log conformation tensor ( $\mathbf{S}$ ) is available. The snapshots can be obtained in the offline stage by solving a high-fidelity solver for varying input parameters ranging from boundary conditions to constitutive parameters like the relaxation time ( $\lambda$ ), the polymer viscosity ( $\eta_p$ ) or the solvent viscosity ( $\eta_s$ ), to geometrical and shape parameters.

#### Implementation of POD-ROM

There are certain assumptions made in the study of the models such as no shape variation in the geometry is considered, therefore the mesh is fixed regardless of the choice of

the parameters. As a consequence of this choice, the subspace for the velocity field is automatically divergence-free as the reduced basis is extracted from divergence-free velocity snapshots. For this reason, the pressure unknowns can be eliminated from the model formulation. If pressure is needed, it can be determined separately in a post-processing step once the ROM is solved. Following the same rationale, the velocity gradient unknowns can also be eliminated using static condensation, that is, including an additional term in the momentum equation resulting from the Schur complement of the velocity gradient unknowns and ensuring DEVSS-G stabilization. It has to be noted that this operation is in theory also possible in the FOM. However, due to the size of the resulting system of equations, the Schur complement would result in a large dense matrix, making the momentum equation intractable. In the ROM, on the other hand, this operation does not affect the complexity of the reduced system of equations. The associated reduced weak form of the problem now only includes the momentum balance equation and the constitutive law counterpart to the FE statement (3.15), and (3.17). Its construction is discussed in the following subsection.

### Reduced form and related issues

The reduced weak form is obtained by replacing  $\mathbf{u}$  and  $\mathbf{S}$  in (3.15), and (3.17) by  $\mathbf{u}_{\text{POD}}$  and,  $\mathbf{S}_{\text{POD}}$  which are expressed as:

$$\mathbf{u}_{\text{POD}}(\mathbf{x}, t; \mu) = \sum_{j=1}^r \boldsymbol{\phi}^j(\mathbf{x}) z_j(t; \mu) \quad (3.21)$$

$$\mathbf{S}_{\text{POD}}(\mathbf{x}, t; \mu) = \sum_{j=1}^{r'} \boldsymbol{\Theta}^j(\mathbf{x}) w_j(t; \mu) \quad (3.22)$$

where, the reduced basis functions  $\{\boldsymbol{\phi}^j\}_{j=1}^r$  and  $\{\boldsymbol{\Theta}^j\}_{j=1}^{r'}$  are obtained by,

$$\boldsymbol{\phi}^j = \text{POD}([\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^N]) \in \mathbb{R}^{\mathcal{N}} \quad (3.23)$$

$$\boldsymbol{\Theta}^j = \text{POD}([\mathbf{S}^1, \mathbf{S}^2, \dots, \mathbf{S}^{N'}]) \in \mathbb{R}^{\mathcal{N}'} \quad (3.24)$$

$\{z_1, \dots, z_r\}^T$  and  $\{w_1, \dots, w_{r'}\}^T$  represents the coefficients of the POD expansion of the velocity field and log conformation tensor, respectively.

### Projection-based reduction

In the second step, the weak forms are projected onto the properly selected low-dimensional subspaces. For the momentum equation, we obtain the standard Galerkin approach, i.e. selecting the test functions as the  $\boldsymbol{\phi}^i(\mathbf{x})$ .

$$\langle \nabla \boldsymbol{\phi}^i, \eta_a \nabla \mathbf{u}_{\text{POD}} + 2\eta_s \mathbf{D}_{\text{POD}} \rangle + \langle \nabla \boldsymbol{\phi}^i, \mathcal{S}(\mathbf{u}_{\text{POD}}) \rangle = \langle \boldsymbol{\phi}^i, \nabla \cdot \boldsymbol{\tau}_e \rangle + \langle \boldsymbol{\phi}^i, \bar{\mathbf{q}} \rangle_{\Gamma_N} \quad \forall i = 1, \dots, r \quad (3.25)$$

where  $\mathcal{S}(\mathbf{u}_{\text{POD}})$  is the linear form associated with the Schur complement of the velocity gradient unknowns.

The reduced weak form for the constitutive equation reads as:

$$\langle \boldsymbol{\Psi}^i, \frac{\partial \mathbf{S}_{\text{POD}}}{\partial t} + \mathbf{u}_{\text{POD}} \cdot \nabla \mathbf{S}_{\text{POD}} - (\nabla \vec{u}_{\text{POD}})^\top \cdot \mathbf{S}_{\text{POD}} - \mathbf{s}_{\text{POD}} \cdot \nabla \mathbf{u}_{\text{POD}} \rangle = \langle \boldsymbol{\Psi}^i, \mathbf{g}(\mathbf{S}_{\text{POD}}) \rangle \quad \forall i = 1, \dots, r' \quad (3.26)$$

Note that the terms in the left-hand side of the reduced momentum equation (3.25) are affine with respect to the parameters  $\eta_a$  and  $\eta_s$  and can be efficiently reduced during the offline stage. By way of example, we consider the reduction of the first term:

$$\langle \nabla \phi^i, \eta_a \nabla \mathbf{u}_{\text{POD}} \rangle = \eta_a \sum_{j=1}^r \langle \nabla \phi^i, \nabla \phi^j \rangle z_j, \quad (3.27)$$

in which the  $(r \times r)$  operator  $\langle \nabla \phi^i, \eta_a \nabla \mathbf{u}_{\text{POD}} \rangle$  can be computed once and for all in the offline stage. During the online stage, if new parameters  $\eta_a$  and  $\eta_s$  are prescribed, the evaluation of the left-hand side operator of the momentum balance equation can be done in reduced complexity (i.e. it does not depend on the original dimension  $\mathcal{N}$ ) since it only requires  $\mathcal{O}(r \times r)$  operations. This step is crucial for retaining the computational efficiency of the ROM, however, the same idea cannot be straightforwardly applied to the right-hand sides of the equations (3.25) and (3.26), as the nonlinearity of  $\boldsymbol{\tau}_e$  and  $\mathbf{g}$  makes the evaluation of the corresponding terms in the reduced weak form is dependent on the number of quadrature points in the FOM since this operation cannot be performed offline. Particular attention is due when dealing with non-linear terms for improving the efficiency of the offline-online strategy. This will be discussed in the following section.

The second challenge to be addressed is the choice of the test functions  $\Psi^i(\mathbf{x})$  on the constitutive equation (3.26). This depends on the stability properties of the resulting scheme that is applied in the online solution. As already mentioned in the introduction, we discuss three possible choices in section 3.2.3.

### Reduction of the non-linear terms

In this work, we experimented with the reduction of the nonlinear terms appearing in (3.25) and (3.26) using two of the most commonly used and representative approaches: DEIM [108] and reduced integration (or hyper-reduction) [113]. In DEIM, the non-linear terms are approximated by using a low-rank interpolative basis in order to construct an affine decomposition. The method not only specifies the way to construct the basis, but it also determines the reduced set of optimal interpolation points at which the nonlinear function must be evaluated. These can be selected either among the nodes of the grid or the set of quadrature points. In this case, we chose to extract the optimal points from the nodal points. To build the interpolation functions, first, we need to generate snapshots of the non-linear terms. For instance, to apply this methodology to the nonlinear term in the momentum balance (3.25), first we need to create snapshots of the extra-stress tensor by computing  $\boldsymbol{\tau}_e$  as a function of  $\mathcal{S}$ , and generate its set of basis vectors  $\{\boldsymbol{\xi}^i\}_{i=1}^{r''} = \text{POD}([\boldsymbol{\tau}_e^1, \boldsymbol{\tau}_e^2, \dots, \boldsymbol{\tau}_e^N]) \in \mathcal{N}'$ . The POD expansion of the viscoelastic stress tensor is then given by:

$$\boldsymbol{\tau}_e(\mathbf{x}, t; \mu) \approx \sum_{j=1}^{r''} \boldsymbol{\xi}^j(\mathbf{x}) m_j(t; \mu), \quad (3.28)$$

where the POD coefficients  $m_j(t; \mu)$  can be expressed as linear combination of values of  $\boldsymbol{\tau}_e$  sampled at nodal points  $\mathbf{x}_p$ :

$$m_j(t; \mu) = \sum_{p=1}^{r''} \mathbb{E}_{jp} \boldsymbol{\tau}_e(\mathbf{x}_p, t; \mu) \quad j = 1, \dots, r'' \quad (3.29)$$

The interpolation points can be determined through the DEIM algorithm using the greedy approach as explained in detail in chapter 2.2. Once the interpolation points are

found, the operator  $\mathbb{E}$  can be determined as the inverse of the linear operator  $\boldsymbol{\xi}^j(\mathbf{x}_p)$ , whose rows corresponds to the evaluation of the basis functions,  $\{\boldsymbol{\xi}^j\}_{j=1}^{r''}$  at points  $\mathbf{x}_p$ .

Now in a discrete setting, the nonlinear function in the momentum balance (3.25) can be represented as:

$$\langle \boldsymbol{\phi}^i, \nabla \cdot \boldsymbol{\tau}_e(\mathbf{x}, t; \mu) \rangle = \sum_{j,p=1}^{r''} \langle \boldsymbol{\phi}^i, \nabla \cdot \boldsymbol{\xi}^j \rangle \mathbb{E}_{jp} \boldsymbol{\tau}_e(\mathbf{x}_p, t; \mu) \quad i = 1, \dots, r \quad (3.30)$$

Here, the operator  $\sum_{j,p=1}^{r''} \langle \boldsymbol{\phi}^i, \nabla \cdot \boldsymbol{\xi}^j \rangle \mathbb{E}_{jp}$  can be computed once and for all during the offline stage. In the online stage, now the right-hand side of the momentum balance eq. (3.25) can be evaluated by computing  $\boldsymbol{\tau}_e$  at only  $\mathbf{x}_p$  points which has a single operation of performing a matrix-vector product, whose complexity is now of the  $\mathcal{O}(r \times r'')$  drastically reduced from  $(\mathcal{N} \times \mathcal{N}')$ . It has to be remarked that the term  $\langle \boldsymbol{\phi}^i, \nabla \cdot \boldsymbol{\xi}^j \rangle$  is linear when the viscoelastic stress tensor  $\boldsymbol{\tau}_e(\mathbf{x}, t; \mu)$  is considered as the primary variable but in the algebraic formulation, this term is computed from  $\mathbf{S}$  which has a nonlinear relationship, thus computing this term at fewer DEIM points can lead to increase in efficiency of the ROM.

On the other hand, in the hyper reduction (HR) formulation, we seek for direct approximation of the integral  $\int_{\Omega} \boldsymbol{\phi}^i(\mathbf{x}) \nabla \cdot \boldsymbol{\tau}_e(\mathbf{x}, t; \mu) d\Omega, \forall i = 1, \dots, r$  based on Gaussian quadrature rule instead of finding an approximation of the vector projected onto the set of interpolative basis functions. Given the classical quadrature rule we have:

$$\langle \boldsymbol{\phi}^i, \nabla \cdot \boldsymbol{\tau}_e(\mathbf{x}, t; \mu) \rangle = \sum_{k=1}^{\mathcal{N}_q} \omega_k \boldsymbol{\phi}^i(\mathbf{x}_k) \nabla \cdot \boldsymbol{\tau}_e(\mathbf{x}_k, t; \mu) \quad i = 1, \dots, r \quad (3.31)$$

The fundamental idea of the hyper reduced formulation is to approximate (3.31) as a weighted sum of the integrand evaluated at a reduced set of quadrature points such that:

$$\langle \boldsymbol{\phi}^i(\mathbf{x}), \nabla \cdot \boldsymbol{\tau}_e(\mathbf{x}, t; \mu) \rangle \approx \langle \boldsymbol{\phi}^i(\mathbf{x}), \nabla \cdot \boldsymbol{\tau}_e(\mathbf{x}, t; \mu) \rangle_{\text{HR}} \quad (3.32)$$

where,

$$\langle \boldsymbol{\phi}^i(\mathbf{x}), \nabla \cdot \boldsymbol{\tau}_e(\mathbf{x}, t; \mu) \rangle_{\text{HR}} = \sum_{q=1}^{r''} \omega_q^{\text{HR}} \boldsymbol{\phi}^i(\mathbf{x}_q) \nabla \cdot \boldsymbol{\tau}_e(\mathbf{x}_q, t; \mu) \quad r'' \ll \mathcal{N}_q \quad (3.33)$$

The use of sparse or reduced quadrature rule implies that the points  $\mathbf{x}_q \subset \mathcal{N}_q$  need to be carefully selected from among those of the full quadrature points. Eq. (3.33) requires  $r''$  to be known a priori, so in fact we approximate:

$$\langle \boldsymbol{\phi}^i(\mathbf{x}), \nabla \cdot \boldsymbol{\tau}_e(\mathbf{x}, t; \mu) \rangle_{\text{HR}} \approx \sum_{q=1}^{\mathcal{N}_q} \omega_q^{\text{HR}} \boldsymbol{\phi}^i(\mathbf{x}_q) \nabla \cdot \boldsymbol{\tau}_e(\mathbf{x}_q, t; \mu) \quad i = 1, \dots, r \quad (3.34)$$

The approximation becomes exact if,

$$\|\boldsymbol{\omega}^{\text{HR}}\|_{\ell_0} = r''$$

where, the operator  $\|\cdot\|_{\ell_0}$  determines the number of non-zero elements in the vector of integration weights,  $\boldsymbol{\omega}^{\text{HR}}$ . The position of the reduced integration points  $\mathbf{x}_q$  and their

associated positive weights  $\boldsymbol{\omega}^{\text{HR}}$  can be determined by solving the following optimization problem such that the volume of the physical domain is exactly integrated and also the positive quadrature weights are retained:

$$\begin{aligned}
(\boldsymbol{\omega}^{\text{HR}}, \boldsymbol{x}_q) &= \arg \min_{\boldsymbol{\omega}} \sum_{i=1}^r \sum_{k=1}^N \left( \langle \phi^i(\boldsymbol{x}) \nabla \cdot \boldsymbol{\tau}_e^k(\boldsymbol{x}, t; \mu) \rangle - \langle \phi^i(\boldsymbol{x}_q) \nabla \cdot \boldsymbol{\tau}_e^k(\boldsymbol{x}_q, t; \mu) \rangle_{\text{HR}} \right)^2 + \gamma \|\boldsymbol{\omega}\|_{\ell_0}, \\
\sum_{j=1}^{\mathcal{N}_q} \omega_j &= |\Omega|, \\
\omega_j &\geq 0 \quad \forall j
\end{aligned} \tag{3.35}$$

In the above minimization problem, the term  $\gamma \|\boldsymbol{\omega}\|_{\ell_0}$  is added explicitly to introduce sparsity.  $\gamma$  is a regularization parameter that can be tuned as per the user's requirements to prioritize sparsity over accuracy when  $\gamma \gg 1$  or vice versa. As explained in chapter 2, solving  $\|\boldsymbol{\omega}\|_{\ell_0}$  is a daunting task as it requires solving a non-negative least squares minimization problem which can be very expensive for a large scale system as such and time complexity of solving the above optimization problem becomes NP-hard. Another alternative would be to replace  $\ell_0$ -norm with a surrogate norm, which oftentimes is  $\ell_1$ -norm. But this choice is not consistent with the constraint expressed in the optimization problem (3.35), since the sum of the quadrature weights is exactly equivalent to the  $\ell_1$ -norm, and therefore cannot be minimized. Instead, we find sparse representation by using  $\ell_p$  quasi norm with  $0 < p < 1$ , which is achieved with the FOCUSS algorithm as explained in detail in chapter 2.

The major advantage of such a cubature-based reduction approach is that the quadrature weights are guaranteed to be positive which is crucial for retaining the spectral properties of the FE operators, which may not be the case in interpolation-based approaches like DEIM. This may cause the numerical simulation to have some instability in the ROM, which we will discuss shortly in the numerical test analysis.

### Choice of the stabilization for the constitutive equation.

**Offline only stabilization (POD-G):** The first choice is to apply no stabilization for the ROM, which is achieved by performing classical POD-Galerkin projection on the constitutive law. So, the projection space is nothing but the same subspace spanned by the reduced basis function of  $\boldsymbol{S}$  given by:

$$\{\boldsymbol{\Psi}^i\}_{i=1}^{r'} = [\boldsymbol{\Theta}^1, \boldsymbol{\Theta}^2, \dots, \boldsymbol{\Theta}^{r'}] \in \mathbb{R}^{\mathcal{N}' \times r'} \tag{3.36}$$

The reason for not applying stabilization in the reduced model is based on the assumption that the snapshots generated from a stabilized FOM will inherently reproduce basis functions that are stable and noise-free. In theory, this is expected to be true for the advection-diffusion equation even for small diffusion coefficients [144], but since our constitutive law has no diffusion term at all it is not obvious that the basis functions constructed could lead to a stabilized ROM.

**Offline-online stabilization (POD-SUPG):** In the second choice, we use the same stabilization scheme (SUPG) that was used in the FOM i.e. performing reduced approximation by POD-SUPG in the online stage. The only difference with the stabilization term in ROM is that it is expressed in terms of the reduced basis function of  $\boldsymbol{S}$  given by:

$$\{\boldsymbol{\Psi}^i\}_{i=1}^{r'} = [(\boldsymbol{\Theta}^1 + \tau \boldsymbol{u}_{\text{POD}}(\boldsymbol{x}, t; \mu) \cdot \nabla \boldsymbol{\Theta}^1), \dots, (\boldsymbol{\Theta}^{r'} + \tau \boldsymbol{u}_{\text{POD}}(\boldsymbol{x}, t; \mu) \cdot \nabla \boldsymbol{\Theta}^{r'})] \in \mathbb{R}^{\mathcal{N}' \times r'} \tag{3.37}$$

However, this introduces further non-linear coupling because the projection operator dependency on  $\mathbf{u}_{\text{POD}}$  is non-affine (since  $\tau = \frac{\beta h}{2\|\mathbf{u}_{\text{POD}}\|}$ ). Despite the projection subspace being expressed in terms of the basis function  $\{\boldsymbol{\Theta}^i\}_{i=1}^{r'}$  which can be computed offline once and all, the FOM operator has to be updated online for each time step and for all the training samples, which are not cost-effective at all.

**Offline only stabilization (POD-PG):** In the third choice, the residual of the constitutive equation is projected onto a subspace spanned by a fixed basis computed in the offline step once and for all, similar to the first offline stabilization approach. The only difference is in the generation of the reduced bases that are obtained by applying dimensionality reduction to the set of snapshots from the FOM defined by:

$$\{\boldsymbol{\Psi}^i\}_{i=1}^{r'} = \text{POD}([\mathbf{S}^1 + \tau \mathbf{u}^1 \cdot \nabla \mathbf{S}^1], \dots, [\mathbf{S}^N + \tau \mathbf{u}^N \cdot \nabla \mathbf{S}^N]) \in \mathbb{R}^{N'} \quad (3.38)$$

Hence, in this offline stabilization, the reduction is performed by POD-PG approximation. Since now the rank of the test function subspace is unrelated to the one of the subspace for,  $\mathbf{S}$  we make sure the number of modes for the projection space is always at least equal to or more than the number of modes we have in the basis for  $\mathbf{S}$ . The reason for choosing more number of modes for the test function subspace  $\boldsymbol{\Psi}^i$  is because of the presence of a derivative term in the equation (3.38), and by enriching with additional modes in the ROM can well capture the micro-scale change in the flows, thus providing an accurate representation of the FOM. In case the resulting system is over-determined, we proceed to solve it in the least-squares sense. A detailed comparison is made among the above-mentioned stabilization approaches in the numerical analysis section 3.3.

### 3.3 Numerical analysis of two benchmark problems.

The above-discussed POD-ROM is herein assessed and validated for two classical benchmark cases: Flow past a sphere and 4:1 contraction flow problem. In this work, the parameterized complex fluid is analyzed and studied by varying a single input flow parameter which is the relaxation time of the fluid ( $\lambda$ ) or in this case the Deborah number (see for definition (3.4)), while keeping the rest of the fluid parameters fixed.

**Remark:** Quantities like Deborah number (3.4) and Weissenberg number (3.6) can be considered equivalent and result in identical definitions [122] when  $t_c = \dot{\gamma}_c^{-1}$ , with  $\dot{\gamma}_c = \frac{U}{l_c}$ . Here, ' $l_c$ ' represents the characteristic length, and, ' $U$ ' is the characteristic velocity scale which is given by the inlet velocity for the investigated numerical tests. In benchmark 1, the characteristic length is the radius of the sphere and in benchmark 2, it is given by the width of the channel.

Finally, for every Deborah number in the training set,  $\lambda \in \mathbb{R}^{N_{hf}}$  and for each time step  $\{t_j\}_{j=0}^n$  the field variables  $\mathbf{u}, p, \mathbf{S}, \boldsymbol{\tau}_e$  including the nonlinear functions are computed and assembled into different snapshot matrices. Using POD, the reduced basis functions are obtained respectively as explained in the section 3.2.3. It should be noted that the same time stepping scheme is used in the ROM as well with an equal tolerance limit in order to produce a fair comparison when estimating the computational speed-ups. The tolerance limit for the time discretization solver is set to  $10^{-4}$ . The error metrics for quantifying the quality of the ROM are discussed in the subsequent section, followed by the discussion on the geometrical, material, and numerical parameters for the benchmark cases.

### 3.3.1 Posteriori error metrics

The reconstruction error or the ROM error is obtained by evaluating the maximum of  $\mathcal{L}_2$  norm between the full order solution and reduced solution of log-conformation tensor  $\mathbf{S}$  over time and then finding the average value of the error among all the  $\lambda$  values from the training set.

$$\text{ROM error} = \frac{1}{N_{hf}} \max_{t_j} \frac{\sqrt{\langle \tilde{\mathbf{S}}(t_j), \tilde{\mathbf{S}}(t_j); \lambda_k \rangle}}{\sqrt{\langle \mathbf{S}(t_n), \mathbf{S}(t_n); \lambda_k \rangle}} \quad \begin{matrix} j = 0, \dots, n \\ \forall k=1, \dots, N_{hf} \end{matrix} \quad (3.39)$$

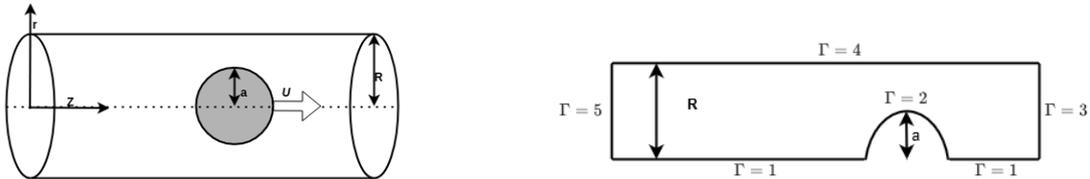
$\langle \tilde{\mathbf{S}}(t_j), \tilde{\mathbf{S}}(t_j); \lambda_k \rangle$  represents the sum of squared  $\mathcal{L}_2$  norm error over the sampled Deborah numbers, for each time step. The error is taken as the difference between FOM solution  $\{\mathbf{S}(t_j)\}_{j=0}^n \in \mathbb{R}^{\mathcal{N}'}$  and ROM solution  $\{\mathbf{S}_{\text{POD}}(t_j)\}_{j=0}^n \in \mathbb{R}^{\mathcal{N}'}$ . Similarly,  $\langle \mathbf{S}^k(t_n), \mathbf{S}^k(t_n) \rangle$  represents the sum of the squared of  $\mathcal{L}_2$  norm of the FOM snapshots over  $\lambda_k$  at the final  $n^{\text{th}}$  time step. We choose to scale the error with respect to the fully developed  $\mathcal{L}_2$  norm of  $\mathbf{S}$  since this value is more representative of the characteristic magnitude of the log conformation tensor at steady state, which is our main quantity of interest. We also evaluate the error between the snapshots and their orthogonal projection on a POD basis. This POD error quantifies the low-rank approximation of the snapshots and serves as a lower bound for ROM error given by,

$$\text{POD error} = \frac{1}{N_{hf}} \max_{t_j} \frac{\sqrt{\langle \mathbf{S}_{\text{proj}}(t_j), \mathbf{S}_{\text{proj}}(t_j); \lambda_k \rangle}}{\sqrt{\langle \mathbf{S}(t_n), \mathbf{S}(t_n); \lambda_k \rangle}} \quad \begin{matrix} j = 0, \dots, n \\ \forall k=1, \dots, N_{hf} \end{matrix} \quad (3.40)$$

where,  $\mathbf{S}_{\text{proj}}(t_j) = \mathbf{S}(t_j) - \langle \boldsymbol{\Theta}^i, \mathbf{S}(t_j) \rangle \boldsymbol{\Theta}^i$ ,  $i = 1, \dots, r'$ .

### 3.3.2 Description of the problems

**Flow past a sphere:** A sphere of radius  $a$  is placed with its center on the axis of the cylindrical tube of radius,  $R$  as shown in figure 3.4. This problem is axisymmetric around the axis, thus the 3D geometry can be reduced to 2D. The sphere moves with a constant velocity  $U$  along the axis of symmetry, or it is equivalent to moving the cylinder with velocity  $-U$  by keeping the sphere fixed. External boundaries are represented by  $\Gamma = \bigcup_{i=1}^5 \Gamma_i$ . For velocity field, Dirichlet conditions are applied, i.e.  $\mathbf{u} = (-U, 0)$  on the boundary  $\Gamma = 3$  and  $\Gamma = 4$ . Symmetry on  $\Gamma = 1$  and the no-slip condition is applied at  $\Gamma = 2$ . At time  $t = 0$ ,  $\mathbf{S} = \mathbf{0}$  on inlet and symmetry, i.e. at  $\Gamma = 3$ , and  $\Gamma = 1$  respectively. Since inertia is neglected, no initial condition is required for the velocity field.

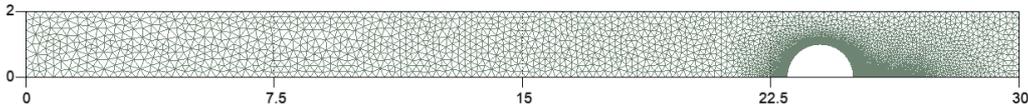


**Figure 3.4:** (left) Schematics of the flow past a sphere benchmark, (bottom) computational domain.

**4:1-Step flow problem:** A sketch of this benchmark problem is shown in figure 3.6. This problem is also axisymmetric, thus the 3D geometry can be reduced to 2D. External boundaries are represented by,  $\Gamma = \bigcup_{i=1}^8 \Gamma_i$  as shown in figure 3.7. The boundary conditions applied are: Symmetry on  $\Gamma = 1$  and  $\Gamma = 2$ . A constant velocity field, in the

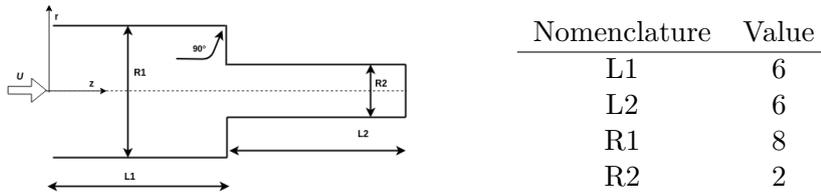
Geometrical parameter	Dimension
Length of the domain	30.0
Height of the domain	2.0
Radius of the sphere $a$	1.0
Mid-position of the sphere	24.0

**Table 3.1:** Geometrical parameters



**Figure 3.5:** Mesh representation

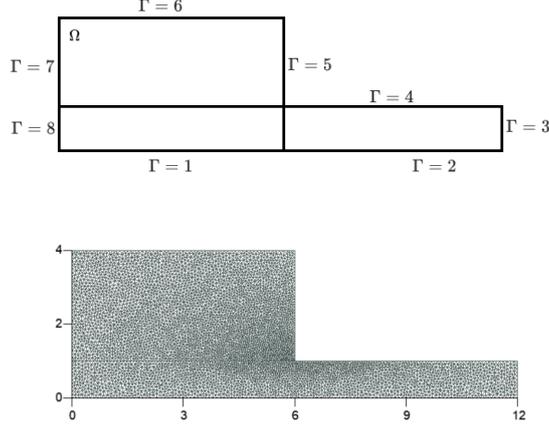
x-direction,  $\mathbf{u} = (U, \theta)$  is applied on inlet  $\Gamma = 7$  and  $\Gamma = 8$ . Homogeneous Dirichlet condition,  $\mathbf{u} = (0, 0)$  is applied on walls  $\Gamma = 4, 5$ , and  $6$ . At the time  $t = 0$ ,  $\mathbf{S} = \mathbf{0}$  on inlet and symmetry.



**Figure 3.6:** Schematics of the step flow benchmark.

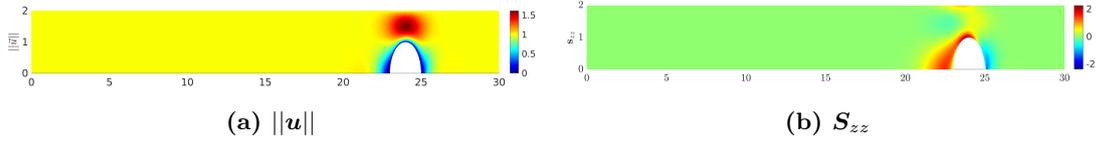
The physical domain and the mesh are built using the GMSH software [151]. The region around the singular points where large gradients can be expected is further refined than the rest of the geometry in both the benchmark problems, as shown in the figures 3.5 and 3.7. For all our experiments, we adopt a mobility coefficient  $\alpha = 0.2$  and a viscosity ratio  $\frac{\eta_s}{\eta_p + \eta_s} = 0.1$ . Simulations are run up to  $t = 20$ , which is enough to ensure that all flows reach a steady state.

The snapshots are generated for input flow parameter  $\lambda$  lying in the range  $[0.1, 1.5]$  discretized in ten linear steps. The solutions are stored for 200-time steps for each  $\lambda$  with a logarithmic spacing of the time samples. We use logarithmic spacing because we want the dataset of snapshots to be more representative of the flow behavior during the initial transient phase and avoid redundant information from oversampling the steady state (during which most of the snapshots are almost identical). The training set consists of 2000 snapshots. The 2D velocity magnitude profile and log conformation tensor in axial direction for both the benchmark cases are shown in figures 3.8 and 3.9 respectively for  $\lambda = 0.5667$ . It can be observed from the velocity profile in fig. 3.8a that the boundary conditions are satisfied, and the flow magnitude is highest between the no-slip spherical surface and the top wall of the domain for the flow past the sphere problem, while the re-entrant region for the 4:1 contraction flow problem as expected. For the log conformation tensor profile, we observe that they are highest along the leading edge of the sphere (see

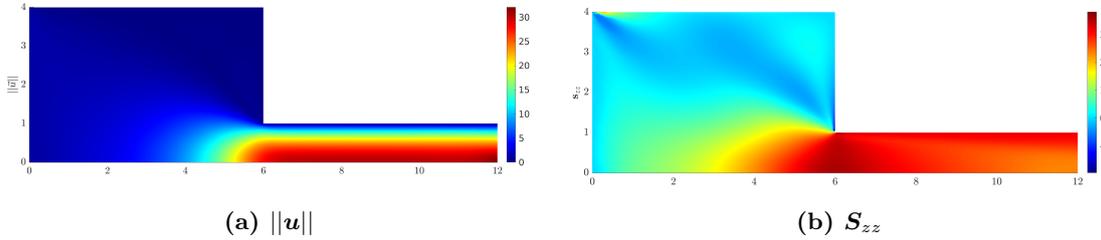


**Figure 3.7:** (*Top*) Computational domain, (*bottom*) mesh representation

fig. 3.8b) as compared to the rest of the domain and along the incoming flow region and the sharp corner for the contraction flow problem in figure 3.9b.



**Figure 3.8:** (a) 2D velocity magnitude profile , and (b) log conformation tensor  $S_{zz}$  profile in axial direction for  $\lambda = 0.5667$  for flow past the sphere benchmark.



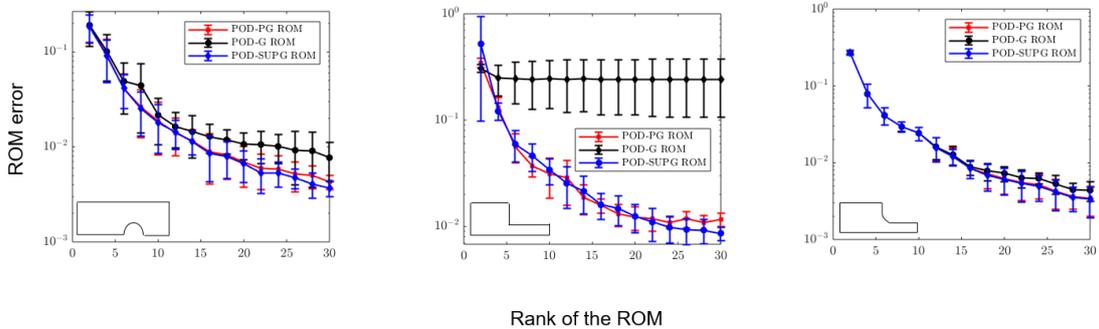
**Figure 3.9:** (a) 2D Velocity magnitude profile and (b) log conformation tensor  $S_{zz}$  profile in axial direction for  $\lambda = 0.5667$  for 4:1-step flow benchmark.

### 3.3.3 Discussions

The constitutive equation in the FOM is solved using SUPG stabilization to obtain solutions void of spurious oscillations and numerical instability. As explained previously, we construct the ROM with three different stabilization techniques, upon which the residual of the constitutive equation is projected, and make a comparative analysis in both the benchmark cases. Figure 3.10 represents the ROM error of the log-conformation tensor and the respective standard deviation of all the  $\lambda$  values from its mean error against the rank of the solution for both the benchmark cases. It can be observed from the figure that the ROM error obtained using offline stabilization (POD-PG), and online stabilization (POD-SUPG) are in good agreement in both cases and has an accuracy of the same order  $\mathcal{O}(10^{-2} - 10^{-3})$ . On the other hand, when ROM is constructed using no stabilization (POD-G), we observe

a completely different trend for the two cases as seen on the left and center plots of figure 3.10. For flow past the sphere benchmark problem, the ROM error appears to have a smooth decay with a slower convergence as compared to the other stabilization techniques applied and might require more than 30 modes to achieve the same order of accuracy. On the other hand, for the 4:1-step flow problem, the reconstruction error starts stagnating from the 5<sup>th</sup> mode onwards and even further enrichment with POD modes does not improve the ROM error. This could be due to the geometry of this benchmark problem, which has a sharp entrant corner in the flow direction, producing pronounced singular effects. Hence, the POD modes recovered from the FOM solutions may not be completely stable, which could lead to stagnancy in the convergence plot.

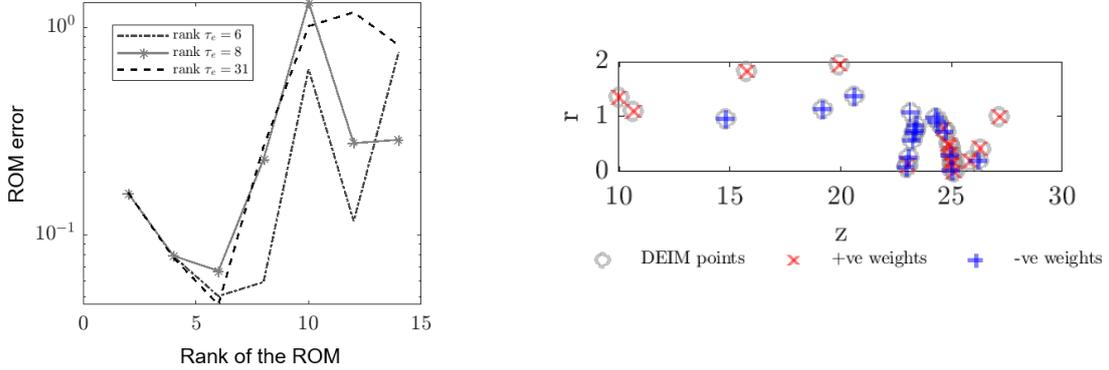
To verify this assumption, we slightly change the geometry from a sharp corner to a slightly rounded corner by maintaining the arc length at 1.6. We also changed the inlet velocity from a constant value to a parabolic velocity profile, i.e.  $\mathbf{u} = (1 - \frac{z^2}{16}, 0)$  where  $z$  is the direction perpendicular to the flow. By implementing this strategy in our work, we observe a smooth decay of the reconstruction error as seen on the right side of the figure 3.10 for all three cases of stabilization.



**Figure 3.10:** ROM error of the solution  $\mathbf{S}$  for the benchmark cases: flow past a sphere (*left*), step flow with sharp corner (*middle*) and step flow with rounded corner (*right*).

We can therefore infer from our observation that the ROM built with no stabilization may not always produce stable numerical solutions, especially for problems that are mesh sensitive or which have a singularity in the geometry, but the proposed offline strategy retains numerically stable solution and comparable results as with the online stabilization at least in the range of  $\mathcal{O}(10^{-2})$  to  $\mathcal{O}(10^{-3})$  within the 25 modes even in the complex geometry. By projecting the residual of the constitutive equation on the offline reduced bases, we avoid computation of the additional non-linear function online at each time step, reducing the computational time by,  $\mathcal{O}(10)$  as will be discussed below.

**Reduction for non-linear terms:** So far, we have discussed the ROM without considering non-linear reduction. As already mentioned, using only linear reduction techniques for non-linear systems is not viable as there is no significant reduction in the CPU time, in such scenarios the essence of model order reduction is lost. Our next step is to take into consideration the reduction of non-linear terms present in both the momentum and constitutive equation, as explained in the section 3.2.3. First, we begin with the reduction of only the momentum balance non-linear term (3.25) using DEIM for the sphere benchmark case. We obtain a set of 31 optimal DEIM points based according to the truncated rank of the input basis of  $\boldsymbol{\tau}_e$ . Finally, in the ROM, we evaluate the non-linear term  $\langle \nabla \phi, \boldsymbol{\tau}_e(\mathbf{x}, t; \mu) \rangle$  only at the selected interpolation points. The right side of the figure 3.11 represents the location of DEIM points, and it can be seen that the points are located mostly near the spherical region, which is quite natural as the flow behavior is expected to change in that



**Figure 3.11:** ROM error of the solution  $\mathbf{S}$  for different rank of  $\tau_e$  (*left*), location of DEIM points and its nature of corresponding weights (*right*).

area. The left side of the figure 3.11 shows the ROM error of the solution  $\mathbf{S}$  and the respective standard deviation of all the  $\lambda$  values from its mean error for three different truncated modes of  $\tau_e$ . It can be observed, that the reconstruction error blows up after a certain rank of the ROM is reached i.e. after 6 modes only, the error starts diverging in all three cases. This could be due to the reason that interpolation-based approaches like DEIM cannot preserve the stability properties of the original numerical scheme used for the FOM. To confirm this assumption, we evaluate the integration weights corresponding to the DEIM points obtained. We draw parallels to the integration weights because the approximation of the non-linear term using DEIM can be considered equivalent to an integral scheme solved using the quadrature rule. Hence, the integrals of the interpolative basis functions are equivalent to the Gaussian weights. The nature of the integration weights is shown in the plot of figure 3.11. Negative weights are represented with a red ‘ $\times$ ’ symbol, and blue ‘+’ represents the positive nature of the weights. It can be seen that DEIM cannot retain the positive nature of the integration weights for certain points, which is very crucial for preserving the positive definiteness of the finite element matrices. We do not exceed DEIM points by enriching with more than 31 modes because as observed the integration weights are negative in nature and going beyond 31 would not improve the solution and produce a similar kind of divergence. Since DEIM failed to produce an accurate result for the flow past the sphere benchmark case, we do not test further for the step flow problem.

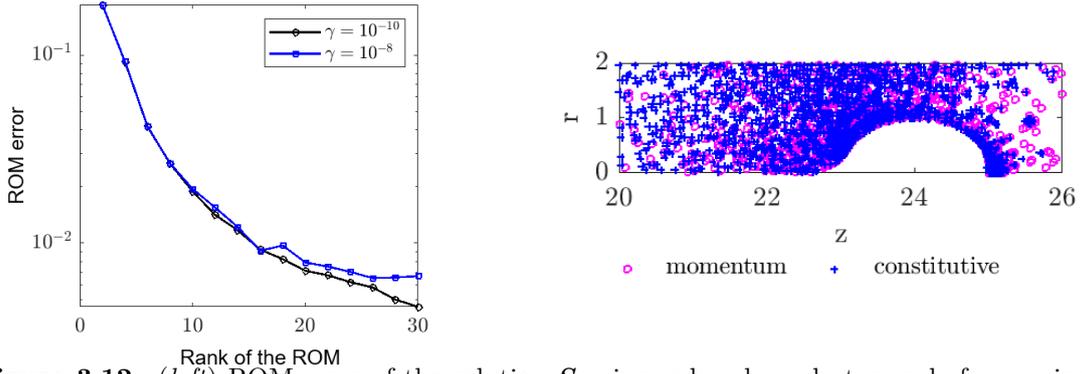
In the next step, we approach the non-linear reduction of both the terms in the momentum equation and constitutive equation with a reduced quadrature rule using the FOCUSS algorithm. By tuning the regularization parameter  $\gamma$  and the norm power  $p$  in the minimization problem for the sparse quadrature recovery rule as shown in 3.2.3, we can reduce or increase the quadrature points while maintaining a strict tolerance limit. In this work, we choose to keep  $p$  fixed at  $p = 0.5$ , and by fine-tuning  $\gamma$  values to  $\gamma = 10^{-8}$  and  $\gamma = 10^{-10}$  while setting tolerance limit for accuracy at  $\epsilon_{tol} = 10^{-10}$ , we obtain different quadrature points respectively shown in table 3.2.

The total quadrature points in the flow past the sphere problem is 50475 and the contraction flow problem is 74913. From the data in the table 3.2, it can be seen that the quadrature points have been reduced to less than 10% of the total quadrature points in both the benchmark cases. Finally, in the ROM, we evaluate the non-linear terms present in both equations at the selected set of quadrature points. Figure 3.12 and 3.13 represent the location of these reduced set of Gaussian points and are mostly located in the region

where flow gradients are expected to change. For example, in the flow past the sphere problem, the reduced points are mostly located near the spherical region and toward the leading edge of the sphere. While, for the step flow benchmark problem, the points are mostly located near the incoming flow direction and sharp re-entrant region. The positivity of weights is ensured while solving the minimization problem, which leads to a stable hyper ROM.

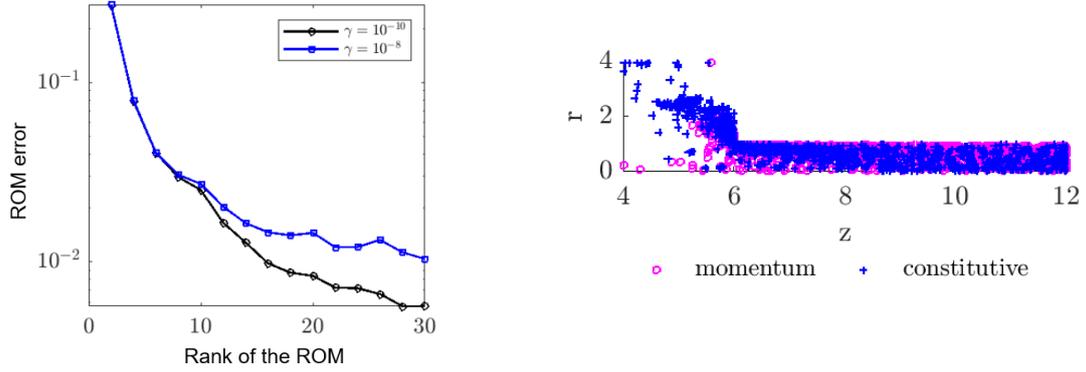
**Table 3.2:** Reduced quadrature points

Benchmark 1		
	Constitutive	Momentum
$\gamma = 10^{-8}$	2718	4638
$\gamma = 10^{-10}$	2943	4830
Benchmark 2		
	Constitutive	Momentum
$\gamma = 10^{-8}$	4479	4746
$\gamma = 10^{-10}$	5586	5772

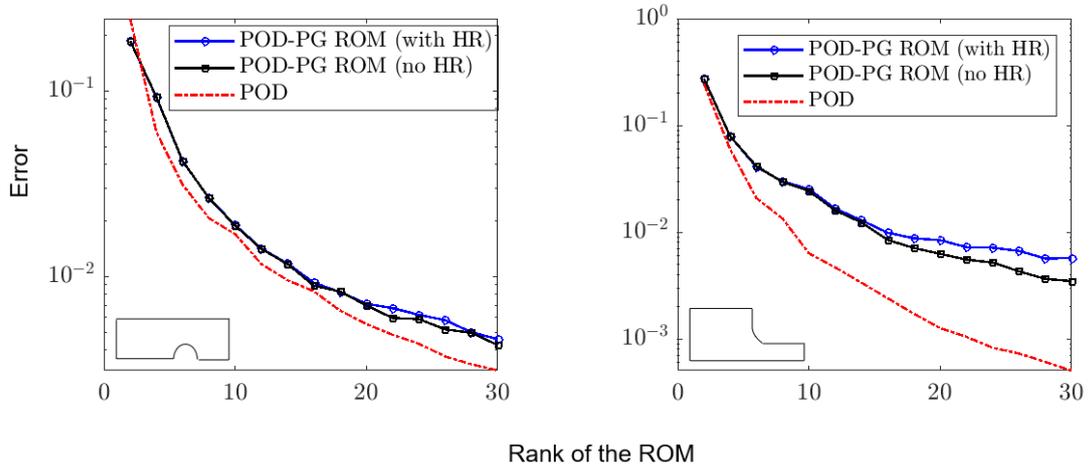


**Figure 3.12:** (*left*) ROM error of the solution  $\mathbf{S}$  using reduced quadrature rule for varying  $\gamma$  values of Benchmark 1, (*right*) location of reduced quadrature points.

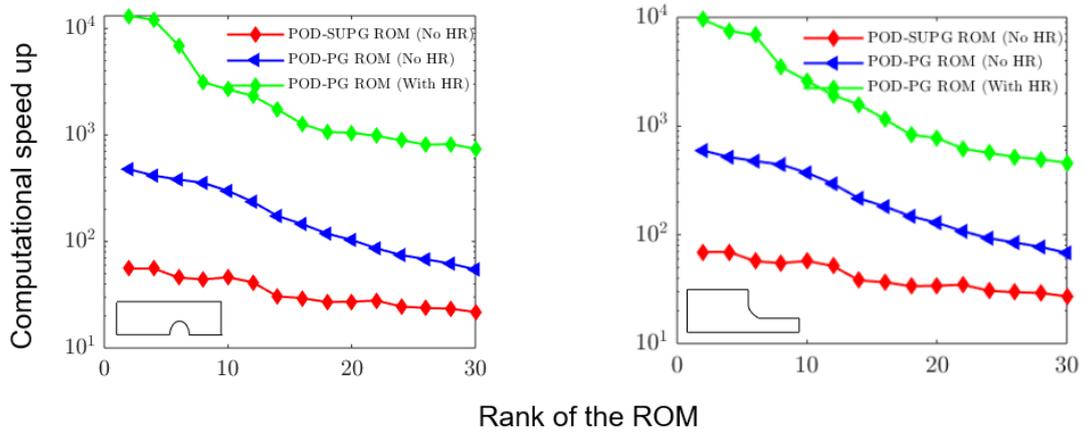
This is realized from the plot shown in figure 3.12 and 3.13 where the ROM error of the solution  $\mathbf{S}$  is shown against the rank for different values of regularization parameter  $\gamma$ . A smooth decay of ROM error is observed when  $\gamma = 10^{-10}$ , for both the benchmark cases with an accuracy of  $\mathcal{O}(10^{-2} - 10^{-3})$ , which is the same as the order of accuracy when no hyper reduction is applied. It should be noted that these results are achieved using offline stabilization only. When  $\gamma = 10^{-8}$ , the accuracy is of the same order up to at least 26 modes for the sphere benchmark and after that, it starts stagnating, on the other hand for the contraction flow problem the convergence is a bit slower for  $\gamma = 10^{-8}$  and may require more than 30 modes to achieve same order of accuracy as obtained with  $\gamma = 10^{-10}$ . These results are observed for the rounded geometry case and may not have the same consistency when the geometry has a sharp corner. Further, a comparative plot of ROM error with POD projection error is shown in figure 3.14 and as expected POD error decreases monotonically and is lower than ROM achieving an order of accuracy  $\mathcal{O}(10^{-3})$  serving as a lower bound for ROM error. With the hyper reduction, we also achieve an excellent computational speed-up of almost twice the order of the reduced solution obtained without hyper reduction as shown in figure 3.15 or in other words, it can be said that it is  $\mathcal{O}(10^4)$  times faster than the solution obtained using FOM.



**Figure 3.13:** (left) ROM error of the solution  $\mathcal{S}$  using reduced quadrature rule for varying  $\gamma$  values of Benchmark 2, (right) location of reduced quadrature points.



**Figure 3.14:** POD projection error and ROM error of  $\mathcal{S}$  with offline stabilization only (POD-PG) with and without hyper reduction for both the benchmark cases.



**Figure 3.15:** Computational speed ups obtained without hyper reduction using POD-PG ROM and POD-SUPG ROM, and POD-PG ROM with hyper reduction for both the benchmark cases.

### 3.4 Conclusions

We introduced a hyper reduction framework for the dimensionality reduction of complex fluids like viscoelastic flows, and a thorough analysis has been conducted. The algebraic problem is greatly simplified in the ROM formulation. By construction of the reduced basis for the velocity field, the incompressibility constraint is automatically satisfied, thus removing the pressure degrees of freedom. We intrinsically removed DEVSS-G stabilization from the momentum equation, thereby removing the extra velocity gradient degrees of freedom from the system. Stabilization of the advection term in the constitutive equation can be simplified through an offline stabilization procedure instead of an expensive SUPG. We observed that the use of interpolation techniques such as DEIM in the reduction of the nonlinear terms can cause a systematic loss of accuracy of the ROM. This can be avoided using a sparse quadrature approach and ensuring the non-negativity of the resulting quadrature weights. Even if the number of nonlinear function evaluations is in general higher in this case, this approach still achieves  $(10^4)\times$  speed-ups with ROM average errors in the range of  $\mathcal{O}(10^{-2} - 10^{-3})$ . In the benchmark analyzed, the POD-PG stabilization strategy combined with the hyper reduction of the non-linear terms appears to be the trade-off between accuracy and complexity while avoiding the need for online stabilization. Since stabilization concerns the offline stage only, the complexity of the ROM is freed from the dimension of the original FOM, therefore it is reasonable to expect even more significant speed-ups for 3D flows using the proposed approach. In this work, we considered only one input parameter and assumed the dataset of snapshots to be sufficiently rich for achieving a good quality ROM but for a multidimensional parametric system, the sampling space has to be sufficiently rich to accurately represent the large-scale system, and large sample space means the computation of high fidelity models for each sample point. Even though it only has an offline cost, it still is expensive and needs to be minimized. Therefore, the next chapter is dedicated to an adaptive sampling strategy for achieving an optimal set of parametric points for dealing with complex problems with efficiency and robustness.





## Chapter 4

# Iterative multi-fidelity modeling for multi-dimensional input parametric PDE systems

### Abstract

In this chapter, we propose a parametric sampling strategy for the reduction of large-scale PDE systems with multi-dimensional input parametric spaces, and also present its applicability to an inertialess flow of parameterized viscoelastic fluid for the first time. By leveraging models of different fidelity, the design of this methodology allows a user to adaptively sample points ad hoc from a discrete training set with no prior requirement of residual-based error estimators. It is achieved by exploiting low-fidelity models throughout the parametric space to sample points using an efficient sampling strategy, and at the sampled parametric points, high-fidelity models are evaluated to recover the reduced basis functions. Since the proposed methodology leverages the use of low-fidelity models to assimilate the solution database, it significantly reduces the computational cost in the offline stage. The highlight of this chapter is to present the construction of the low-fidelity model, and a sampling strategy based on the discrete empirical interpolation method (DEIM).

## 4.1 Introduction

### 4.1.1 Motivation and background

The convergence and efficiency of a ROM for approximation of the solutions of a large-scale PDE system depends heavily on the choice of the elements that constitute the "reduced basis" [152], or in other words, the input parameter selection has to be appropriate for which the snapshots are generated. It needs to be rich enough to adequately cover important regions of the parametric space. There are discretization schemes commonly used for sampling the parametric space, such as uniform sampling, and random sampling. But both sampling has certain drawbacks, for example, an extensive grid search in uniform sampling would become expensive in a multi-dimensional parametric system while random sampling would miss some crucial parts of the function in the parametric space. On the other hand, another sampling technique, known as Latin Hypercube Sampling (LHS) [153, 154] provides a compromise between uniformity and size of the sample, which makes it more efficient than uniform sampling and gives often better accuracy than random sampling. Some statistically-based sampling methods like Monte Carlo methods are also among other popularly used techniques [155]. Even with efficient sampling techniques, the complexity can

grow exponentially with the increase in the dimensionality of the parametric space, which is the case for many engineering applications. Therefore, the reduced basis modeling (RBM) based on greedy sampling emerged as a promising tool for reducing the computational cost of FOM by generating high-fidelity snapshots at only a select few optimal parametric points.

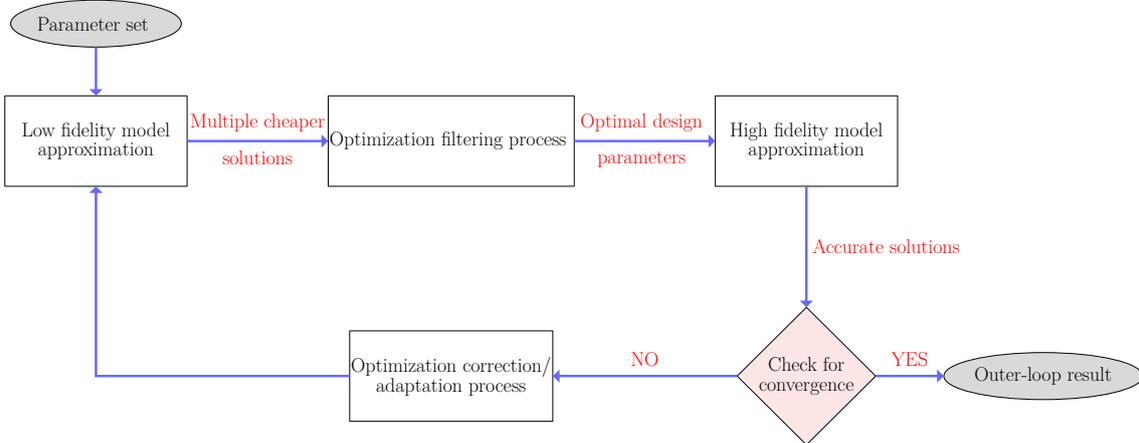
The basic idea in RBM is to adaptively choose sample points by finding the location at which the estimated error of the reduced model is maximum in the offline phase, thanks to rigorous error estimators. The greedy RBM was a subject of research for a very long time, first applied to find reduced models for the parameterized steady incompressible Navier–Stokes equations [49] and then further developed for a variety of parameterized parabolic PDEs [54, 55, 156] and also applied to several optimal control and inverse problems [157, 158]. During the past two decades, RBM had a significant contribution to the development of rigorous error bounds for Stokes flow problems, with a special focus on the inf sup stability conditions that can be referred to in the articles [48, 53, 59].

Even though, the RBM methods are not completely void of bottlenecks; it requires sharp, and rigorous error estimators that are problem specific for ensuring the reliability of the method. Additionally, the error is estimated over a discrete training set, which must be a good surrogate of the continuous parametric space. Inefficient greedy sampling could yet occur from this, particularly for high dimensional parametric PDEs. To mitigate this issue, the authors in the article [159] have performed the greedy algorithm on random training sets of small sizes in every iterative cycle instead of estimating the error over the entire training set. The authors have successfully demonstrated a 16-parametric dimension system for a diffusion equation problem. Wilcox et al. in their research work [160], solved a sequence of optimization problems on a parametric space which is not a discrete set but rather continuous, to find optimal points adaptively in a greedy manner using both error estimator or residual error indicator. The authors demonstrated the proposed methodology on a thermal problem for the design of a fin based on two input parameters Biot number and conduction coefficient for 11, and 21 parametric dimensions respectively. The papers [161–163] also provide references to quite a few literary works on goal-oriented sampling methods. Numerous other studies have examined the use of subspace angles to assess the model’s sensitivity to parameter changes [164–166] or sensitivity analysis to sample adaptively from the parametric space [167].

#### 4.1.2 Overview of the idea: Iterative multi-fidelity model order reduction

In this work, we suggest a sampling strategy that uses a multi-fidelity modeling approach as an alternative to the conventional greedy sampling technique that is driven by error estimators. Multi-fidelity modeling employs models of varied accuracy to estimate the same output quantity at a considerably faster convergence rate than using a single model with a higher approximation quality [168]. For instance, in optimization problems, an optimizer is constructed to supply the input design variables at each iteration, and the model then assesses the value of the related objective function, its corresponding gradients, and the value of the constraint. Typically, high-fidelity models are retained in the loop to establish accuracy and convergence guarantees on the low-fidelity models, which are used to determine the best design parameters while taking advantage of computing speedups [169]. Low-fidelity model estimates are cheaper model approximations that can be a coarse-discretized model, a ROM, or even a simplified physics model that can approximate the same output as the high-fidelity model but with lower accuracy. Figure 4.1 represents a flow of the multi-fidelity approach applied to an optimization problem. Multi-fidelity methods have been successfully applied in optimization problems based on co-kriging models [169],

uncertainty analysis [170], Monte Carlo simulations [171, 172] to name a few. Such methods have also been extended to the machine learning and Physics-Informed Neural Network (PINN) domain, which can be referred to in [173, 174]. The authors in [173] have presented the connection of fidelity of different accuracy with neural networks by manipulating the width and depth of the network architecture. A comprehensive review of the past works and recent advances in the area of multi-fidelity modeling can be found in the survey [168].



**Figure 4.1:** Multi-fidelity for outer loop applications

Recently, in the work of Kast et al. [175], a multi-fidelity setup is exploited in the context of reduced-order modeling by solving a nonlinear structural dynamic problem based on three input parametric dimensions. First, an appropriate sampling set is exploited in the parametric space by leveraging a collection of low-fidelity models, followed by multi-fidelity Gaussian Process Regression (GPR) for approximation of the reduced coefficients in the online stage, therefore allowing an efficient decoupling of offline-online strategy. Another work based on a multi-fidelity adaptation strategy can be found in the work [90] where the authors combined data-driven models with projection-based ROM and adapted the ROM for any change in the input parameter by low-rank updates to the reduced operators without rebuilding the FE operators from scratch. This work is focused on addressing the complexities of cases when the underlying properties of the PDE system are not static, but undergo dynamic changes due to the change in the latent variables.

Motivated by the multi-fidelity modeling approach, we conjunct the multi-fidelity method with physics-based reduced order modeling for deriving low-cost ROMs efficiently without the need for problem-specific error estimators. Our approach is based on a "learn and adapt" framework. In the first step, a low-fidelity model learns to sample points from a large input parametric space, and in the second step, the low-fidelity model improves by adapting to the current ROM approximation and the procedure continues until the low-fidelity model is a good representation of a FOM. By keeping a high-fidelity solution in the loop, not only accuracy and convergence is achieved, the requirement of error estimators or upper bounds is not a pre-requisite. The details are explained in section 4.2. The goal of this chapter is to explore the parametric space efficiently and generate appropriate snapshots for a multi-dimensional parametric system irrespective of the problem definition, the underlying discretization techniques used for solving the problem such as the finite element method (FEM), or finite volume method (FVM), or for cases when posteriori error estimators are unavailable or difficult to obtain as in the case of a complex viscoelastic fluid flow problem.

In fact, we evaluate the max norm error between the high-fidelity solution and reduced-order solution at the computed parametric points until it establishes the acceptable accuracy,

hence our sampling process is adhoc based on heuristics. The greedy selection of points can also be tuned as per the user's requirements for the efficient performance of the algorithm, which will be reflected in the numerical examples. We first introduce the algorithm on a 2D heat conduction problem with 2 input parameters and make a qualitative comparison with the existing greedy RBM. Next, we study a 2D advection-diffusion reaction problem in a parametric space of 9 dimensions and further experiment this methodology to an inertialess flow past a sphere benchmark problem obeying Giesekus constitutive model, which is detailed in chapter 3.

We focus on two main aspects in this chapter: first, is the construction of an initial low-fidelity model explained in section 4.2.1 and the second is the sampling strategy using the DEIM technique explained in 4.2.2. The idea of using DEIM in iterative multi-fidelity modeling is not related to the approximation of the nonlinear term, but only to the notion of greedy selection of "optimal" parametric points.

## 4.2 Iterative multi-fidelity modeling (IMF) for building PODG-ROM

The proposed sampling methodology is unrelated to any particular PDE definition, but in order to compare it to the examples presented in the numerical analysis section 4.3 and 4.4, we adhere to a linear elliptic parameterized PDE. Denoting with  $\boldsymbol{\mu}$  as a set of input parameters with its variation over the parametric domain  $\mathcal{D} \in \mathbb{R}^P, p \geq 1$ ,  $\mathbf{x} \in \Omega$  being the spatial coordinates, the problem is to find solution  $u(\boldsymbol{\mu})$  in some finite-dimensional discrete space  $\mathcal{V}^{\mathcal{N}} \subset \mathcal{V}$  where  $\mathcal{N} = \dim(\mathcal{V}^{\mathcal{N}})$  such that

$$a(u, v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall v \in \mathcal{V}^{\mathcal{N}} \quad (4.1)$$

In this work, we used finite elements to obtain the discrete solution to (4.1), however, the proposed method is also applicable to other numerical discretization solvers. After spatial discretization, the FE solution of the field variable 'u', can be approximated as:

$$u \approx u_h(\mathbf{x}; \boldsymbol{\mu}) = \sum_{i=1}^{\mathcal{N}} N_i(\mathbf{x})(u_h(\boldsymbol{\mu}))_i \quad (4.2)$$

$N_i$  are the shape functions of choice, and  $(u_h(\boldsymbol{\mu}))_i$  represent the scalar values of the field  $u_h$  at discretization points  $x_i \in \mathbb{R}^d, d = 1, 2, 3$  and  $i = 1 \dots \mathcal{N}$ .

Traditionally, if  $\{\mathbf{u}_{\text{HF}}^k(\boldsymbol{\mu})\}_{k=1}^N \in \mathbb{R}^{\mathcal{N}}$  represents the high-fidelity snapshots of the PDE problem 4.1 at distinct parameter values  $\boldsymbol{\mu} \in \bar{\mathcal{E}}_{\text{train}} \subset \mathcal{D}$ , of cardinality  $|\bar{\mathcal{E}}_{\text{train}}| = N$ , that can be suitably well approximated in a low-dimensional manifold, then the solution can be represented in a separated form as,

$$u_{\text{HF}}(\mathbf{x}; \boldsymbol{\mu}) \approx \sum_{i=1}^r \phi_{\text{HF}}^i(\mathbf{x}) \psi_{\text{LF}}^i(\boldsymbol{\mu}) \quad (4.3)$$

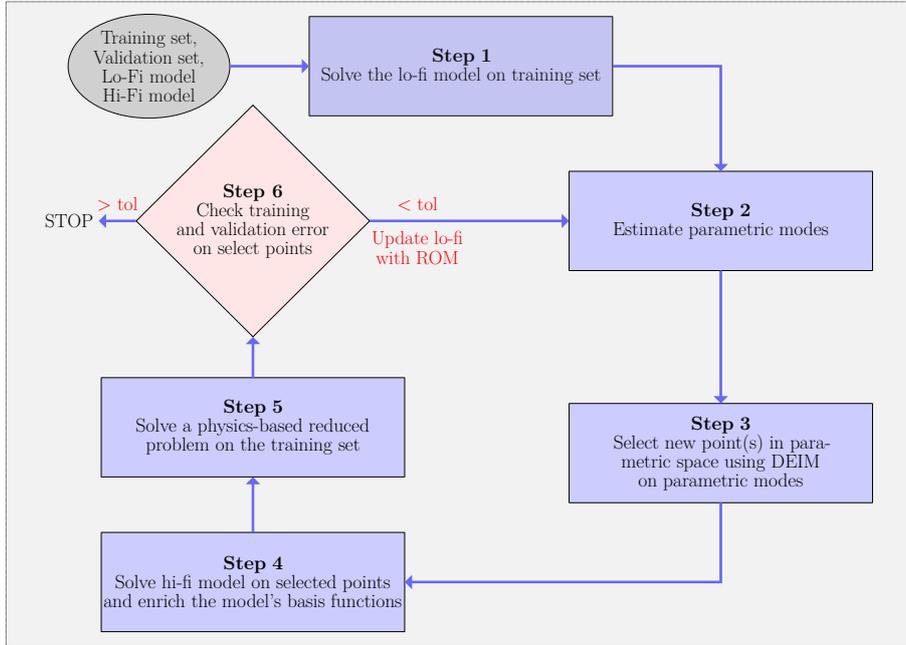
Here,  $\{\phi_{\text{HF}}^i\}_{i=1}^r \in \mathbb{R}^{\mathcal{N}}$  represents the high-fidelity basis functions which spans the low dimensional subspace, i.e.  $\mathcal{V}^R = \text{span}\{\phi_{\text{HF}}^1, \dots, \phi_{\text{HF}}^r\} \subset \mathcal{V}^{\mathcal{N}}$ , and  $\psi_{\text{LF}}(\boldsymbol{\mu}) : \mathcal{D} \rightarrow \mathbb{R}$  are parametric functions that span the parametric space. It is true that high-fidelity models can capture the intricacies of complex PDE systems, but they are also equally expensive to train and the offline cost to recover the basis functions is quite high. If the parametric functions  $\psi_{\text{LF}}^i(\boldsymbol{\mu})$  were previously known, we could easily extract a set of points  $\boldsymbol{\mu}^{\mathbf{P}} \subset \bar{\mathcal{E}}_{\text{train}}$  where  $\mathbf{P} \subset \{1, \dots, N\}$  using any efficient sampling technique. The optimal basis functions might then be recovered by generating high-fidelity snapshots at the computed set of points.

However,  $\psi_{\text{LF}}^i(\boldsymbol{\mu})$  are not known a priori, but we can reasonably assure that if a low-fidelity model is instead used for approximation, then by similar expression as given in eq.(4.3) we have,

$$u_{\text{LF}}(\boldsymbol{x}; \boldsymbol{\mu}) \approx \sum_{i=1}^r \phi_{\text{LF}}^i(\boldsymbol{x}) \psi_{\text{LF}}^i(\boldsymbol{\mu}) \quad (4.4)$$

where  $\psi_{\text{LF}}^i(\boldsymbol{\mu})$  have similar features as the  $\psi_{\text{HF}}^i(\boldsymbol{\mu})$ . Therefore, we can use those to recover the high-fidelity snapshots at computed points  $u_{\text{HF}}(\boldsymbol{\mu}^{\text{P}})$ . The process can be made iterative, as the newly computed  $\boldsymbol{\mu}^{\text{P}}$  can effectively result in the reconstruction of the high-fidelity basis functions. This subsequently leads to the enrichment of the reduced basis subspace, which causes an improvement in the low-fidelity model approximation.

Therefore, step 1 of the proposed method is to obtain a poor or inexpensive approximation to the FOM using a low-fidelity (lo-fi) model,  $f_{\text{LF}}^{\text{init}} : \Omega \times \mathcal{D} \rightarrow \mathbb{R}$  that maps all the parameters belonging to a given training set  $\boldsymbol{\mu} \in \Xi_{\text{train}} \subset \mathcal{D}$  to produce the same output with lower accuracy. The flow is shown in figure 4.2 and details on the construction of the initial low-fidelity model are explained in section 4.2.1.



**Figure 4.2:** Flow of the iterative multi-fidelity modeling approach.

In step 2, we compute parametric functions  $\{\psi_{\text{LF}}^i\}_{i=1}^r \in \mathbb{R}^N$  from the low-fidelity model approximation by primarily using POD, and then in step 3, we recover "optimal sampling points" using interpolation strategy over these parametric functions to sample optimal points in a greedy procedure,  $\boldsymbol{\mu}^{\text{P}} \subset \Xi_{\text{train}}$ . The details are discussed in section 4.2.2.

Next, in step 4, using a high-fidelity (hi-fi) model we generate snapshots on the select sample points  $\boldsymbol{\mu}^{\text{P}}$  to recover the high-fidelity basis functions  $\phi_{\text{HF}}^i \in \mathcal{V}^R$  and therefore, enrich the reduced basis subspace,  $\mathcal{V}^R = \text{span}(\{\phi_{\text{HF}}^i\}_{i=1}^{r''}) \in \mathbb{R}^N$  where  $r'' \leq \#(\boldsymbol{\mu}^{\text{P}})$ , the construction is properly explained in the section 4.2.3. A high-fidelity model is mathematically defined as  $f_{\text{HF}} : \Omega \times \mathcal{D} \rightarrow \mathbb{R}$  that maps all the selected points  $\boldsymbol{\mu}^{\text{P}} \subset \Xi_{\text{train}}$  to estimate the output with the accuracy that is needed for the task.

In step 5, we then solve a POD-G ROM for all  $\boldsymbol{\mu} \in \Xi_{\text{train}}$ . The functions  $\psi_{\text{LF}}^i$  obtained during the first approximation are likely to not generate exactly the same space as the

$\psi_{\text{HF}}^i$  i.e.  $\text{span}(\psi_{\text{LF}}^i) \neq \text{span}(\psi_{\text{HF}}^i)$ . Therefore, recovery of the high-fidelity basis functions  $\phi_{\text{HF}}^i$  may not be accurate, and reliable to represent the large-scale PDE system, hence the procedure has to undergo certain iterations.

Finally, in step 6, we evaluate the error between the high-fidelity model and reduced basis model approximation at the computed discrete points using the error metrics that are discussed in 4.2.5. If it is below a certain prescribed tolerance level, we terminate the algorithm, else we adapt the low-fidelity model with the current ROM approximation (refer section 4.2.4) and repeat the procedure until the ROM constructed represents the FOM adequately. To measure the overall performance of the algorithm, we also check for validation error by computing the error on another set of parameters belonging to a given validation set  $\mathcal{E}_{\text{val}} \subset \mathcal{D}$ .

The sample points that are obtained provide a locally optimal choice at each stage of the iterative cycle, however as iteration continues and new points are added in each iteration the algorithm converges towards the global solution with certain accuracy in very reduced complexity.

### 4.2.1 Construction of the initial low-fidelity model

We propose two types of sketch models for the construction of an initial low-fidelity model, depending on the availability of either of the two conditions:

1. No database of high-fidelity solutions is present a priori,
2. A database of high-fidelity solutions is available

**Coarse sketch model:** When no database of solutions is present a priori, the initial low-fidelity model is built using a derefined version of a high-fidelity model, which is nothing but a coarse finite element model. We refer to it as a "coarse sketch model" for future discussions. Snapshots generated using this coarse sketch appear to be a wide matrix, denoted by  $\mathbf{S}_{\text{LF}} = [\mathbf{u}_{\text{LF}}^1, \mathbf{u}_{\text{LF}}^2, \dots, \mathbf{u}_{\text{LF}}^N] \in \mathbb{R}^{m \times N}$ , where  $\mathbf{u}_{\text{LF}} = \mathbf{u}_h(\boldsymbol{\mu})$  at discretization points  $\mathbf{x} \in \mathbb{R}^m$ , such that  $m \ll \mathcal{N}$ .

**POD-G ROM:** Now, let us assume the case when we already have some solutions of the large-scale PDE system available to us, be it experimental or numerical data. This is materialized in our work by solving a high-fidelity model for any random training parameters,  $\mathbf{X} = [\mathbf{u}_{\text{HF}}^1(\boldsymbol{\mu}^k), \mathbf{u}_{\text{HF}}^2(\boldsymbol{\mu}^k), \dots, \mathbf{u}_{\text{HF}}^K(\boldsymbol{\mu}^k)] \in \mathbb{R}^{\mathcal{N} \times K}$  with  $\mathcal{N} \gg K$ , and  $k \subset \{1, \dots, N\}$  and term it here as a **random sketch model**. Then the initial low-fidelity model is a ROM approximation, which is constructed by Galerkin projection of the PDE system onto the reduced basis functions computed from this sketch model.

**Remark:** Although the selection parameters for a random sketch model are entirely user-dependent, it is advised to start by building the reduced bases with just a few snapshots. For the primary purpose of lowering the offline cost of MOR, the low-fidelity model approximation from such a random sketch must remain a less expensive approximation to the FOM at the initial stage of the method. Then, as the iteration advances, the quality of the low-fidelity model improves and converges to the FOM accurately.

### 4.2.2 Parametric point selection

As previously mentioned, sampling points are extracted from the parametric modes of the low-fidelity model approximation. This is based on the heuristic assumptions that

the low-fidelity model, although a poor approximation to the high-fidelity model, may nonetheless accurately reflect the essential features of the high-fidelity model's parametric dependence. A brief description of how the parametric points are sampled from a given training set,  $\boldsymbol{\mu}^P \subset \Xi_{train}$  using DEIM is explained in the algorithm 2. DEIM finds the sample points in a greedy way from an input basis which is given here by the parametric functions computed by performing SVD on the low-fidelity model approximation,

$$\mathbf{S}_{LF} = \boldsymbol{\Phi}_{LF} \boldsymbol{\Sigma}_{LF} \boldsymbol{\Psi}_{LF}^T \quad (4.5)$$

where,  $\boldsymbol{\Phi}_{LF} = [\phi_{LF}^1, \phi_{LF}^2, \dots, \phi_{LF}^r] \in \mathbb{R}^{m \times r}$  with  $m \ll \mathcal{N}$ , and  $r \leq N$  are poorly approximated POD modes while  $\boldsymbol{\Psi}_{LF} = [\psi_{LF}^1, \psi_{LF}^2, \dots, \psi_{LF}^r] \in \mathbb{R}^{N \times r}$  denotes the parametric modes that span the parametric subspace. The rectangular diagonal matrix  $\boldsymbol{\Sigma}_{LF} \in \mathbb{R}^{r \times r}$  contains the corresponding non-negative singular values,  $\sigma_1 \geq \dots \geq \sigma_r \geq 0$  accounting to the information content of the low-fidelity model solution data. The process of sampling starts by selecting the index with the largest magnitude, corresponding to the first entry of the input basis  $\{\psi_{LF}^i\}_{i=1}^r$ . The remaining points are selected by finding the location at which the residual of the current approximation is maximum (refer to the algorithm 2). The points that are computed are unique due to the linear independence of the input basis, which guarantees that the indices are hierarchical and non-repetitive in nature.

---

**Algorithm 2** DEIM sampling adopted from [108]

---

- 1: INPUT: Parametric functions  $\boldsymbol{\Psi}_{LF} \in \mathbb{R}^{N \times r}$
  - 2: OUTPUT: Sampled parametric points  $\boldsymbol{\mu}^P = [\boldsymbol{\mu}^{(1)}, \dots, \boldsymbol{\mu}^{(r)}]^T \subset \Xi_{train}$
  - 3:  $\boldsymbol{\mu}^{(1)} = \max \left\{ \left| \psi_{LF(\cdot, 1)} \right| \right\}$
  - 4:  $\boldsymbol{\mu}^P = [\boldsymbol{\mu}^{(1)}]$
  - 5: **for**  $l = 2 : r$  **do**
  - 6:     Solve  $c = [\boldsymbol{\Psi}_{LF(1:l-1, 1:l-1)}]^{-1} \psi_{LF(1:l-1, l)}$
  - 7:      $\mathbf{r} = \psi_{LF(\cdot, l)} - \boldsymbol{\Psi}_{LF} c$
  - 8:      $\boldsymbol{\mu}^{(l)} = \max \{ |\mathbf{r}| \}$
  - 9:      $\boldsymbol{\mu}^P = [\boldsymbol{\mu}^P, \boldsymbol{\mu}^{(l)}]^T$
  - 10: **end for**
- 

This sampling procedure is resumed after every iteration of the proposed algorithm and is not restarted from the beginning. As a result, instead of oversampling the same points from the training set, we are able to sample distinct points from it. The parametric functions obtained in  $i^{th}$  iteration of the multi-fidelity algorithm are orthogonalized with respect to the parametric functions obtained in  $(i-1)^{th}$  iteration through Gram-Schmidt orthonormalization in order to prevent repetition and picking up points closer to previously calculated points. This step is crucial for finding the best and most distinctive points throughout each iteration cycle, enabling us to explore the parametric space more thoroughly.

**Remark:** A general note to take into account while selecting parametric points is that it's always a better choice to pick from the first 'r' truncated parametric functions due to its content of the highest energy or information of the system, arranged in descending order.

It is noteworthy that the proposed methodology doesn't necessarily perform like a classical greedy sampling procedure due to the nature of the selection of points from the parametric functions. Hence, depending on the available computing resources, the level of "greediness" can be fine-tuned. In other words, since DEIM will generate the same number

of points as the rank of parametric functions, the user can decide to select all the sample points at once, or also has the option to select one parametric point per iteration. This is one of the advantages of this approach, where the selection of points per iteration is completely user-dependent, which can be assimilated for parallel computations. A general remark has to be made, incorporating such a step can also lead to the selection of excess sample points than required which deviates from the main objective of selecting a few optimal points and hence needs to be taken care of.

### 4.2.3 Recovery of the reduced basis functions

The high-fidelity basis function in the first iteration of the proposed method is recovered by performing SVD on the select snapshots,  $\mathbf{S}_{\text{HF}} = \{\mathbf{u}_{\text{HF}}(\boldsymbol{\mu}^P)\}$  for all  $\boldsymbol{\mu}^P \in \Xi_{\text{train}}$ .

$$\text{svd}(\mathbf{S}_{\text{HF}}) = \boldsymbol{\Phi}_{\text{HF}} \boldsymbol{\Sigma}_{\text{HF}} \boldsymbol{\Psi}_{\text{HF}}^{\text{T}} \quad (4.6)$$

where,  $\boldsymbol{\Phi}_{\text{HF}} = [\boldsymbol{\phi}_{\text{HF}}^1, \boldsymbol{\phi}_{\text{HF}}^2, \dots, \boldsymbol{\phi}_{\text{HF}}^{r''}] \in \mathbb{R}^{\mathcal{N} \times r''}$  with  $r'' \leq \#(\boldsymbol{\mu}^P)$  contains the high-fidelity reduced bases that span the low-dimensional subspace  $\mathcal{V}^R$  and  $\boldsymbol{\Psi}_{\text{HF}} = [\boldsymbol{\psi}_{\text{HF}}^1, \boldsymbol{\psi}_{\text{HF}}^2, \dots, \boldsymbol{\psi}_{\text{HF}}^{r''}] \in \mathbb{R}^{\mathcal{N} \times r''}$  denotes the parametric modes. Similarly, the rectangular diagonal matrix  $\boldsymbol{\Sigma}_{\text{HF}} \in \mathbb{R}^{r'' \times r''}$  contains the corresponding non-negative singular values,  $\sigma_1 \geq \dots \geq \sigma_{r''} \geq 0$  accounting for the information content of the high-fidelity model solution data. At  $(i+1)^{\text{th}}$  iteration of the algorithm, the reduced subspace  $\mathcal{V}^R$  is updated through the Gram-Schmidt procedure (refer algorithm 3).

---

**Algorithm 3** Gram-Schmidt orthonormalization at the  $(i+1)^{\text{th}}$  iteration of the proposed method

---

```

1: for  $l = 1 : \dim(\mathbf{S}_{\text{HF}})$  do
2:    $\boldsymbol{\phi}_{\text{HF}}^l = \mathbf{S}_{\text{HF}}^l - \boldsymbol{\Phi}_{\text{HF}}^i \langle \boldsymbol{\Phi}_{\text{HF}}^i, \mathbf{S}_{\text{HF}}^l \rangle$ 
3:   if  $\frac{\|\boldsymbol{\phi}_{\text{HF}}^l\|}{\|\mathbf{S}_{\text{HF}}^l\|} > \epsilon_g$  then
4:      $\boldsymbol{\Phi}_{\text{HF}}^{i+1} = \boldsymbol{\Phi}_{\text{HF}}^i \oplus \frac{\boldsymbol{\phi}_{\text{HF}}^l}{\|\boldsymbol{\phi}_{\text{HF}}^l\|}$ 
5:   end if
6:   Update,  $\boldsymbol{\Phi}_{\text{HF}}^i = \boldsymbol{\Phi}_{\text{HF}}^{i+1}$ 
7: end for

```

---

### 4.2.4 Updating low-fidelity model

In this part, we demonstrate how the current POD-G ROM approximation can be used to update the low-fidelity model for each iteration of the algorithm until convergence, i.e. we approximate the solution  $u_{\text{LF}} : \Omega \times \mathcal{D} \rightarrow \mathbb{R}$  with a function  $u^R \in \mathcal{V}^R$  defined by,

$$u^R(\mathbf{x}; \boldsymbol{\mu}) = \sum_{i=1}^{r''} \boldsymbol{\phi}_{\text{HF}}^i(b(\boldsymbol{\mu}))_i = \boldsymbol{\Phi}_{\text{HF}} \mathbf{b}(\boldsymbol{\mu}) \quad (4.7)$$

where, the POD expansion coefficients  $\mathbf{b}(\boldsymbol{\mu}) = (b_1, b_2, \dots, b_{r''})^{\text{T}}$  can be calculated by Galerkin projection of the PDE system onto the basis functions  $\boldsymbol{\phi}_{\text{HF}}^i$ . The initial low-fidelity snapshots data is now updated with the current reduced solution, such that  $\mathbf{S}_{\text{LF}} = [u_1^R, u_2^R, \dots, u_N^R] \in \mathbb{R}^{\mathcal{N} \times r''}$ .

**Remark:** To improve the efficiency of the method, one can also approximate the low-fidelity data with the coefficients of the POD expansion instead of the reduced solution itself and replace  $\mathbf{S}_{\text{LF}}$  with  $\mathbf{B}$  where  $\mathbf{B} = [\mathbf{b}^1, \mathbf{b}^2, \dots, \mathbf{b}^N] \in \mathbb{R}^{r'' \times N}$ . Here  $\mathbf{b}^i = (b_1, b_2, \dots, b_{r''})^{\text{T}}$

represents the POD coefficients from eq. (4.7). This process can reduce the cost of exploration of the parametric space using low-fidelity approximation from  $\mathcal{O}(\mathcal{N})$  to  $\mathcal{O}(r'')$ .

By performing SVD on  $\mathbf{B}$  we have,

$$\mathbf{B} = \boldsymbol{\varphi} \boldsymbol{\varsigma} \hat{\boldsymbol{\psi}}^T \quad (4.8)$$

Rewriting eq. (4.7) we have,

$$\tilde{u}^R(\mathbf{x}; \boldsymbol{\mu}) \approx u^R(\mathbf{x}; \boldsymbol{\mu}) = \underbrace{\boldsymbol{\Phi}_{\text{HF}}}_{\tilde{\boldsymbol{\Phi}}_{\text{HF}}} \boldsymbol{\varphi} \boldsymbol{\varsigma} \hat{\boldsymbol{\psi}}^T \quad (4.9)$$

The original high-fidelity basis functions can now be replaced by the approximate left singular vectors by the expression shown in eq. (4.9). This scaling factor is committed to improving the accuracy of the basis functions that could be lost if reduced coefficients are used in place of reduced-order solutions for the parametric exploration.

The next section is dedicated to a discussion on error metrics which are used to test the reliability of the approach.

#### 4.2.5 Error metrics

If  $u_{\text{HF}}(\mathbf{x}; \boldsymbol{\mu}^P)$  and  $u^R(\mathbf{x}; \boldsymbol{\mu}^P)$  represents the FOM and ROM solution respectively at the computed parametric points  $\boldsymbol{\mu}^P$  from the training set  $\Xi_{\text{train}} \subset \mathcal{D}$ , then the max norm of the relative error is estimated at the sample points such that

$$\epsilon_{\text{train}} = \max_{\boldsymbol{\mu}^P} \sqrt{\frac{\sum_{i=1}^{\mathcal{N}} \|u_{\text{HF}}(x_i; \boldsymbol{\mu}^P) - u^R(x_i; \boldsymbol{\mu}^P)\|_2^2}{\sum_{i=1}^{\mathcal{N}} \|u_{\text{HF}}(x_i; \boldsymbol{\mu}^P)\|_2^2}} \quad (4.10)$$

In order to better understand the quality of the reduced model, the algorithm is validated on another set of parameters,  $\Xi_{\text{val}} \subset \mathcal{D}$  for the problem defined. By validating on a different set of points, if the error estimation between the FOM and ROM solutions is reduced as the cycle increases, it can be inferred that ROM well approximates the large-scale system for any  $\boldsymbol{\mu} \in \mathcal{D}$ . If the max norm error for both the training and validation set is below a certain tolerance limit, the sketch model constructed can be considered to be reliable.

$$\epsilon_{\text{val}} = \max_{\boldsymbol{\mu}} \sqrt{\frac{\sum_{i=1}^{\mathcal{N}} \|u_{\text{HF}}(x_i; \boldsymbol{\mu}) - u^R(x_i; \boldsymbol{\mu})\|_2^2}{\sum_{i=1}^{\mathcal{N}} \|u_{\text{HF}}(x_i; \boldsymbol{\mu})\|_2^2}} \quad \forall \boldsymbol{\mu} \in \Xi_{\text{val}} \quad (4.11)$$

The error between all the snapshots and the ROM solutions obtained by the iterative multi-fidelity approach is evaluated and can be viewed as a benchmark for the ROM error, provided sufficient snapshots are generated:

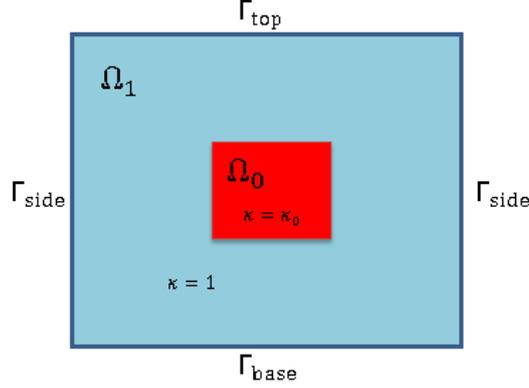
$$\epsilon_{\text{ROM}} = \sqrt{\sum_{i=1}^{\mathcal{N}} \frac{\|u_{\text{HF}}(\mathbf{x}; \boldsymbol{\mu}^{(i)}) - u^R(\mathbf{x}; \boldsymbol{\mu}^{(i)})\|_2^2}{\|u_{\text{HF}}(\mathbf{x}; \boldsymbol{\mu}^{(i)})\|_2^2}} \quad \boldsymbol{\mu} \in \Xi_{\text{train}} \quad (4.12)$$

We also analyze POD projection error, which is given by the error between the snapshots and their projection onto the recovered basis functions:

$$\epsilon_{\text{POD}} = \sqrt{\sum_{i=1}^{\mathcal{N}} \frac{\|u_{\text{HF}}(\mathbf{x}; \boldsymbol{\mu}^{(i)}) - \Pi u_{\text{HF}}(\mathbf{x}; \boldsymbol{\mu}^{(i)})\|_2^2}{\|u_{\text{HF}}(\mathbf{x}; \boldsymbol{\mu}^{(i)})\|_2^2}} \quad \boldsymbol{\mu} \in \Xi_{\text{train}} \quad (4.13)$$

### 4.3 Numerical example of a heat conduction problem

In order to assess the proposed methodology, we first begin by analyzing a simple steady-state heat conduction problem in a 2D domain,  $\Omega = (0, 1) \times (0, 1)$  as shown in figure 4.3. This problem is reproduced from ([9]) where it is solved using classical greedy RBM. The boundary of the domain is split into three parts, the base, the top, and the sides, and  $\Omega_0$  is a square block placed in the center of the domain. Let  $\kappa$  be the thermal conductivity with  $\kappa|_{\Omega_0} = \mu_{[1]}$  and  $\kappa|_{\Omega_1} = 1$  where  $\Omega_1 = \Omega \setminus \Omega_0$ .



**Figure 4.3:** Geometry of heat conduction problem

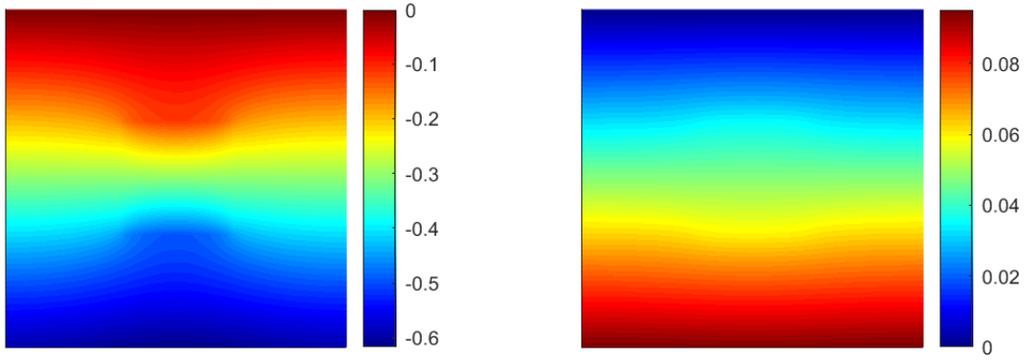
Two input parameters are considered for this problem  $\boldsymbol{\mu} = [\mu_{[1]}, \mu_{[2]}]$ , where  $\mu_{[1]}$  is the conductivity in  $\Omega_0$  the region, and the second parameter  $\mu_{[2]}$  is the constant heat flux over the bottom boundary. The strong formulation for this parameterized problem is governed by Poisson's equation. For some parameter value  $\boldsymbol{\mu} \in \mathcal{D}$ , find  $u(\boldsymbol{\mu})$  such that

$$\begin{aligned} \nabla \cdot \kappa_{\boldsymbol{\mu}} \nabla u(\boldsymbol{\mu}) &= 0 && \text{in } \Omega, \\ u(\boldsymbol{\mu}) &= 0 && \text{on } \Gamma_{\text{top}} \\ \kappa_{\boldsymbol{\mu}} \nabla u(\boldsymbol{\mu}) \cdot \boldsymbol{n} &= 0 && \text{on } \Gamma_{\text{side}} \\ \kappa_{\boldsymbol{\mu}} \nabla u(\boldsymbol{\mu}) \cdot \boldsymbol{n} &= \mu_{[2]} && \text{on } \Gamma_{\text{base}} \end{aligned} \quad (4.14)$$

Here,  $u(\boldsymbol{\mu})$  is the scalar temperature field variable, and  $\kappa_{\boldsymbol{\mu}}$  is given such that  $\kappa_{\boldsymbol{\mu}} = \varphi_1 + \mu_{[1]}\varphi_0$ , where  $\varphi$  is the characteristic function with subscript donating the corresponding domain. Defining  $\mathcal{V}^{\mathcal{N}} = \{v \in H_0^1(\Omega) | v|_{\Gamma_{\text{top}}} = 0\}$ , the weak parametrized formulation then reads: for some parameter  $\boldsymbol{\mu} \in \mathcal{D}$ , find  $u(\boldsymbol{\mu}) \in \mathcal{V}^{\mathcal{N}}$  such that,

$$\begin{aligned} a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) &= f(v; \boldsymbol{\mu}) \quad \forall v \in \mathcal{V}^{\mathcal{N}}, \\ a(w, v; \boldsymbol{\mu}) &= \int_{\Omega} \kappa_{\boldsymbol{\mu}} \nabla w \cdot \nabla v \text{ and } f(v; \boldsymbol{\mu}) = \mu_{[2]} \int_{\Gamma_{\text{base}}} v, \end{aligned} \quad (4.15)$$

for all  $v, w \in \mathcal{V}^{\mathcal{N}}$ . The selected range for the parameters is  $\boldsymbol{\mu} = [\mu_{[1]}, \mu_{[2]}] \in \mathcal{D} = [0.1, 10] \times [-1, 1]$ . A total of, 2050 sample points are generated in which the training set  $\Xi_{\text{train}}$  comprises of 2000 points and the validation set  $\Xi_{\text{val}}$  consists of 50 points. For  $\mu_{[1]}$ , the points are generated using uniform discretization whereas for the second input parameter  $\mu_{[2]}$ , the points are generated using log space. The graphical representation of the temperature field for two different sets of parameters is shown in figure 4.4.



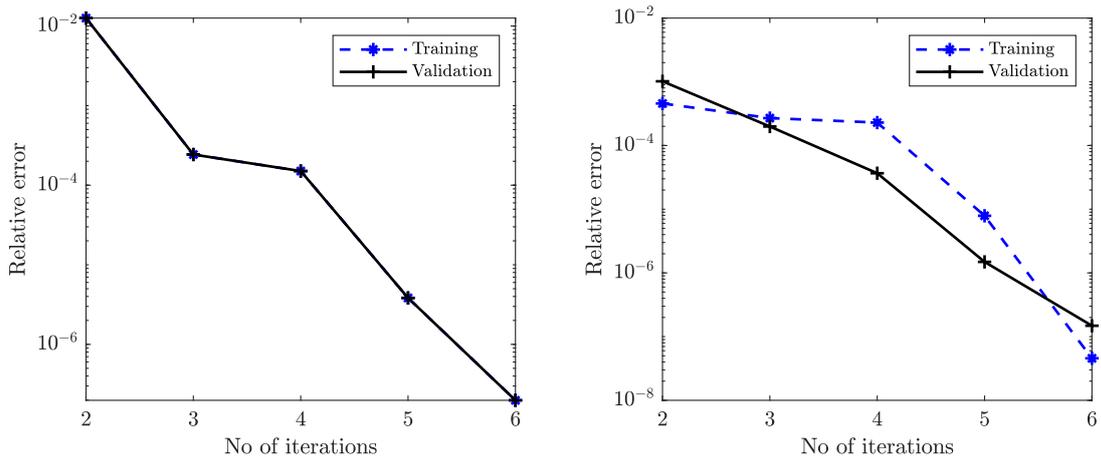
(a)  $\mu_{[1]} = 0.1, \mu_{[2]} = -1$

(b)  $\mu_{[1]} = 0.4, \mu_{[2]} = 0.5$

**Figure 4.4:** Two different representative solutions for the parameterized conductivity problem.

### 4.3.1 Results and discussions

We now perform the proposed algorithm based on two different initialization of the low-fidelity model. The first analysis is based on the assumption that we already have some random dataset of solutions of a FOM. Then the initial low-fidelity model is a ROM constructed from this random sketch model. In this example, we initialize the random sketch with 2 linearly independent snapshots from the training set. Since this is a low-rank linear problem, we chose to select one sample point per iteration. With a target tolerance set to  $\epsilon = 10^{-6}$ , 6 iteration cycles are required to achieve the desired accuracy, as shown in figure 4.5a. Also, we show the convergence plot for parameters belonging to the validation set, which decays smoothly until the target accuracy is achieved. This implies, that the quality of ROM constructed with the proposed iterative multi-fidelity approach represents well the large-scale PDE system for any parameter belonging to the parametric space  $\mathcal{D}$ .



(a) Initial lo-fi model with a POD-G ROM

(b) Initial lo-fi model with a coarse sketch model

**Figure 4.5:** Error  $\epsilon_{train}$  and  $\epsilon_{val}$  between FOM and reconstructed ROM solution

Figure 4.6 illustrates the parametric functions generated from low-fidelity model approximation during each iteration of the proposed algorithm, along with the computed DEIM points. The initial rank of the random sketch is 2, so in iteration 1 of the iterative

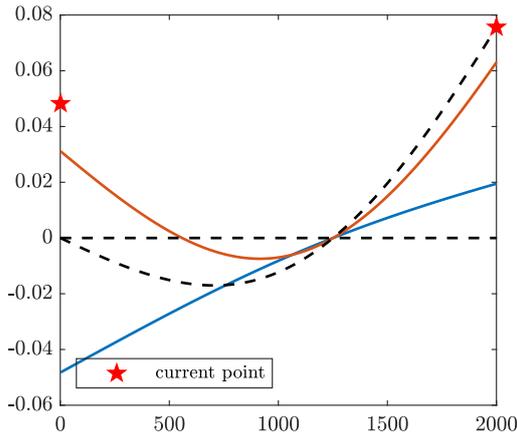
multi-fidelity algorithm, we have two parametric functions, as one can see in 4.6a. In the rest of the plots 4.6b-4.6f, we show a single parametric function and its corresponding DEIM point (since we pick only one point per iteration), instead of showing all the modes generated during each iteration of the DEIM algorithm. The figures reflect the unique sampling points obtained in every iteration, which is expected for preserving the non-redundancy of the proposed algorithm.

The second analysis is for the cases when no database of solutions is available a priori, we construct the initial low-fidelity model from a coarse sketch model as shown in figure 4.7a. Figure 4.7b represents a fine grid model used for generating the high-fidelity solution. We observe that the same number of iterations are required as in the previous case to achieve accuracy of  $\mathcal{O}(10^{-6})$  for parameters belonging to both the training and validation set, shown in figure 4.5b. This specific example has only a parametric dimension of 2 and also this problem is not mesh-dependent, so both the sketch models consisted of sampling the same number of points. However, we will observe that the selection of points is not consistent in a complex problem as the advection-diffusion problem in 9 dimensions discussed in section 4.4 and more or fewer sampling points may be required depending on the initialization of the low-fidelity model to retain the same target accuracy.

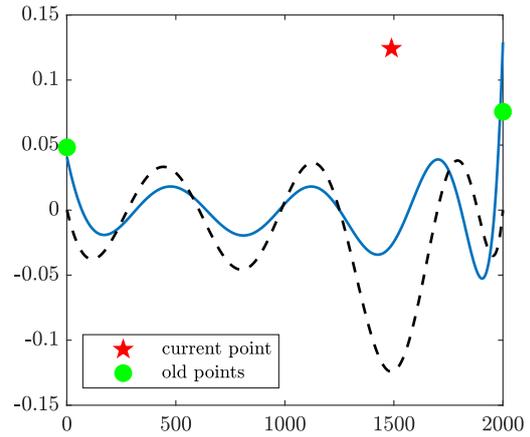
The parametric points sampled using both the sketch models are represented in figure 4.8a. We can notice that irrespective of the sketch model chosen to construct the low-fidelity model, the proposed method in this example extracted almost exact points in both cases. We also show a plot of the Gaussian distribution of the sampled points conducted over a trial of 10 experiments using different random sketch models in figure 4.8b. In other words, each experiment is initialized by different random snapshots without any repetition. The figure reflects a very interesting behavior, showing that six out of seven sampled points are nearly identical in every trial except for one point that may have a larger variance than the other six, which was noticed when the third point was sampled.

We also plot the  $\ell_2$  norm POD projection error ( $\epsilon_{\text{POD}}$ ) and ROM error ( $\epsilon_{\text{ROM}}$ ) against the rank of the low-fidelity model using both the sketch models in figure 4.9a, as per the definitions in section 4.2.5. We can observe both the POD and ROM error decays as the rank of the system is increased. Also, the POD error curve is observed to be lower than the ROM error as expected, which is represented by the decay of singular values. Now we make a comparison by solving the same problem using the greedy RBM algorithm, the lower bound is computed using the min theta approach [9]. The convergence plot for the relative  $\ell_2$ -norm ROM error ( $\epsilon_{\text{ROM}}$ ) for all the parameters in the training set and the error bound is shown in figure 4.9a. It is observed that the ROM error has a smooth exponential decay and is lower than the max norm error as expected.

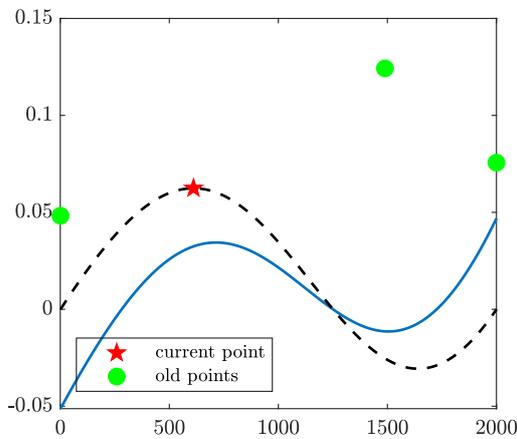
For the target accuracy set to  $\mathcal{O}(10^{-6})$ , the rank of the low-fidelity model achieved is 7, implying that 7 parametric points were sampled to recover the basis functions similar to the findings of our proposed method. Thus, it is evident that the qualitative performance of the proposed method is comparable with the greedy RBM. We also show an illustration of sampled points obtained by greedy RBM in figure 4.9b. We notice that the parametric points picked in the proposed method are not completely comparable to greedy RBM, however, the sampled points in both cases follow a logarithmic trend which could be due to the reason how we discretize our training set, where we chose to discretize  $\boldsymbol{\mu}_{[2]}$  using a log space and a uniform discretization for  $\boldsymbol{\mu}_{[1]}$ . If we apply DEIM on the parametric functions obtained by POD on high-fidelity snapshots for all  $\boldsymbol{\mu} \in \Xi_{\text{train}}$  and plot the sampled points in the same figure 4.9b, we observe that the selection of points is comparable with the proposed method and lie on the same logarithmic trend.



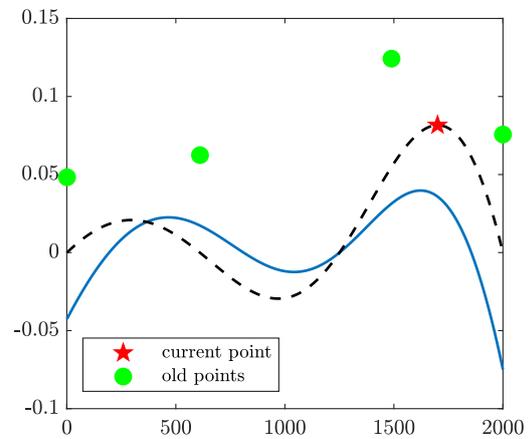
(a) Iteration 1



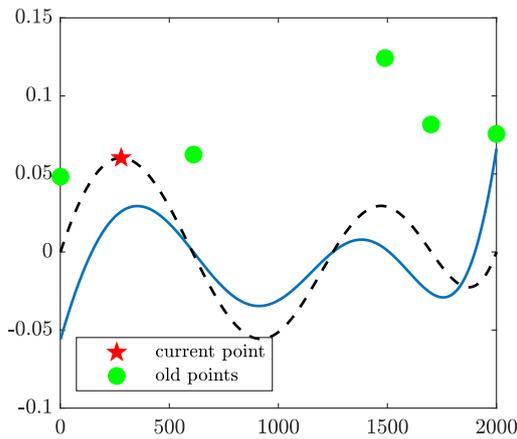
(b) Iteration 2



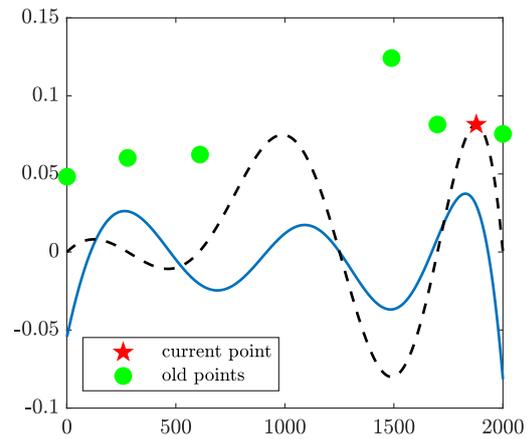
(c) Iteration 3



(d) Iteration 4



(e) Iteration 5

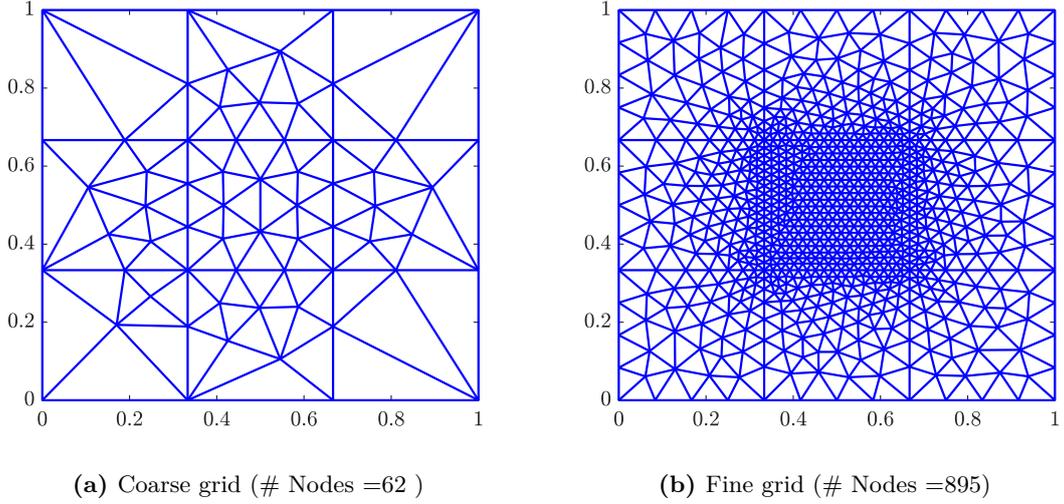


(f) Iteration 6

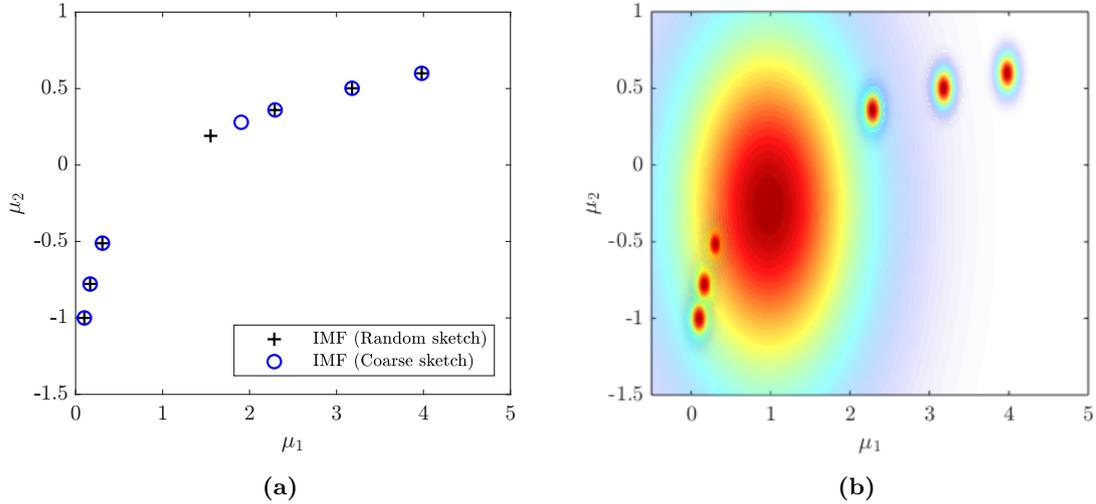
**Figure 4.6:** Parametric modes represented by blue and red curves along with its residual represented by black dotted lines, and DEIM points obtained during each loop of the proposed method.

#### 4.4 Numerical example of an advection-diffusion problem

In this section, we study a 2D advection-diffusion problem with a source term for a 9 parametric dimension. As it can be seen in figure 4.10, the domain is divided into 9



**Figure 4.7:** (a) a coarse discretized mesh for an initial low-fidelity model, and (b) a fine discretized mesh for high-fidelity model approximations.



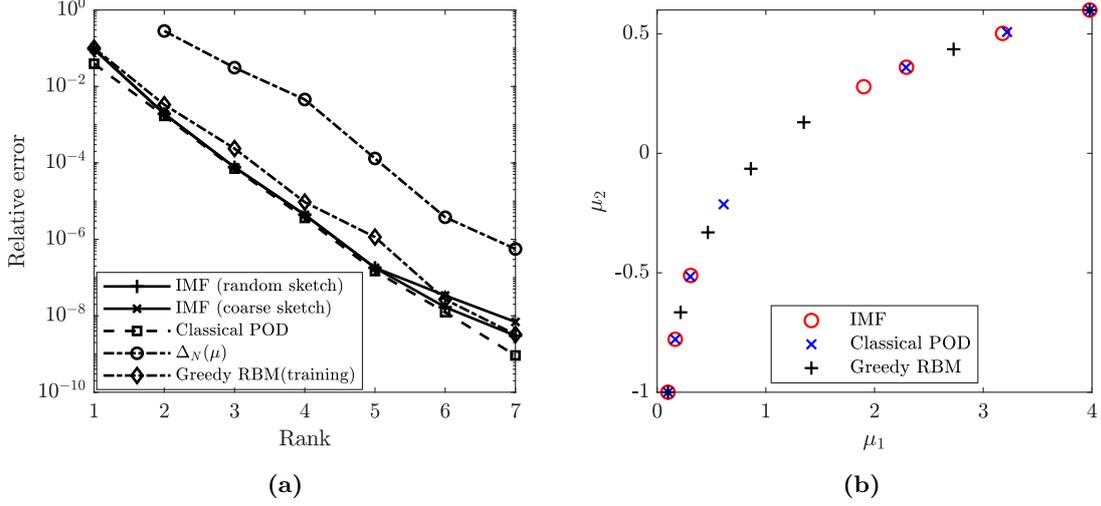
**Figure 4.8:** Sampled parametric points using two sketch models (a), Gaussian distribution of sampling points over 10 trials (b).

subdomains where each region has a different diffusivity coefficient which serves as the input parameter.

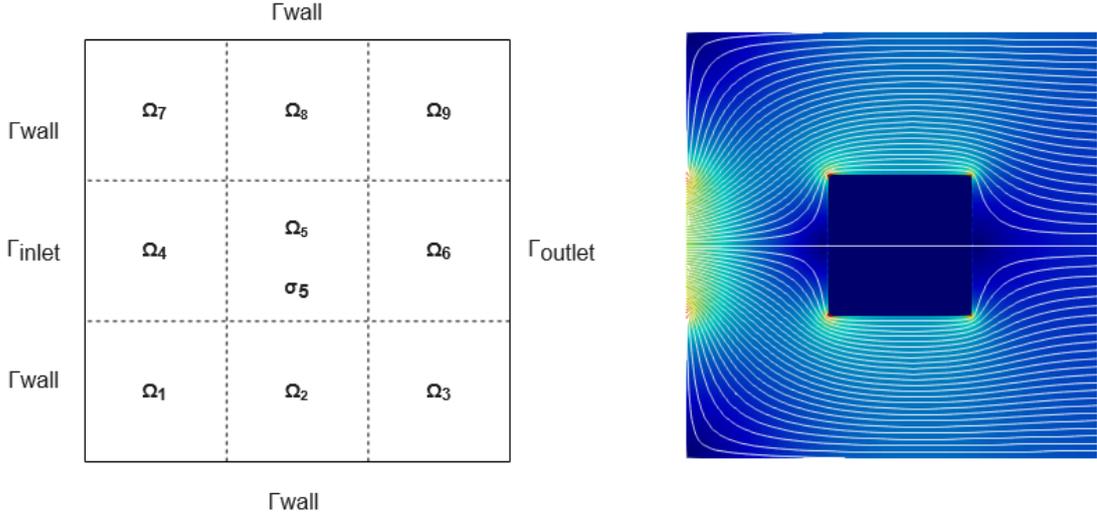
#### 4.4.1 Problem setting

The strong form of the parameterized advection-diffusion reaction equation is governed by the elliptic PDE. For some parameter value,  $\boldsymbol{\mu} \in \mathcal{D}$  find  $u(\boldsymbol{\mu})$  such that:

$$\begin{aligned}
 \mathbf{b} \cdot \nabla u(\boldsymbol{\mu}) - \mathbb{K}(\mu_i) \Delta u(\boldsymbol{\mu}) + \sigma &= 0 & \forall \Omega_i \quad i = 1, \dots, 9 \\
 u(\boldsymbol{\mu}) &= 0 & \text{at } \Gamma_{inlet} \\
 \nabla u(\boldsymbol{\mu}) \cdot \hat{\mathbf{n}} &= 0 & \text{at } \Gamma_{walls} \\
 \sigma &= 1 & \text{at } \Omega_5
 \end{aligned} \tag{4.16}$$



**Figure 4.9:** Error  $\epsilon_{\text{POD}}$  of the classical POD projection, posteriori error bound by greedy RBM and error  $\epsilon_{\text{ROM}}$  of the solution of the reduced order model obtained by the proposed method (a), and sampled parametric points using the proposed method, classical POD and greedy RBM (b).



**Figure 4.10:** (left) Geometrical set up of advection-diffusion problem in a 9 block system and (right) advective flow field.

where,  $u(\boldsymbol{\mu})$  is the unknown field for a spatial domain  $\Omega = [0, 1] \times [0, 1]$ . The diffusion coefficient is given by  $\mathbb{K}(\boldsymbol{\mu}) = \boldsymbol{\mu}$  for the input parameter  $\boldsymbol{\mu} \in \mathcal{D} = [0.01, 10]^9$  and  $\sigma$  is the constant reaction term.  $\mathbf{b}$  is a given advected flow field which is obtained by solving a potential flow problem in the same bounded domain such that  $\nabla \cdot \mathbf{b} = 0$ . We consider a low permeability zone in  $\Omega_5$ , and relatively higher permeability over the rest of the domains, and with a constant velocity at the inlet, we obtain the streamlines as shown in figure 4.10.

The weak parameterized formulation using SUPG then reads as: for some parameter  $\boldsymbol{\mu} \in \mathcal{D}$ , find  $u(\mathbf{x}; \boldsymbol{\mu}) \in \mathcal{V}^{\mathcal{N}}$  where  $\mathcal{V}^{\mathcal{N}} = \{v \in (H_0^1(\Omega))^2 | v|_{\Gamma_{\text{inlet}}} = 0\}$ ,

$$\langle v + \tau \nabla v, \mathbf{b} \cdot \nabla u \rangle + \langle \nabla v, \mathbb{K}(\boldsymbol{\mu}) \nabla u \rangle - \langle \tau \nabla v, \mathbb{K}(\boldsymbol{\mu}) \Delta u \rangle + \langle v + \tau \nabla v, \sigma \rangle = 0 \quad \forall v \in \mathcal{V}^{\mathcal{N}} \quad (4.17)$$

Here  $\tau = \frac{\beta h}{2\|\mathbf{b}\|}$ , where  $h$  is the element size in the direction of the advected velocity and  $\|\mathbf{b}\|$  is the (local) characteristic velocity. The parameter  $\beta$  depends on the Péclet number that varies between (0,1). Note that the third term in the equation (4.17) is zero if linear elements are used for domain discretization [148].

The reduced weak form of eq. (4.17) can be obtained by replacing  $u$  by,  $u_{\text{POD}}$  which can be expressed as:

$$u_{\text{POD}}(\mathbf{x}; \boldsymbol{\mu}) = \sum_{j=1}^r \phi^j(\mathbf{x}) z_j(\boldsymbol{\mu}) \quad (4.18)$$

where,  $\mathbf{z} = \{z_1, \dots, z_r\}^T$  represents the coefficients of the POD expansion and the reduced basis function  $\{\phi^j\}_{j=1}^r$  is obtained by,

$$\phi^j = \text{POD}([\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^N]) \in \mathbb{R}^{\mathcal{N}} \quad (4.19)$$

In the second step, the weak form is projected onto the properly selected low-dimensional subspace such that,

$$\langle \phi^k + \tau \nabla \phi^k, \mathbf{b} \cdot \nabla u_{\text{POD}} \rangle + \langle \nabla \phi^k, \mathbb{K}(\boldsymbol{\mu}) \nabla u_{\text{POD}} \rangle + \langle \phi^k + \tau \nabla \phi^k, \sigma \rangle = 0 \quad \forall k = 1, \dots, r \quad (4.20)$$

Note that only the diffusion term in the equation (4.20) is affine with respect to the input parameter  $\mathbb{K}(\boldsymbol{\mu})$  and can be efficiently reduced during the offline stage,

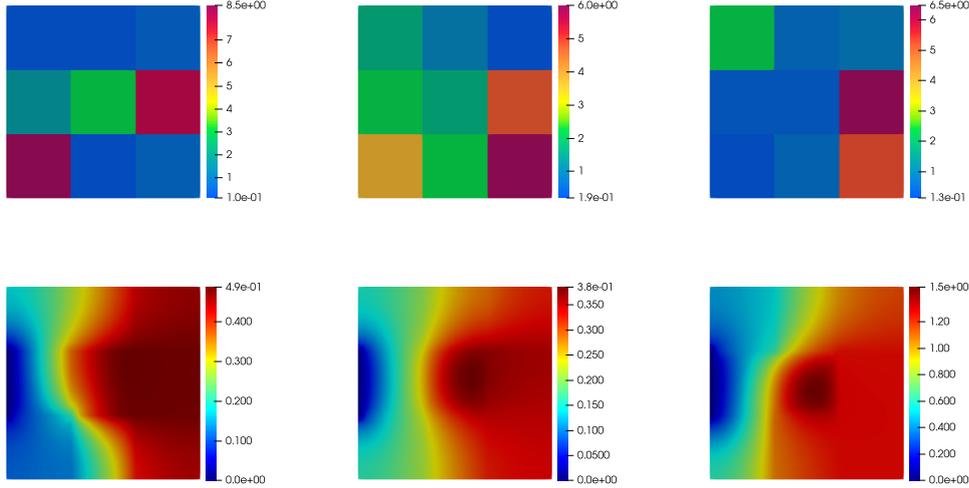
$$\langle \nabla \phi^k, \mathbb{K}(\boldsymbol{\mu}) \nabla u_{\text{POD}} \rangle = \mathbb{K}(\boldsymbol{\mu}) \sum_{j=1}^r \langle \nabla \phi^k, \nabla \phi^j \rangle z_j, \quad (4.21)$$

in which the  $(r \times r)$  operator  $\langle \nabla \phi^k, \mathbb{K}(\boldsymbol{\mu}) \nabla u_{\text{POD}} \rangle$  can be computed once and for all in the offline stage. During the online stage, if new parameter  $\mathbb{K}(\boldsymbol{\mu})$  is prescribed, the evaluation of the diffusion operator of eq. (4.20) can be done in reduced complexity (i.e. it does not depend on the original dimension  $\mathcal{N}$ ) since it only requires  $\mathcal{O}(r \times r)$  operations. This step is crucial for retaining the computational efficiency of the ROM, however, the same idea cannot be straightforwardly applied to the convective and source term of the equation (4.20) as the projection operator dependency on the input parameter i.e. diffusion coefficient is non-affine. However, the non-affineness is not addressed in this study; instead, the application of the suggested approach to a multi-dimensional PDE system is the main focus.

#### 4.4.2 Results and discussions

The input parameter  $\boldsymbol{\mu}$  is discretized using the LHS technique with 2500 sample points, from which the training set  $\mathcal{E}_{\text{train}} \subset \mathcal{D}$  consist of 2000 points and the remaining 500 samples are used for the validation set  $\mathcal{E}_{\text{val}} \subset \mathcal{D}$  to certify the quality of reduced basis approximation. Figure 4.11 represents the unknown field with different combinations of diffusion coefficients for each of the 9 blocks.

Similar to the previous numerical example, we perform the algorithm based on two different initialization of the low-fidelity model. First, the discussion is presented for the low-fidelity model approximation by a POD-G ROM built using a random sketch. Three studies are conducted to evaluate the computational performance: the first two examined the impact of the random sketch on the qualitative performance of the suggested method, while the third examined the quantitative performance of the algorithm by adjusting the greedy parameter for sampling. This algorithm is conducted over 10 trials for the first two cases, the convergence plot is shown in figure 4.12. In the first case study, the initial

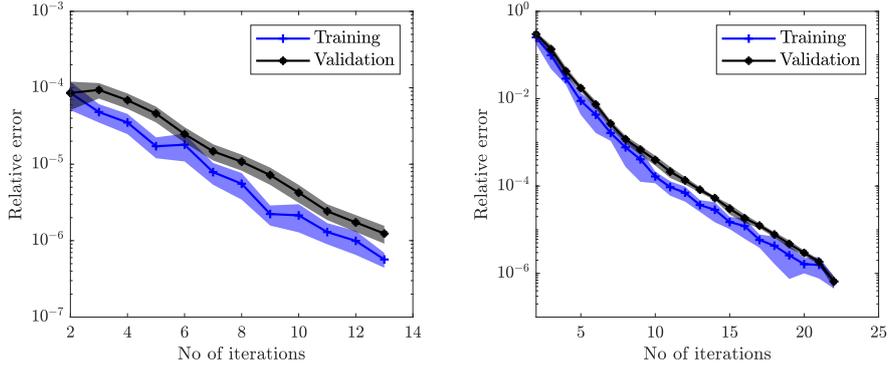


**Figure 4.11:** Three different representative solutions for the advection-diffusion problem shown (*below*) by varying nine different combinations of diffusion coefficient  $\mathbb{K}(\mu)$  in all the three domains (*top*).

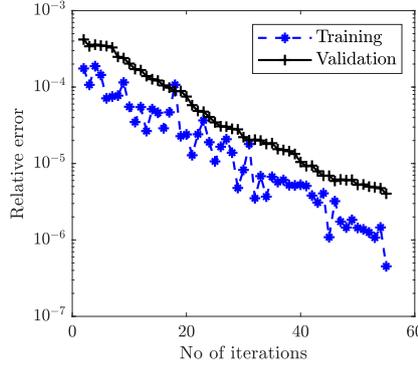
rank of the random sketch for constructing a ROM is chosen to be 100, and during every iteration cycle, 10 points are added incrementally to update the low-fidelity model. With a target tolerance set to  $\epsilon = 10^{-6}$ , we can observe that 13 iteration cycles are required to achieve the desired accuracy in each of the 10 trials, refer to figure 4.12a. A total count of  $100 + (10 * 12) = 220$  parametric points are sampled out of 2000 points from the training set. It is to be noted that the rank of the low-fidelity model is also enhanced by 10 which implies that all the points sampled are unique which is expected and as a consequence, the recovered basis functions by construction are linearly independent. In the second case study, the random sketch is initialized with 10 linearly independent snapshots from the training set instead of 100 and in each iteration, 10 points are sequentially added to recover the basis functions. It is observed from figure 4.12b, that in 22 iterations the target accuracy is achieved with a total sampling of 220 points out of 2000 points from the training parametric set, similar to the first case. Thus, it is evident that irrespective of the size of the random sketch chosen for the initial construction of the low-fidelity model, the algorithm performed well in both scenarios and the final enrichment of the low-fidelity model converged towards the FOM model accurately within the prescribed tolerance. The computational time required to achieve the target accuracy is  $\mathcal{O}(10^3)$  in seconds in both cases, while having comparable computational performance.

In the third case, the study is conducted by sampling 2 parametric points per iteration instead of 10 points. Figure 4.12c reflects 54 iteration cycles required to achieve the same target accuracy, with a total sampling of  $(100 + 54 * 2 = 208)$  points, unlike in the previous two cases where 220 points were selected from the parametric space. The CPU time required in this case is  $\mathcal{O}(10^4)$  seconds, which is one order higher than the previous cases. This implies adding a few points per iteration can minimize the risk of sampling excess points while maintaining the same order of accuracy, but at the cost of higher CPU time. Due to the discrete nature of error evaluation, the relative training error is observed to be noisy, but with the validation error plot, we can see a smooth decay of the curve as the error is evaluated over the entire validation set rather than at select discrete points.

For the second analysis, the initial low-fidelity model is built using a very coarse sketch model, as shown in figure 4.13a. For recovering the high-fidelity solution, a fine discretized model is used in figure 4.13b. The points are added sequentially by incrementing with 10



(a) initial rank 100, incremented by 10 points every iteration (b) initial rank 10, incremented by 10 points every iteration

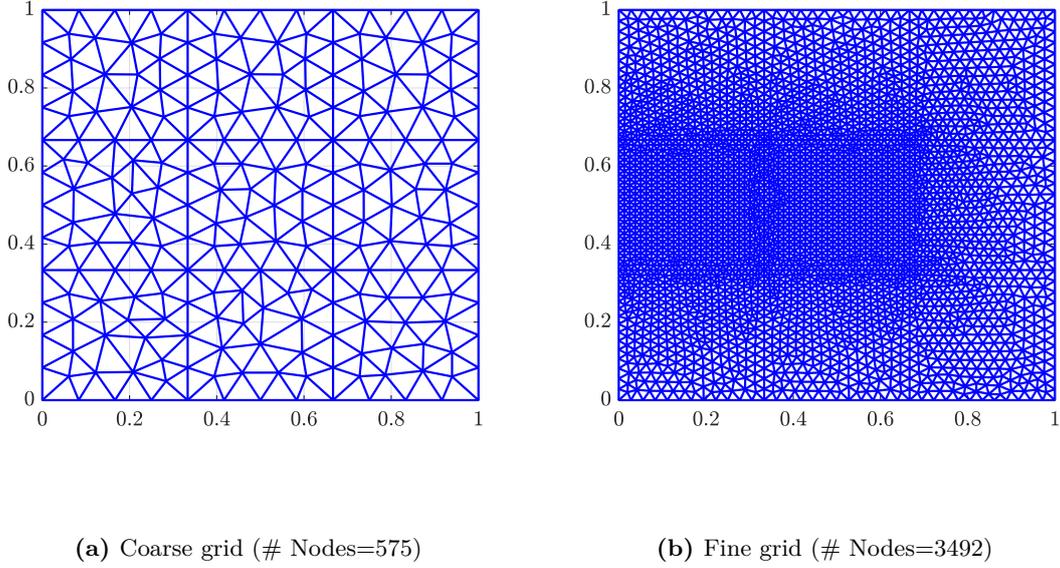


(c) initial rank 100, incremented by 2 points every iteration

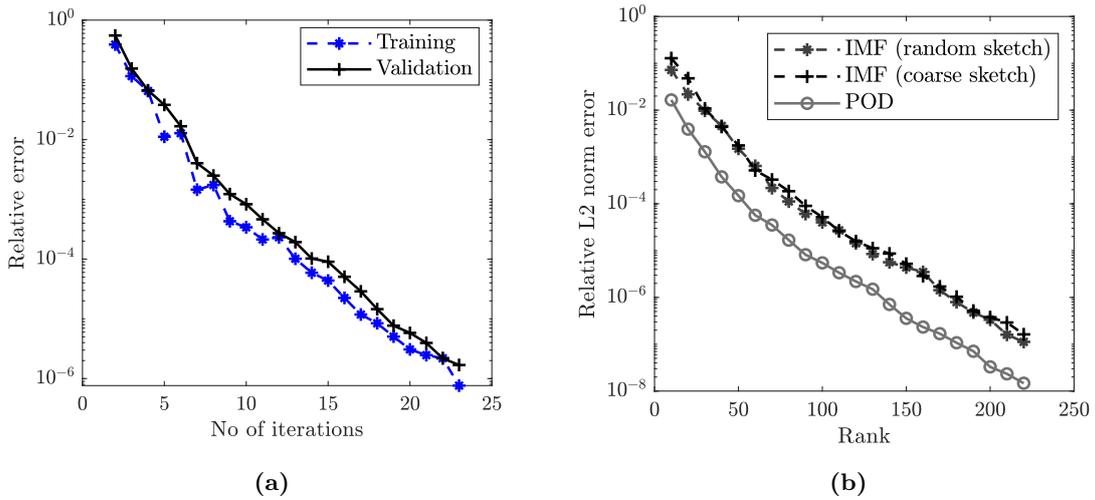
**Figure 4.12:** (a,b) Mean of error  $\epsilon_{train}$  and  $\epsilon_{val}$  between FOM and ROM solution obtained using different random sketch models over 10 trials and its std. deviation. (c) Error  $\epsilon_{train}$  and  $\epsilon_{val}$  between FOM and ROM solution for one trial.

every iteration. The target accuracy is achieved in 23 iterations (total count of sampled points is 230) as shown in figure 4.14a, which is more by 10 points compared to the random sketch model. As is already discussed, such types of PDE problems face numerical stability issues in case of high Peclet number (advection-dominated cases) and can be resolved by applying artificial diffusion in the upwind direction. It is also important to note that the amount of artificial diffusion added to the system depends on the mesh size. So coarser the mesh size, the more diffusion is required, which may affect the solution significantly. Hence, to accurately capture the physical properties of the PDE system, it is very important to consider that the initial grid takes into account all the physical aspects of the problem such that there is no loss of information. This could be the plausible explanation for a higher number of points needed when a coarse sketch model is used, as compared to POD-G ROM for the initial construction of the low-fidelity model. Nevertheless, our main objective to guarantee convergence is achieved irrespective of the initial size of the coarse mesh used and can be confirmed further from the decay of the validation error curve seen in figure 4.14b. The CPU time taken is of the  $\mathcal{O}(10^3)$  seconds, same as the first and second case.

Figure 4.14b shows the plot for  $\ell_2$  norm POD projection error and ROM error for all the parameters belonging to the training set using both sketch models. Both the POD and ROM errors decay as the rank of the system increases, with the POD error serving as a lower bound to the ROM error. This proves the reliability of the proposed method on the quality of ROM constructed such that irrespective of the initial design of the low-fidelity model, the ROM error displays similar decay properties as the POD error.



**Figure 4.13:** (a) Coarse discretized mesh for initial low-fidelity model, and (b) fine discretized mesh for high-fidelity model approximations.



**Figure 4.14:** Error  $\epsilon_{train}$  and  $\epsilon_{val}$  between FOM and ROM solution using coarse sketch model (a). Error  $\epsilon_{POD}$  of the classical POD projection, and error  $\epsilon_{ROM}$  of the solution of the reduced order model obtained by the proposed method (b).

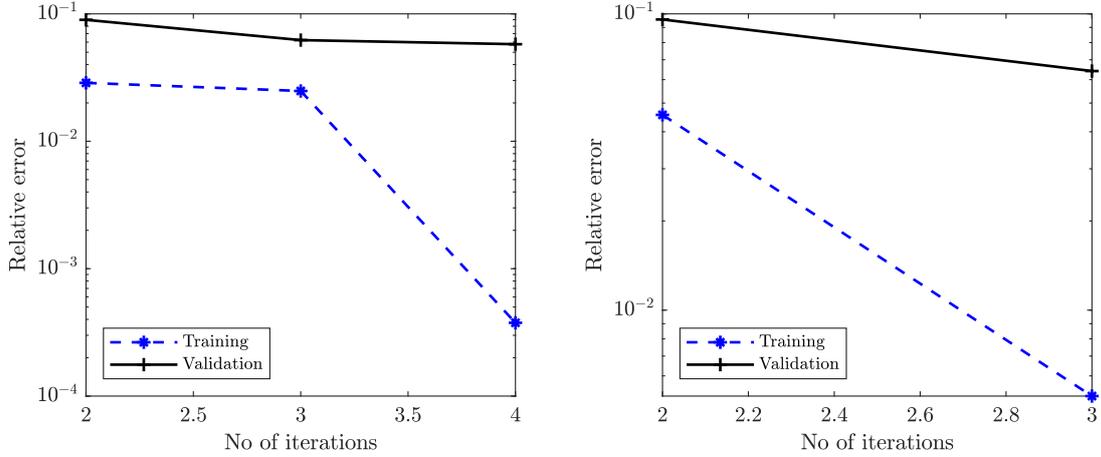
## 4.5 Numerical example of flow past a sphere problem

The third example investigated is the classical benchmark inertialess flow past a sphere problem obeying the Giesekus constitutive model, discussed thoroughly in section 3.3.2 of chapter 3. The quantity of interest is to analyze the variable log conformation tensor  $\mathbf{S}$  for different values of relaxation time of the fluid ( $\lambda$ ) lying in the range  $[0.1, 0.5]$  while keeping the rest of the fluid parameters fixed. The input parameter  $\lambda$  is discretized using the logarithmic scale with 15 sample points from which the training set  $\Xi_{train} \in \mathcal{D}$  consist of initial 10 points and the remaining 5 are used for the validation set  $\Xi_{val} \in \mathcal{D}$ . The training error is obtained by evaluating the maximum of  $\ell_2$ -norm error over the sampled  $\lambda_k$  values

from the training set between the high-fidelity and low-fidelity model approximation of the log-conformation tensor obtained at a fully developed steady state, which is given by,

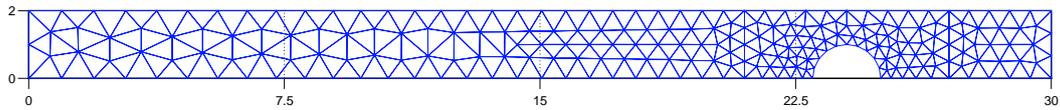
$$\epsilon_{train} = \max_{\lambda_k} \sqrt{\frac{\sum_{i=1}^{N'} \|\mathbf{S}_{HF}(x_i, t_n; \lambda_k) - \mathbf{S}^R(x_i, t_n; \lambda_k)\|_2^2}{\sum_{i=1}^{N'} \|\mathbf{S}_{HF}(x_i, t_n; \lambda_k)\|_2^2}} \quad \lambda_k \in \Xi_{train} \quad (4.22)$$

where  $\mathbf{S}_{HF}(\mathbf{x}, t_n; \lambda_k)$  represents the high fidelity solution at the final  $n^{th}$  time step obtained by solving the constitutive eq. (3.20) at the computed parametric points  $\lambda_k$  and  $\mathbf{S}^R(\mathbf{x}, t_n; \lambda_k)$  represents the equivalent reduced solution as obtained by solving the expression in (3.26).

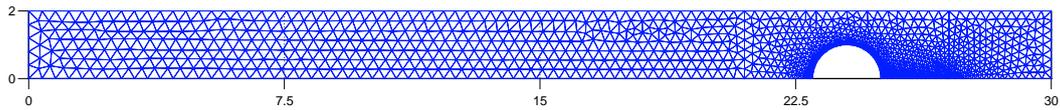


(a) Initial lo-fi model with a POD-G ROM (b) Initial lo-fi model with a coarse sketch model

Figure 4.15: Error  $\epsilon_{train}$  and  $\epsilon_{val}$  between FOM and reconstructed ROM solution.



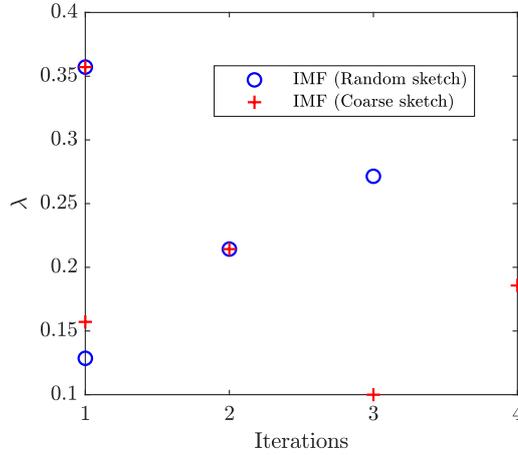
(a) Coarse grid (# Nodes =239)



(b) Fine grid (# Nodes =1381)

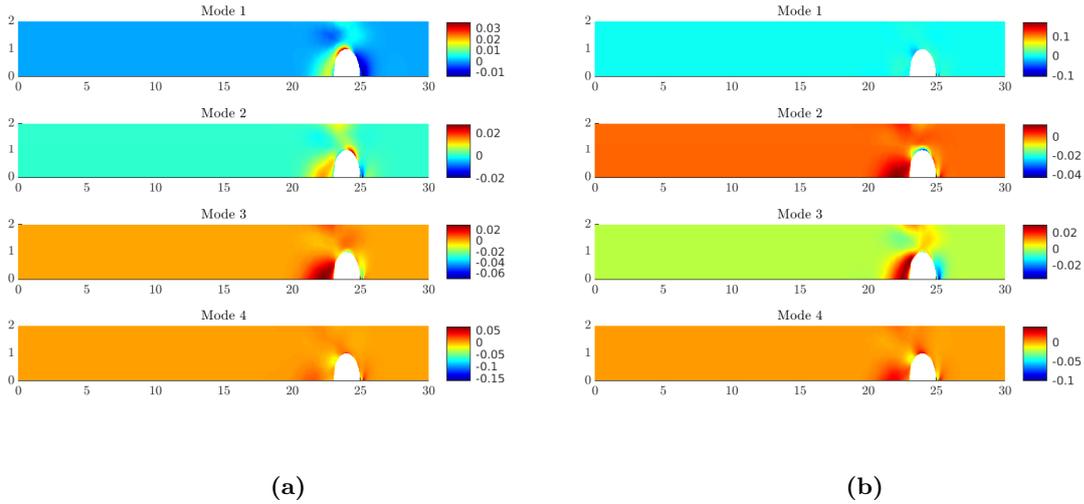
Figure 4.16: (a) a coarse discretized mesh for an initial low-fidelity model, and (b) a fine discretized mesh for high-fidelity model approximations.

The first analysis is predicated on the idea that we already have a random dataset of the FOM solutions. As mentioned previously, the initial low-fidelity model is then a POD-GROM created from this random sketch model. In this illustration, we considered the random sketch to be comprised of two linearly independent snapshots from the training set. During each iteration of the algorithm, we increment by one parametric point to update the low-fidelity model. Figure 4.15a illustrates how 4 iteration cycles are required to reach the acceptable precision when the target tolerance is set to  $\epsilon = 10^{-2}$ . This suggests that the low-fidelity model's rank is increased by a factor of 5 while achieving an accuracy of order  $\mathcal{O}(10^{-3} - 10^{-4})$ . Additionally, we also show the convergence plot for parameters belonging to the validation set, which in 4 iterations achieves an accuracy of order  $\mathcal{O}(10^{-1} - 10^{-2})$ .



**Figure 4.17:** Sampled parametric points in each iteration.

For the second case, we initialize the low-fidelity model using the coarse sketch model as shown in figure 4.16a. For recovering the high-fidelity solutions, the fine mesh model as illustrated in figure 4.16b is used. Likewise, we increment by one parametric point to update the low-fidelity model during each iteration of the algorithm. The target accuracy is achieved in 3 iterations (total count of sampled points is 4) as shown in figure 4.15b. We also show an illustration of sampled points obtained by both the sketch models in figure 4.17. It is clear that the points chosen are not repetitious because the computed points do not exactly correspond to the same location in each iteration. However, the points are not being extracted from the same location when both sketch models are used as was observed in the first numerical example, and the reason could be due to the way the algorithm operates and its dependency on the initial point selection. This needs further investigation to come to a definitive conclusion. We also show plots of the POD modes recovered using the proposed method via random sketch model and by classical SVD in figure 4.18. We observe that the modes generated in both the cases have identical representations, but not in the same order. A thorough study is required for cases where the sampling space comprises larger values of the Deborah number, which may present numerical challenges in contrast to smaller values of the Deborah number. By means of this numerical test, we demonstrated that sampling appropriate parametric points is possible regardless of the PDE formulation, however it may be difficult to construct error estimators, which was the major goal of this methodology.



**Figure 4.18:** POD modes obtained by classical SVD (a) and by proposed IMF approach (b).

## 4.6 Conclusions

In this chapter, we presented the feasibility of a multi-fidelity approach in reduced basis approximation for a multi-dimensional parametric PDE system in an iterative procedure. The parametric sampling is adhoc and extracted from low-fidelity model approximations based on heuristic assumptions. Such approximations of low-accuracy low-fidelity models over high-fidelity models enhance the computational performance in the offline stage significantly. Not to mention, this approach proved to be adequate when a posteriori error estimator is unavailable or difficult to obtain for complex PDEs, which is an essential ingredient for efficient greedy sampling. The greedy selection is user dependent, i.e. one can select a single point or multiple points for each iteration. However, attention is to be given as excess parametric points might be sampled than required to achieve the same target accuracy but at a higher CPU cost. So a compromise can be made on the trade-off between computational efficiency and accuracy. Overall, the construction of reduced basis subspace required less high-fidelity snapshot generation in the proposed method. We also presented two different ways of initializing a low-fidelity model in this work and irrespective of the initial quality of the low-fidelity model approximation, the method is shown to be reliable and stable by converging towards the FOM approximation within the prescribed tolerance.

This methodology is successfully demonstrated on a 2D steady-state advection-diffusion problem for 9 input parametric dimensions. A qualitative comparison is also presented for a simple steady-state heat conduction problem between the proposed method and greedy RBM, in which both cases had comparable computational performance. Last but not least, the algorithm was tested on the flow past sphere benchmark problem suspended in Giesekus fluid with relaxation time as the input fluid parameter. It was demonstrated that regardless of the PDE definition, the methodology allowed optimal parametric sampling in the absence of problem-specific error estimators.

In our current work, we have only performed linear reduction, but we can further embed hyper reduction in our current framework "on the fly" i.e. during the multi-fidelity iterations for treating non-affine problems. This step can alleviate the cost of low-fidelity model approximations significantly. For some of the current research work one can refer to the articles [176, 177] on adaptive hyper reduction techniques which allows enrichment of the reduced integration domain during the online stage as the simulation progresses.

Additionally, during the multi-fidelity iterations, we may also evaluate the low-fidelity model solely on a portion of the randomly chosen parametric points as opposed to the complete training set. This procedure could drastically improve the computational performance of the methodology, specially for nonlinear PDEs. However, there are certain implications to it such as there will be missing information in the parametric subspace and as suggested in the methodology, to sample points using the DEIM strategy, one needs to orthogonalize the current parametric functions in relation to the previous ones. As a possible solution to this problem, Gappy-POD may be used to reconstruct the missing data in the updated parametric functions and can be implemented into this technique for future research work.





## Chapter 5

# POD-ROM for spatially adapted snapshots

### Abstract

This chapter is dedicated to reduced basis construction for spatially adapted snapshots that are generated by resolving the underlying parametric PDE system using adaptive finite elements. This work is set up in two parts: Firstly, a projection-based POD-G ROM is derived by employing a multi-fidelity approach. This method will allow picking appropriate space-adapted snapshots belonging to a given training set via low-fidelity model approximation and using that information we recourse to a high-fidelity model to recover the POD modes that will be vectors of the same size. In the second part of the work, we propose a technique for the recovery of spatially adaptive reduced basis functions that are representative of select adaptive snapshots.

### 5.1 Introduction

Standard model order reduction techniques are applied on static snapshots, meaning all the snapshots use one and the same finite element mesh. Some fundamental problems that often appear in physical or mechanical applications are high gradient solutions in boundary layers, shock waves [178], edge singularities, scale heterogeneity such as in porous media [179], composite structures, and so on that demands very fine numerical resolution to accurately capture the such phenomenon. Indeed, the cost of resolving such physical characteristics with a fine uniform mesh becomes so significantly overwhelming that instead adaptive mesh refinements are employed, which can essentially capture such features with great accuracy while reducing the degrees of freedom profusely. There are numerous techniques developed for the mesh adaptation and can be referred to in the literature [180–184].

The inclusion of spatial adaptivity in the model order reduction framework is quite advantageous from two perspectives: utilizing adaptive finite elements for snapshot generation can greatly reduce the offline computational cost in comparison to the use of a fine uniform mesh. On the other hand, surrogate modeling can guarantee computational speed up as compared to FOM which we have discussed thoroughly in the previous chapters. However, applying classical model order reduction methods on space-adapted snapshots is not feasible. In a discrete setting, the adaptive snapshots are vectors of different lengths, and hence constructing a single snapshot matrix is not possible. This leads to the problem that the usual POD employed by SVD on the snapshot matrix cannot be carried out.

There are different concepts published in the articles [74, 76, 77, 185, 186] where MOR is considered by projection onto the reduced space generated from both space and time

adapted snapshots. In the recent publication of 2019 [76], the authors proposed a POD-Galerkin ROM by defining the reduced basis functions implicitly as a linear combination of original space-adapted snapshots. By doing so, the creation of a common finite element space of all the snapshots is avoided, but this requires evaluation of bi-linear forms for all pairs of snapshots. Parallel to this research work, the authors in the article [77] also proposed to build a POD-ROM for evolving PDE equations for any arbitrary discretization by expressing reduced basis functions in terms of snapshots in a continuous setting that eliminates the requirement of creating a common mesh but at the expense of computing the inner product of the intersections of the finite element ansatz functions. In the context of reduced basis methods [185, 187], the authors derived residual bounds with respect to infinite dimensional truth solution. An adaptive wavelet discretization scheme is used for snapshots generation in the offline phase that allows to numerically approximate dual norm infinite dimensional residual and, in contrast, [74] the authors derive an error bound with mixed finite element formulation. In the article [188], an interpolation approach is outlined to eliminate spatial discretization of the snapshots. The idea is rather to interpolate given snapshots of arbitrary spatial discretization by piecewise polynomials. In another work, [189] the authors performed POD on an ocean model which is discretized using a dynamic spatial adaptive mesh model. A fixed reference mesh is utilized for interpolating all the snapshots so that snapshots at each time level are vectors of the same size, and then the usual POD approximation on this reference mesh is carried out. In fact, in this work as well, we construct a fixed reference mesh and the snapshots from their own mesh are interpolated onto this fixed reference mesh upon which the basis functions are computed via POD approximation. The novelty of this work is we do not interpolate all the snapshots belonging to the training set, rather only a few picked snapshots are interpolated onto the reference mesh.

To materialize the idea, we employ the multi-fidelity modeling approach discussed in the previous chapter combined with POD-ROM approximation for high dimensional parametric PDE systems. We leverage low-fidelity models for parametric space exploration iteratively, and only the selected adaptive snapshots are represented as members of a high-fidelity model for recovery of the basis functions. The high-fidelity model in this case is the fine, uniform reference mesh. The ROM is built by the usual Galerkin projection onto the subspace enriched by the computed RB functions. The procedure is repeated until the reduced subspace is spanned by a set of orthogonal RB functions that are a good representation of the entire set of snapshots. Additionally, in this work, we also present a technique for recovering adaptive reduced basis functions, in a discrete setting, that can be expressed as a linear combination of the select adaptive snapshots computed in the first part.

The overview of this chapter is: We first discuss the problem definition in section 1, followed by a brief explanation of mesh adaptation using in-house IciTech libraries [184, 190, 191] in section 2. Next, we present the framework of the proposed methodology in section 3, while discussing the design of both low-fidelity and high-fidelity models, and finally, in section 4, we demonstrate the methodology on a 2D diffusion-reaction PDE problem utilizing adaptive meshes for a varying range of reaction coefficient.

## 5.2 Problem description

Let us consider a 2D steady-state diffusion reaction problem in a bounded domain  $\Omega = [0, 1] \times [0, 1]$  which satisfies the following equation,

$$-\Delta u + \lambda u = 1 + \lambda \quad \text{on } \Omega \quad u = 0 \quad \text{on } \partial\Omega \quad (5.1)$$

Here "u" is the concentration variable, and we are interested to solve the problem for different values of the reaction coefficient  $\lambda$ . An approximate solution  $u_h \in \mathcal{V}^{\mathcal{N}} \subset \mathcal{V}$  is sought by Galerkin finite element formulation. By multiplying with a test function  $v_h \in \mathcal{V}^{\mathcal{N}}$ , the weak form of the equation can be written as:

$$\int_{\Omega} \nabla u_h \cdot \nabla v_h d\Omega + \int_{\Omega} \lambda u_h v_h d\Omega = (1 + \lambda) \int_{\Omega} v_h d\Omega \quad \forall v_h \in \mathcal{V}^{\mathcal{N}} \quad (5.2)$$

where,

$$u_h(\mathbf{x}; \lambda) = \sum_{k=1}^{\mathcal{N}} N_k(\mathbf{x}) u_k(\lambda)$$

$N_k$  are the shape functions of choice and  $u_k(\lambda)$  represent the scalar values of the field  $u_h$  at discretization points  $\mathbf{x}_k, k = 1 \dots \mathcal{N}$ . The equation (5.1) in algebraic form can be written as,

$$(\mathbf{K} + \lambda \mathbf{M}) \mathbf{u}(\lambda) = (1 + \lambda) \mathbf{f} \quad (5.3)$$

$$\begin{aligned} \mathbf{K} &= \left\{ k_{ij} = \int_{\Omega} \nabla N_i \cdot \nabla N_j d\Omega \right\} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}} \\ \mathbf{M} &= \left\{ m_{ij} = \int_{\Omega} N_i N_j d\Omega \right\} \in \mathbb{R}^{\mathcal{N} \times \mathcal{N}} \\ \mathbf{f} &= \left\{ f_i = \int_{\Omega} N_i d\Omega \right\} \in \mathbb{R}^{\mathcal{N}} \end{aligned}$$

The high fidelity snapshots  $\{\mathbf{u}^i\}_{i=1}^N \in \mathbb{R}^{\mathcal{N}}$  are generated for all the reaction coefficient parameters lying in the training set,  $\lambda \in \Xi_{train} \subset \mathcal{D}$  and  $N = \text{card}(|\Xi_{train}|)$ . Now, the reduced weak form can be obtained by replacing  $u_h$  in eq. (5.2) by,  $u_h^R$  which is expressed as:

$$u_h^R(\mathbf{x}; \lambda) = \sum_{k=1}^r \phi^k(\mathbf{x}) a_k(\lambda) = \Phi \mathbf{a} \quad r \ll \mathcal{N} \quad (5.4)$$

where,  $\mathbf{a} = (a_1, \dots, a_r)^\top$  are the POD expansion coefficients and  $\Phi = [\phi^1, \phi^2, \dots, \phi^r]$  are the POD modes.

By Galerkin projection, equation (5.3) in reduced form is given by,

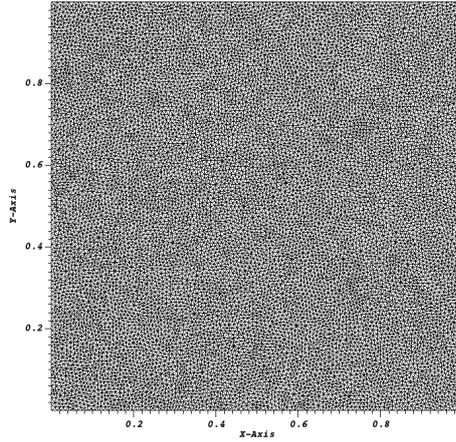
$$\left( \underbrace{\Phi^\top \mathbf{K} \Phi}_{\text{pre-computed}} + \lambda \underbrace{\Phi^\top \mathbf{M} \Phi}_{\text{pre-computed}} \right) \mathbf{a} = (1 + \lambda) \underbrace{\Phi^\top \mathbf{f}}_{\text{pre-computed}} \quad (5.5)$$

The above equation is now a system of order  $\mathcal{O}(r \times r)$  greatly reduced from  $\mathcal{O}(\mathcal{N} \times \mathcal{N})$ .

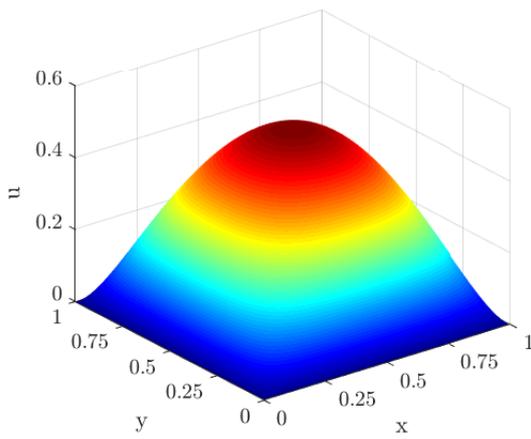
### 5.2.1 Numerical analysis

We solve for the unknown field variable  $u$  for a parameter  $\lambda$  in the range of  $[10^1, 10^3]$  which is spaced logarithmically with 100 points. This problem in particular has a wide range of variations in the solution field, so in order to represent well enough the entire parametric space, we first produce results with a very fine uniform mesh of mesh size  $h = 0.002$  illustrated in figure 5.1. Figure 5.2 illustrates the diffusion profile for  $\lambda = 10, 102, 497$  and 1000.

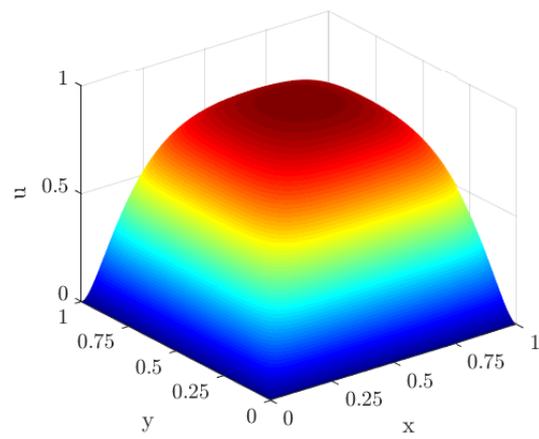
Now, let us look carefully at the solution profile for  $\lambda = 10$  and 1000. We observe that for  $\lambda = 10$ , the diffusion is concentrated in the center as expected (fig. 5.2a), and as the lambda value gets higher the solution becomes more saturated that it almost forms like a cube shape as can be seen in figure 5.2d. Steep gradients are observed along the boundaries



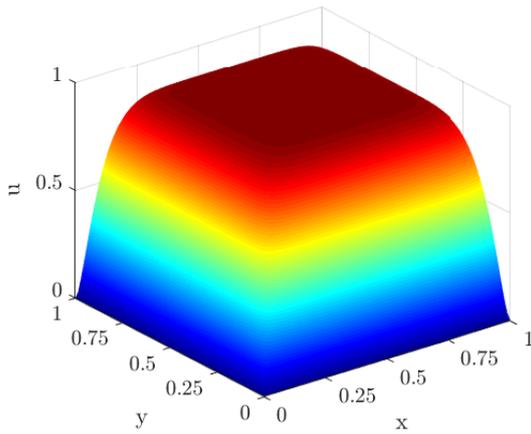
**Figure 5.1:** Uniform fine mesh representation ( $\#Nodes = 316714$ )



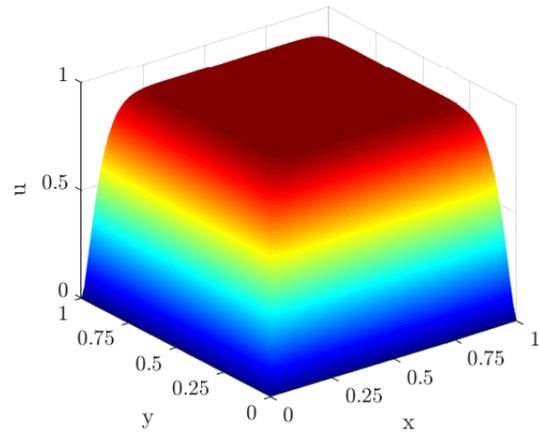
(a)  $\lambda = 10$



(b)  $\lambda = 102$



(c)  $\lambda = 497$



(d)  $\lambda = 1000$

**Figure 5.2:** Solution profile for different values of  $\lambda$

for  $\lambda = 1000$ , therefore it demands a very fine mesh to accurately capture the behavior in that region, whereas for smaller lambda values for example  $\lambda = 10$  the mesh can be

relatively coarser as the gradient change is rather gradual than steep.

Hence, we can deduce that it is not computationally efficient to apply a uniform fine mesh across the domain for all the lambda values, for there is little to no variation in values in the center of the mesh and drastic changes on the boundary for the snapshots where the  $\lambda$  values are large. Thus, this problem can be justified as a classic example of utilizing adaptive meshes to suit the model accordingly and our final aim is to apply POD-ROM on the generated adaptive snapshots. The next section is dedicated to mesh adaptation discussions.

### 5.3 Mesh adaptation

The most commonly used mesh adaptation techniques are h, p, r adaptation, and hp adaptation [180–184]. In h-adaptation, mesh refinement is typically done in a specific location by altering the mesh size, while in p-adaptation the polynomial order of the shape function is increased. It involves enhancing the degrees of interpolation for local elements. On the other hand, in r-adaptation, the nodes are re-positioned especially near the interfaces without changing their topology. The framework of hp-adaptation is a combination of both h- and p-adaptation.

The mesh refinement in the numerical work presented in this work is h-adaptation, using the ICITech library. It is built on a mesh topology optimization generator by executing local operations to incrementally enhance an initially inadequate mesh. In ICITech, an automatic anisotropic mesh adaptation technique is used based on metric tensors [190, 192, 193]. The Riemannian metric space and the unit mesh, rather than Euclidean space, are the foundations of the metric-based approach (for example, equilateral triangles in a 2D mesh or regular tetrahedrons in 3D meshes). In other words, the components that make up the mesh in Euclidean space can be of any shape (unit or not), and they will be converted into unit elements in the corresponding metric tensor space. Since the refinement is carried out in a particular direction [194] in anisotropic mesh adaptation and the metric fields are computed using a posteriori estimator [195], the execution of anisotropic meshing is more challenging and complex than the generation of an isotropic mesh. However, a substantial number of nodes are reduced in anisotropic mesh adaptation, providing a lower, computational time and less memory capacity. In practice, the most well-established error analysis enables the calculation of a metric tensor on an element basis, however, in the IciTech library, the metric field is calculated directly at the nodes of the mesh for direct use in the meshing tools.

Before detailing the methodology, let us first present some notations, properties, and definitions of mesh topology.

#### 5.3.1 Mesh topology

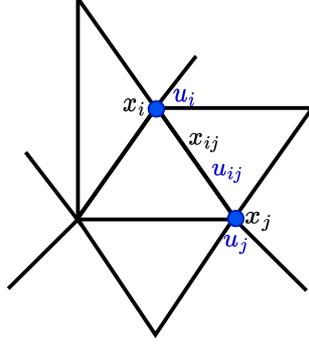
Let  $\mathcal{N}$  be a finite set of nodes in  $\Omega$ , and  $\mathcal{T}$  denote a set of  $T$  elements,  $T$  being a  $d$ -simplex whose vertices belong to  $\mathcal{N}$ . Let  $\mathcal{F}(\mathcal{T})$  be the set of faces of  $\mathcal{T}$ 's elements, while a face is designated by  $F$ . A basic property is

$$1 \leq \text{card}(\mathcal{T}(F)) \leq 2, \forall F \in \mathcal{F}$$

and the faces belonging to the boundary of  $\mathcal{T}$  are:

$$\partial\mathcal{T} = \{F \in \mathcal{F}, \text{card}(\mathcal{T}(F)) = 1\}$$

Each face  $F \in \mathcal{F}(T)$  shares no more than two elements, and only the boundary faces have one element.



**Figure 5.3:** Illustration of the edge vector  $\mathbf{x}_{ij}$  of the edge joining nodes  $i$  and  $j$ , and the edge solution,  $\mathbf{u}_{ij}$  joining the solution at the nodes  $i$  and  $j$ .

**Theorem:**  $(\mathcal{N}, \mathcal{T})$  is a mesh of  $\Omega$  if and only if the elements of  $\mathcal{T}$  are non-degenerated and

$$\sum_{T \in \mathcal{T}} |T| = |\Omega| \quad (5.6)$$

That is, among all mesh topologies, valid meshes are those that satisfy the minimal volume criterion [193, 194]. In addition to satisfying the minimal volume property, one must ensure the topology is unique. This is achieved by checking the geometrical quality of elements. The evaluation of the quality of each element of the mesh topologies is done by computing the shape factor,

$$c(T, \mathbf{x}) = \frac{|\Omega_T|}{h(T)^d} \quad (5.7)$$

where,  $h(T, \mathbf{x}) = \sum_{i,j \in T} \|\mathbf{x}_i - \mathbf{x}_j\|$  such that  $\mathbf{x} : \mathcal{N} \rightarrow \mathbb{R}^d$ . This shape factor varies between 0 and 1, with 1 being the best quality representing the equilateral simplex, and 0 being the flat element.

The operation of the local modification of the mesh topology is done by simply cut-and-paste operation:

$$\mathcal{T} \leftarrow \mathcal{T} - a + b \quad (5.8)$$

$a$  being the subset of elements to be deleted chosen in  $\mathcal{T}$  and  $b$  a new set of elements to be inserted.  $a, b$  must have the same boundary, i.e.  $\partial b = \partial a$ . Although, this operation is valid only if the result is still a mesh topology and follows the above-mentioned criteria. Details on the selection procedure of  $a$  and  $b$  can be referred to in [184].

### 5.3.2 Mesh optimization

Let us recall that  $\mathbf{x}_i \in \mathbb{R}^d, i = 1, \dots, \mathcal{N}$  is the set of nodes of the mesh and that the vector  $\mathbf{x}_{ij}$  connects nodes  $i$  and  $j$ , and  $\mathbf{x}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ , as illustrated in figure 5.3. Let  $\mathcal{Z}(i)$  be the patch associated with the vertex  $\mathbf{x}_i$  of the mesh shared by several edges.

The problem of finding a unitary metric  $\mathbb{M}^i$  associated with the  $i^{\text{th}}$  node can be formulated as the least squares problem [190]. Recall,  $u_h$  is the numerical solution approximated on a given mesh, and let ' $K$ ' be the desired number of edges for the mesh. The aim is to create the "best" mesh  $(\mathcal{N}, \mathcal{T})$ , or in other words, construct an optimal metric field tensor  $\mathbb{M}$  such that the interpolation error  $|u_h - \Pi u_h|$  is minimized in  $\ell_1$ -norm.  $\Pi u_h$  is the linear interpolation of the solution  $u_h$  on the mesh,  $\mathcal{T}$  i.e.  $\Pi u_h(\mathbf{x}_i) = u_h(\mathbf{x}_i) \forall i \in \mathcal{N}$ . The local interpolation error distributed over the edge can be written as,

$$\tilde{e}(s) = \frac{1}{2} \sum_{ij} s_{ij}^2 e_{ij} = \frac{1}{2} \sum_{ij} s_{ij}^2 |\mathbf{g}_{ij} \cdot \mathbf{x}_{ij}| \quad (5.9)$$

where  $s_{ij}$  stands for mesh scaling factor,  $e_{ij}$  is the calculated error, and  $\mathbf{g}_{ij}$  represents the gradient along and in the direction of the edge  $\mathbf{x}_{ij}$ . For the desired number of edges  $K$ , the following minimization problem is solved:

$$\begin{aligned} \min \epsilon(s) &= \min \int_{\Omega} |\tilde{e}(s)| d\Omega \\ \text{s.t.}, \sum_{ij} n_{ij} &= K \end{aligned} \quad (5.10)$$

where  $n_{ij}$  is the evaluation of the number of created edges in relation to the scaling factor  $s_{ij}$ . The resulting optimal metric solution of the equation (5.10) defined at the mesh nodes is then given by,

$$\mathbb{M}^i = \left( \frac{1}{d} \sum_{j \in \mathcal{Z}(i)} s_{ij}^2 \mathbf{x}_{ij} \otimes \mathbf{x}_{ij} \right)^{-1} \quad (5.11)$$

where,

$$s_{ij} = \left( \frac{\lambda}{e_{ij}} \right)^{1/p}$$

and,

$$\lambda = \left( \frac{\sum_i \sum_{j \in \mathcal{Z}(i)} e_{ij}^{\frac{p}{p+2}}}{K} \right)^{\frac{p+2}{p}}$$

$p \in [1, d]$  is the penalty factor. Moreover, when  $s_{ij} = 1, \forall i, \forall j \in \mathcal{Z}(i)$  then the mesh is optimal. The recovered edge gradient  $\mathbf{g}_{ij}$  is reconstructed from the numerical solution  $u_h$  by solving the following minimization problem:

$$\begin{aligned} \mathbf{g}_{ij} &= \operatorname{argmin}_{\mathbf{g}} \sum_{j \in \mathcal{Z}(i)} |(\mathbf{g} - \nabla u_h) \cdot \mathbf{x}_{ij}|^2 \\ &= \operatorname{argmin}_{\mathbf{g}} \sum_{j \in \mathcal{Z}(i)} |\mathbf{g} \cdot \mathbf{x}_{ij} - (u_h(\mathbf{x}_i) - u_h(\mathbf{x}_j))|^2 \end{aligned} \quad (5.12)$$

Also, we are seeking a unitary metric tensor where each transformed edge vector is almost equal to 1, so that the sum of the transformed edge vectors in  $\mathcal{Z}(i)$  is,

$$\sum_{j \in \mathcal{Z}(i)} (\mathbb{M}^i \mathbf{x}_{ij}, \mathbf{x}_{ij}) = \sum_{j \in \mathcal{Z}(i)} 1 = |\mathcal{Z}(i)| \quad (5.13)$$

From equations (5.12), (5.13) and (5.10) an approximation of the optimal metric tensor is derived. The volume is adapted by local mesh modification of the previous mesh using local mesh operations such as node insertion, deletion, edge and face swap, collapse, and node displacement while satisfying the minimum volume theorem [194]. Finally, on the new mesh, the field solutions are linearly interpolated. The process is considered converged when the difference between the computed metrics in the current and previous iteration is small.

### 5.3.3 Extension to parametric field adaptation

The error metric discussed above has been built taking into consideration only one field variable. The work has been further extended to construct a unique metric directly from a multi-component vector field containing velocity, pressure, or other field variables [192]. In the considered diffusion-reaction problem, a unique metric is built taking into consideration all the snapshots generated for different values of reaction coefficient  $\lambda$  belonging to the training set  $\Xi_{train}$ ,  $\mathbf{u} = \{u_1, u_2, \dots, u_N\}$ . The error vector for all these snapshots is then defined as,  $\mathbf{e}_{ij} = \{e_{ij}^1, e_{ij}^2, \dots, e_{ij}^N\}$ . Then, the stretch factor is

$$\mathbf{s}_{ij} = \left( \frac{\|\lambda_{ij}\|}{\|\mathbf{e}_{ij}\|} \right)^{\frac{1}{p}}$$

The norm can be  $\ell_2$ ,  $\ell_1$  or  $\ell_\infty$ . In the numerical example presented, we used the  $\ell_2$  norm to compute the error,

$$\|\mathbf{e}_{ij}\|_{\ell_2} = \left( \sum_{k=1}^N (e_{ij}^k)^2 \right)^{\frac{1}{2}}$$

## 5.4 Part 1: Iterative multi-fidelity modeling (IMF) for building PODG-ROM for spatially adapted snapshots

Traditionally, if  $\{u_h^i(\mathbf{x}; \lambda)\}_{i=1}^N \in \mathbb{R}^{\mathcal{N}}$  represents the FOM solutions of the PDE problem (5.1) that belong to a single and same finite element space  $\mathcal{V}^{\mathcal{N}}$ , and can be suitably well approximated in a low-dimensional manifold (definition refer 2.2), then the solution can be represented in a separated form as,

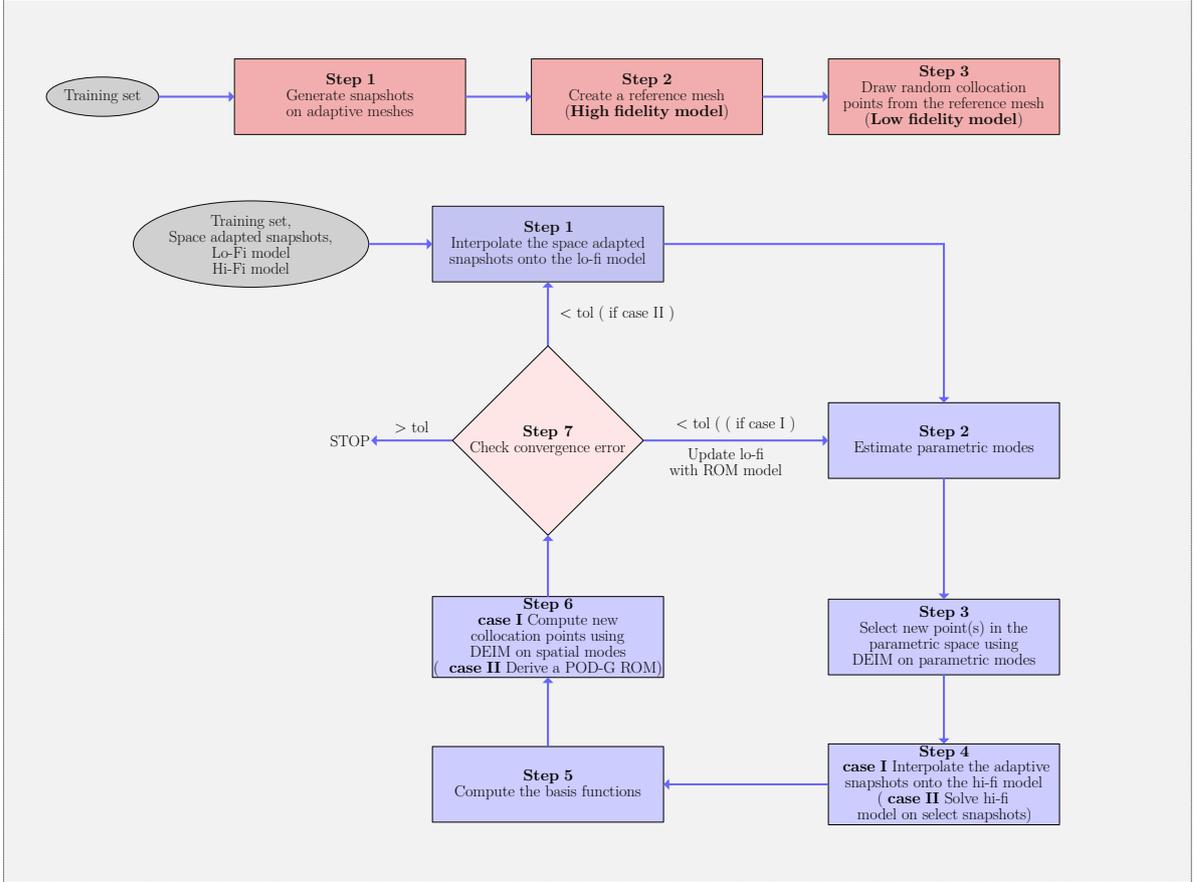
$$u_h(\mathbf{x}; \lambda) = \sum_{i=1}^r \phi^i(\mathbf{x})(\psi(\lambda))_i \quad (5.14)$$

Here,  $\{\phi^i\}_{i=1}^r \in \mathbb{R}^{\mathcal{N}}$  represents the reduced basis functions which spans the low dimensional subspace, i.e.  $\mathcal{V}^R = \text{span}\{\phi^1, \dots, \phi^r\} \subset \mathcal{V}^{\mathcal{N}}$  and  $\psi(\lambda) : \mathcal{D} \rightarrow \mathbb{R}$  are parametric functions that span the parametric space.

However, in this work we consider that the snapshots belong to different subspaces  $\mathbf{u}_h^1 \in \mathcal{V}_1^{\mathcal{N}_1}$ ,  $\mathbf{u}_h^2 \in \mathcal{V}_2^{\mathcal{N}_2}$ ,  $\dots$ ,  $\mathbf{u}_h^N \in \mathcal{V}_N^{\mathcal{N}_N}$  where  $\mathcal{V}_1^{\mathcal{N}_1}, \mathcal{V}_2^{\mathcal{N}_2} \dots \mathcal{V}_N^{\mathcal{N}_N} \subset \mathcal{V}$  and  $\mathcal{N}_1, \mathcal{N}_2, \dots, \mathcal{N}_N$  are the dimensions of the respective spaces. Hence, constructing the basis functions  $\phi^i(\mathbf{x})$  and  $\psi_i(\lambda)$  with traditional POD is impossible unless all the snapshots are interpolated onto a reference mesh such that the snapshots are the representative solution of the reference mesh.

Similar to the approach of multi-fidelity methods as explained in the previous chapter is extended to RB construction of spatially adapted snapshots. Although the idea remains the same, the framework is slightly different in this case. The flow of the proposed method is shown in figure 5.4.

We have divided the flow diagram into two parts: the first part shows the steps for the initialization of a low-fidelity model, and the second part reflects the steps of the methodology. To create a low-fidelity model approximation, first, we need to generate adaptive snapshots which are done using the procedure explained in section 5.3, followed by the construction of a reference mesh. A reference mesh can be of two types: a uniform fine mesh, or also it could be a commonly adapted mesh which is a good representative mesh for all the snapshots (sec. 5.3.3). In the numerical example presented, the reference model



**Figure 5.4:** Flow of the proposed method

$(\mathcal{N}_+, \mathcal{T}_{M_+})$  is represented by a uniform fine mesh, with  $\hat{\mathbf{x}}_i, i = 1 \dots \mathcal{N}_+$  denoting its spatial coordinates. We introduce a space  $\mathcal{V}^{\mathcal{N}_+} \subset \mathcal{V}$  such that  $\mathcal{V}_1^{\mathcal{N}_1} + \mathcal{V}_2^{\mathcal{N}_2} + \dots + \mathcal{V}_N^{\mathcal{N}_N} \subset \mathcal{V}^{\mathcal{N}_+}$ .

A low-cost, low fidelity model approximation mathematically can then be defined as a function that maps all the solutions lying in different finite element spaces to a set of randomly drawn collocation points of the reference mesh,

$$f_{\text{LF}}^{\text{init}} : u_h^i(\mathbf{x}) \rightarrow u_{\text{LF}}^i(\hat{\mathbf{x}}_g), \quad i = 1 \dots N \quad (5.15)$$

Here,  $\hat{\mathbf{x}}_g = \text{RANDSAMPLE}(\mathcal{N}_+, 'm') \subset \hat{\mathbf{x}}, \quad m \ll \mathcal{N}_+$ . RANDSAMPLE is a MATLAB function that returns 'm' values sampled uniformly at random without repetition. Note that collocation points in this work are given either by nodal points or integration points.

In step 2, parametric functions are computed from this low fidelity model approximation that spans the parametric subspace,  $\mathcal{D} = \text{span}\{\boldsymbol{\psi}_i\}_{i=1}^N \in \mathbb{R}^N$ . Followed by parametric sampling in step 3 which is obtained by applying DEIM over these parametric functions. The detailed steps are explained in the algorithm 5

Next, in step 4, two choices are proposed: The first choice is to interpolate the selected adaptive snapshots onto the reference mesh. A high-fidelity model approximation can then be mathematically defined as a function that maps all the picked adaptive snapshots onto the nodal (integration) points of the reference mesh,

$$f_{\text{HF}} : u_h(\mathbf{x}; \lambda_k) \rightarrow u_{\text{HF}}(\hat{\mathbf{x}}; \lambda_k), \quad \forall k \in (1, N), u_{\text{HF}} \in \mathbb{R}^{\mathcal{N}_+} \quad (5.16)$$

The second choice is to directly solve the PDE system (5.1) for the select sampled points using the fine uniform grid.

In step 5, high fidelity basis functions  $\phi_{\text{HF}}^i \in \mathcal{V}_+^R$  are then recovered with usual POD procedure and therefore enriching the reduced basis subspace,  $\mathcal{V}_+^R = \text{span}(\{\phi_{\text{HF}}^i\}_{i=1}^k) \in \mathbb{R}^{\mathcal{N}_+}$  where  $R = \dim(\mathcal{V}_+^R)$ , the construction is properly explained in the algorithm 4.

Since the first approximation for the parametric functions is likely to have some error, the recovery of high-fidelity basis functions may not be accurate and reliable to represent the large-scale PDE system, hence the procedure is an iterative one.

Now, in step 6, we construct two types of ROM. The first choice is an updated collocation model (a similar procedure used for approximating the initial low fidelity model), except the points drawn, are not random but rather are computed using DEIM on the recovered spatial basis functions. The second choice is a physics-based reduced order model by Galerkin projection, as derived in eq. (5.5).

Finally, in step 7, the accuracy of the low-fidelity model is evaluated. If the  $\ell_2$  norm error between the reduced bases of the current and the previous iteration of the loop is below a certain prescribed tolerance, the algorithm is terminated. Else, we update the low-fidelity data with the current ROM approximation, either by interpolation of the snapshots onto the collocation points or by POD-G ROM approximation. The above procedure is repeated until the low-fidelity model constructed can approximate all the adaptive snapshots from the training set lying in different finite element spaces adequately.

---

**Algorithm 4** POD procedure for basis computation.

---

```

1:  $\Phi \leftarrow \text{POD}(\mathbf{S}, \Phi, \delta_g)$ 
2: if  $\Phi = [ ]$  then
3:    $[\phi_k, \sigma_k, \psi_k] = \text{svd}(\mathbf{S})$ 
4:    $\Phi \leftarrow [\phi_1, \dots, \phi_k]$ 
5: end if
6: for  $l = 1 : \dim(\mathbf{S})$  do
7:    $\phi^l = \mathbf{S}^l - \Phi \langle \Phi, \mathbf{S}^l \rangle$ 
8:   if  $\frac{\|\phi^l\|}{\|\mathbf{S}^l\|} > \delta_g$  then
9:      $\Phi \leftarrow [\Phi, \frac{\phi^l}{\|\phi^l\|}]$ 
10:  end if
11: end for

```

---



---

**Algorithm 5** DEIM Algorithm adopted from [108]

---

```

1:  $\Xi_g \leftarrow \text{DEIM}(\Psi)$ 
2:  $\lambda^{(1)} = \max\{|\psi_1|\}, \quad \Xi_g = [\lambda^{(1)}]$ 
3: for  $l = 2 : \dim(\Psi)$  do
4:   Solve  $\mathbf{c} = [\Psi_{(1:l-1, 1:l-1)}]^{-1} \psi_{(1:l-1, l)}$ 
5:    $\mathbf{r} = \psi_l - \Psi \mathbf{c}$ 
6:    $\lambda^{(l)} = \max\{|\mathbf{r}|\}$ 
7:    $\Xi_g \leftarrow [\Xi_g, \lambda^{(l)}]^\top$ 
8: end for

```

---

## 5.5 Part 2: Recovery of the adaptive basis functions

In this section, we expand the above-discussed methodology into the construction of spatially adaptive reduced basis functions that can be explicitly defined as a linear combination of

‘ $K$ ’ selected parameters such that  $K = \text{card}(\lambda_k)$  and  $k \in (1, N)$ . Since the picked snapshots lie in different finite element spaces, it is important to construct a common finite element space of subsets of select snapshots. Let us define a new finite element space corresponding to the linear combination of the adaptive meshes that corresponds to the selected snapshots,  $\mathcal{V}_{1..K}$  with the properties such that,

1.  $\sum_{n=1}^K \alpha_n \mathcal{V}_k^{\mathcal{N}_k} \subset \mathcal{V}_{1..K} \quad \forall k \in (1, N)$
2.  $\mathcal{V}_{1..n}$  is of the same type as  $\mathcal{V}_k^{\mathcal{N}_k}$  with  $n = 1 \dots K$

Finally, the adaptive reduced basis functions can be expressed as,

$$\tilde{\phi}^n = \sum_{n=1}^K \hat{u}_h^n(\cdot; \lambda_k) \alpha_n \quad \forall k \in (1, N), \quad (5.17)$$

where,  $\{\hat{u}_h^n\}_{n=1}^K \in \mathcal{V}_{1..K}$  are snapshots belonging to the new commonly adapted mesh. First, we need to compute the coefficients  $\alpha_n$ . Once the selected adaptive snapshots have been identified at the end of the proposed algorithm, QR decomposition is applied to the high-fidelity snapshot matrix consisting of the selected snapshots interpolated onto the high-fidelity model. It follows:

$$\mathbf{S}_{\text{HF}} = \mathbf{Q}_{\text{HF}} \mathbf{R} \quad (5.18)$$

Where  $\mathbf{Q}_{\text{HF}} = [\phi_{\text{HF}}^1, \phi_{\text{HF}}^2, \dots, \phi_{\text{HF}}^r] \in \mathbb{R}^{\mathcal{N}^+}$  and  $\mathbf{R}$  is an upper triangular matrix with diagonal elements as positive that can be obtained by,  $\mathbf{R} = \mathbf{Q}_{\text{HF}}^\top \mathbf{S}_{\text{HF}}$ . Since  $\mathbf{Q}_{\text{HF}}$  is an orthogonal matrix, and  $\mathbf{Q}^\top \mathbf{Q}_{\text{HF}} = \mathbf{I}$ , we can obtain  $\mathbf{Q}_{\text{HF}}^{-1} = \mathbf{Q}_{\text{HF}}^\top$ . The coefficients  $\alpha_n$  is then given by the coefficients of the  $\mathbf{R}$  matrix.

Let us assume that the number of picked adaptive snapshots is three, then the basis functions are given by,

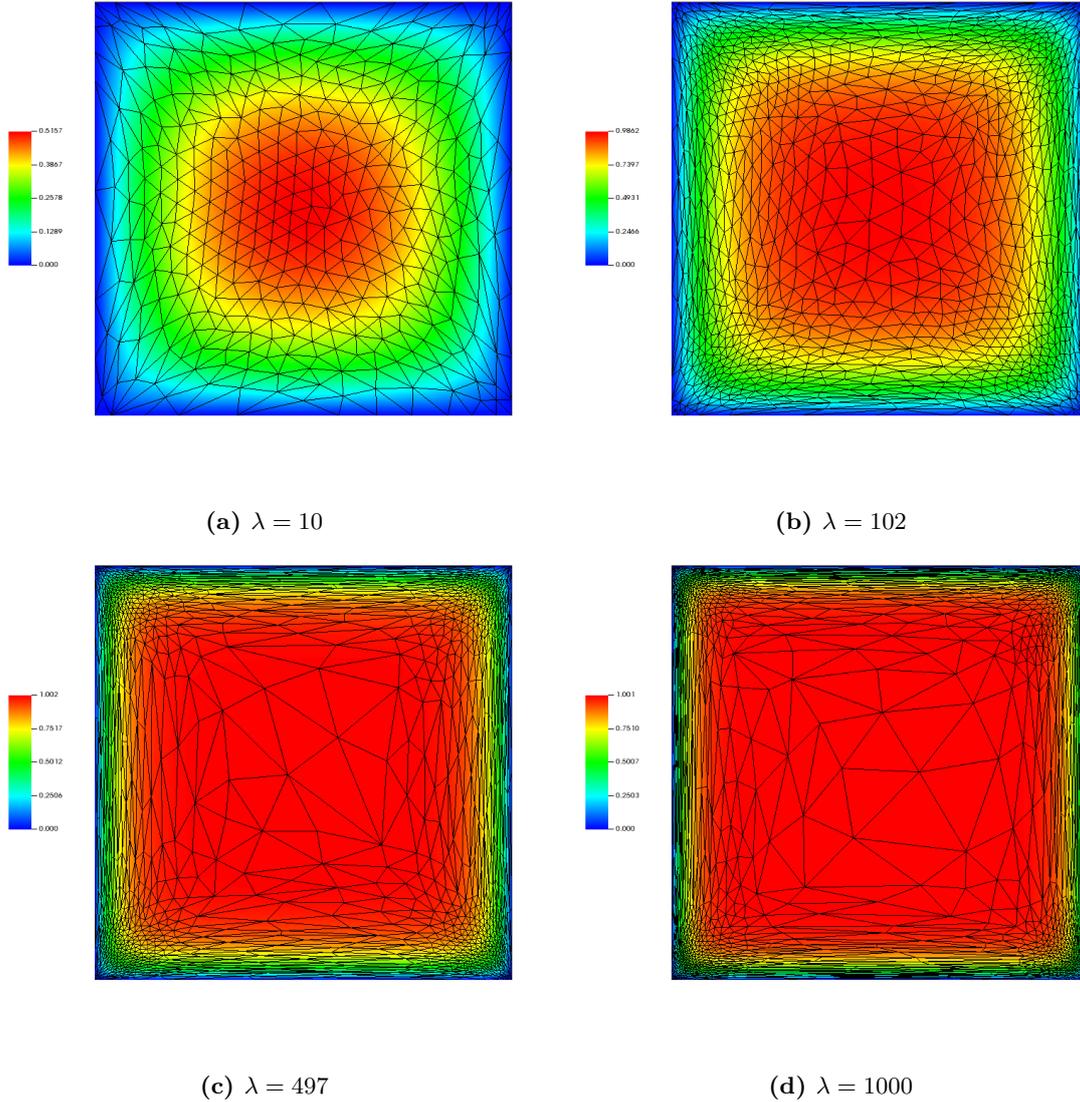
$$\begin{aligned} \tilde{\phi}^1 &= \hat{u}_h^1 \alpha_1 && \in \mathcal{V}_1 \\ \tilde{\phi}^2 &= \hat{u}_h^1 \alpha_1 + \hat{u}_h^2 \alpha_2 && \in \mathcal{V}_{12} \\ \tilde{\phi}^3 &= \hat{u}_h^1 \alpha_1 + \hat{u}_h^2 \alpha_2 + \hat{u}_h^3 \alpha_3 && \in \mathcal{V}_{123} \end{aligned} \quad (5.19)$$

## 5.6 Results and discussions

### 5.6.1 Numerical analysis on mesh adaptation

In this section, we present the FOM solutions of the diffusion-reaction problem explained in section 5.2 generated using adaptive finite elements, and compare the computational performance with the results that are obtained utilizing uniform fine mesh.

Figures 5.5a, 5.5b, 5.5c, and 5.5d represents adaptive FOM solutions for  $\lambda = 10, 102, 497$  and  $\lambda = 1000$  respectively. We can clearly see the difference in the mesh topology among the four different lambdas. The number of nodes to construct the mesh representing  $\lambda = 10$  is 380, for  $\lambda = 102$  is 1030, for  $\lambda = 497$  is, and for  $\lambda = 1000$  is 1284. On the other hand, the uniform fine mesh consisted of order  $\mathcal{O}(10^5)$  nodes, which is 2 orders of magnitude higher than the adaptive mesh representing  $\lambda = 1000$ . Also, the time taken to generate snapshots utilizing uniform fine mesh is of order  $\mathcal{O}(10^3)$  seconds, while using the spatially adaptive mesh takes almost equal to  $\mathcal{O}(10^2)$  seconds. This clearly states the benefit of using adaptive meshes for snapshot generation as compared to using uniform fine mesh for all ranges of lambda values.

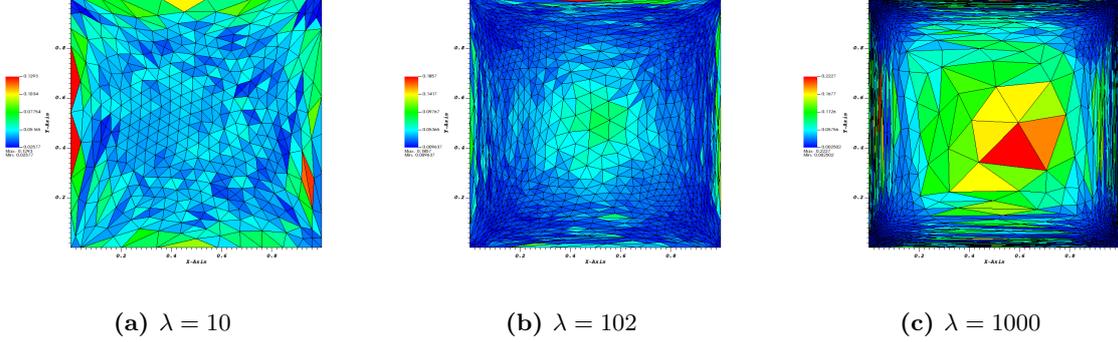


**Figure 5.5:** Representative solution with adaptive meshes

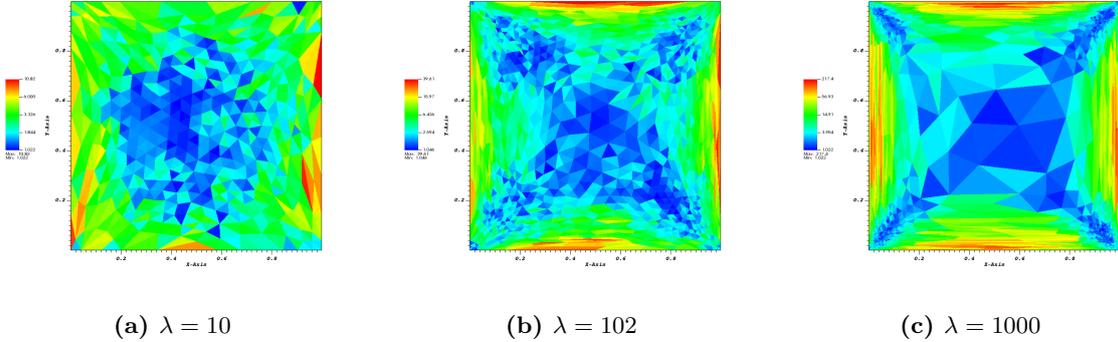
For  $\lambda = 10$ , one can observe from the figure 5.5a, that the mesh is almost uniform across the domain except in the corners and edges which are relatively stretched as compared to the elements in the center which are almost isotropic with minimum mesh size  $h = 0.01$  (see figure 5.6a). On the other hand, for  $\lambda = 102$  (fig. 5.5b), elements in the center are coarser compared to  $\lambda = 10$  and fine across the corners to capture the gradual rising of the slope accurately and stretched elements in the edges. Figure 5.7a-5.7c illustrates the quality of the element given by the stretch factor. The quality of each element is mapped with different colors. The quality is measured by the ratio of the longer edge and the shorter edge, the more stretched an element is, the poorer quality it has.

For lambda values that are rather large, for example with  $\lambda = 1000$  (fig. 5.5d) the mesh representation is drastically uneven since there is very little variation observed in the center, allowing the mesh in the center to be represented by very coarse elements. In contrast, drastic changes are observed along the boundaries causing the mesh size to get minimal in the corners with minimum mesh size  $h = 0.002$  (see figure 5.6c), and very stretched elements along the edges of the domain. One can see from figure 5.7c, that the elements on the diagonals and in the center have good quality elements with almost identical being

isotropic, whereas the elements on the edges are highly stretched out with a ratio as large as 217.



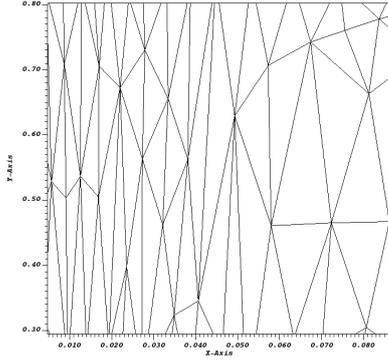
**Figure 5.6:** Illustration of element size for different lambda values.



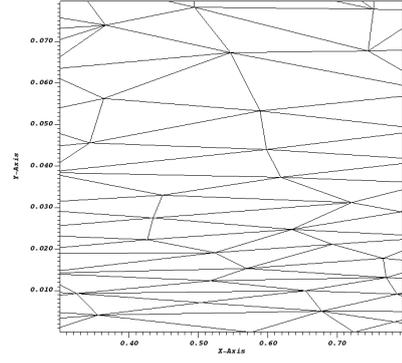
**Figure 5.7:** Illustration of the stretch factor of mesh elements for different lambda values.

To visualize the stretch on the edges of the mesh representing  $\lambda = 1000$ , we plot a zoomed version along the left and bottom edge of the mesh in figures 5.8a and 5.8b respectively. If we look carefully at the solution plot in figure 5.2c at the same spot as a zoomed version (figure 5.8a), it can be observed that the slope is very steep along the  $x$ -direction while remaining almost static in the  $y$ -direction. Hence, according to the mesh adaptation criteria, it naturally adds more elements in the  $x$ -direction, while greatly stretching out the elements in the  $y$ -direction(5.8a). The opposite is true for the zoomed version in the bottom edge of the mesh (figure 5.8b). Hence, the advantage of using anisotropic mesh adaptation that accurately captures the sharp gradient behavior with very few elements, as small as  $\mathcal{O}(10^3)$  all by stretching the elements at a great length along the direction where no change is observed. One needs to recognize that the mesh is well adapted and efficient for this particular lambda, the quality of elements might not be a good representation for other lambda values.

If we were to perform POD in a usual procedure, we would require to interpolate all the snapshots onto the uniform fine mesh that is built by taking into consideration the same fineness as the finest triangle in the adaptive meshes. To overcome this problem, one can interpolate all the snapshots onto a commonly adapted mesh, such that it need not be a super-fine uniform mesh, but a mesh fine enough to represent all the snapshots without losing too many features. Figure 5.9a shows the commonly adapted mesh that is a good representation of all the snapshots belonging to the training set and has 18598 nodes which is smaller by one order of magnitude compared to the uniform mesh. Also, notice the quality of the elements on the boundary is drastically improved after performing the



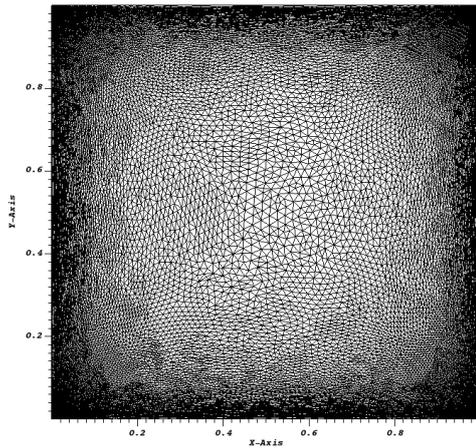
(a) Zoom along the left edge of the mesh in x-direction



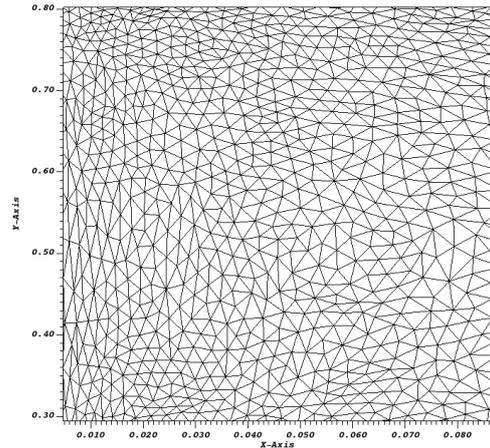
(b) Zoom along the bottom edge of the mesh in y-direction

**Figure 5.8:** Zoom in on the edge of the mesh representing  $\lambda = 1000$ .

mesh adaptation of the snapshots onto this commonly adapted mesh, see figure 5.9b. The stretch factor of the mesh elements is shown in figure 5.10.



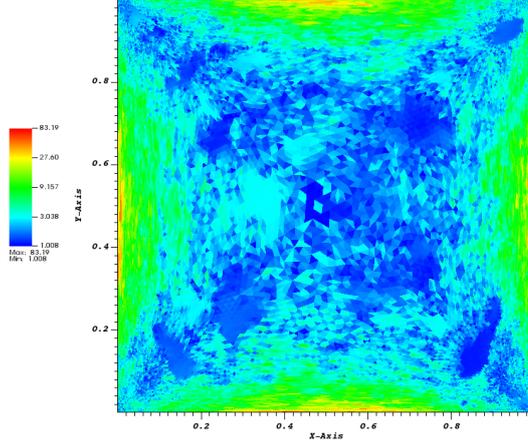
(a)



(b)

**Figure 5.9:** (a) Illustration of commonly adapted mesh ( $\#Nodes = 18598$ ) and (b) zoom in on the left edge of the mesh.

Even if we interpolate the adaptive snapshots onto a commonly adapted mesh and conduct the usual POD procedure to generate basis functions, it still is not cost-effective, especially if the input parametric space is high-dimensional. In such scenarios, the proposed method is indeed quite beneficial as it allows us to sample only the important parametric points without needing to construct any sharp error bounds while maintaining the desired accuracy, and only on the select points, the adaptive snapshots are mapped onto the reference mesh. Note, the objective is to focus on the qualitative performance of the algorithm that can sample points efficiently to build a ROM, so the results that are produced are based on a uniform reference mesh instead of a commonly adapted mesh.



**Figure 5.10:** Stretch factor of the mesh elements of the commonly adapted mesh.

### 5.6.2 Numerical analysis on iterative multi-fidelity reduced-order modeling

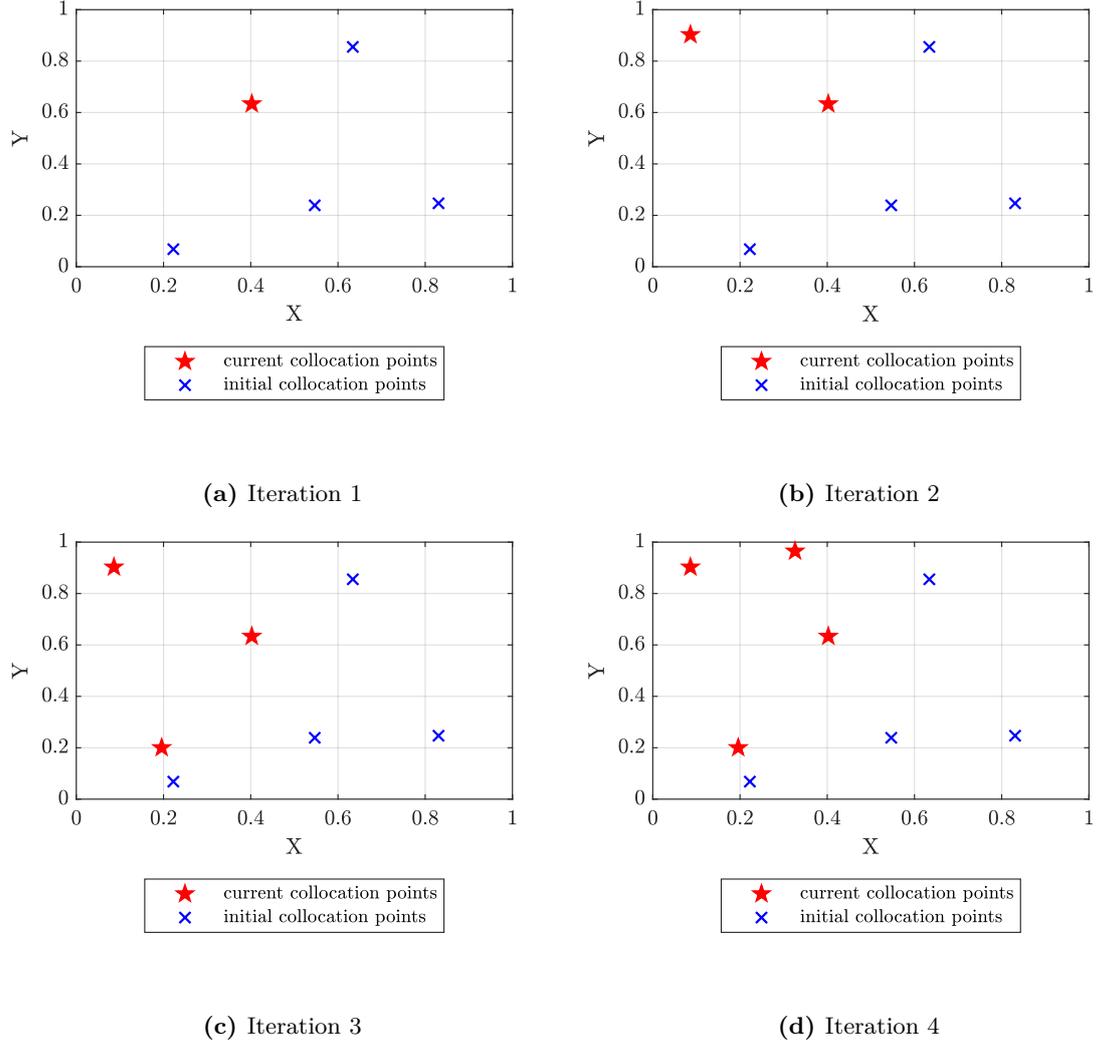
Next, we produce results obtained with the proposed iterative multi-fidelity methodology, as explained in section 5.4. As mentioned previously, we have two ways to update the low-fidelity model approximation:

1. by interpolation of select spatial adaptive snapshots onto a set of computed collocation points of a given reference mesh,
2. by solving a physics-based ROM.

In both cases, the initial low-fidelity model is constructed by interpolating all the adaptive snapshots onto 4 randomly drawn collocation points of the given reference mesh, as shown in figure 5.11a. In other words, the initial rank of the low-fidelity model in both cases is set as four. Since this is a low-rank linear problem, a single parametric point is picked per iteration. Figure 5.11a-5.11d illustrates the current and old collocation points computed during each iteration of the algorithm for case 1. We can observe that as iteration proceeds, the size of collocation points is improved by one. This is due to the enrichment of reduced basis functions, which are also updated by rank one in each iteration. If the  $\ell_2$ -norm error between the reduced basis functions obtained in the previous and current iteration is below a certain tolerance, the algorithm is converged. So, with a target tolerance set to  $\epsilon = 10^{-4}$ , 4 cycles are required to achieve the desired accuracy in both cases, therefore the total rank achieved is 4. We also plot the parametric points that are sampled using both approaches in figure 5.12a and observe that the sampled points are distributed across the parametric space in both cases and none of them lie closer to each other. Thus, we are able to successfully construct the reduced subspace that is spanned by linearly independent basis functions.

In order to verify the accuracy of the algorithm, we plot the POD and ROM error in figure 5.12b for case 2 and compare it with the classical POD and ROM error when evaluated on a fine uniform grid without spatial adaptation. Recall, the POD error is given by the error between the high-fidelity snapshots and their orthogonal projection on the computed POD modes,

$$\epsilon_{\text{POD}} = \frac{\sum_{i=1}^N \|\mathbf{u}_{\text{HF}}^i - \langle \Phi_{\text{HF}}, \mathbf{u}_{\text{HF}}^i \rangle \Phi_{\text{HF}}\|_2^2}{\sum_{i=1}^N \|\mathbf{u}_{\text{HF}}^i\|_2^2} \quad (5.20)$$



**Figure 5.11:** Illustration of collocation points generated at every iteration of the proposed method.

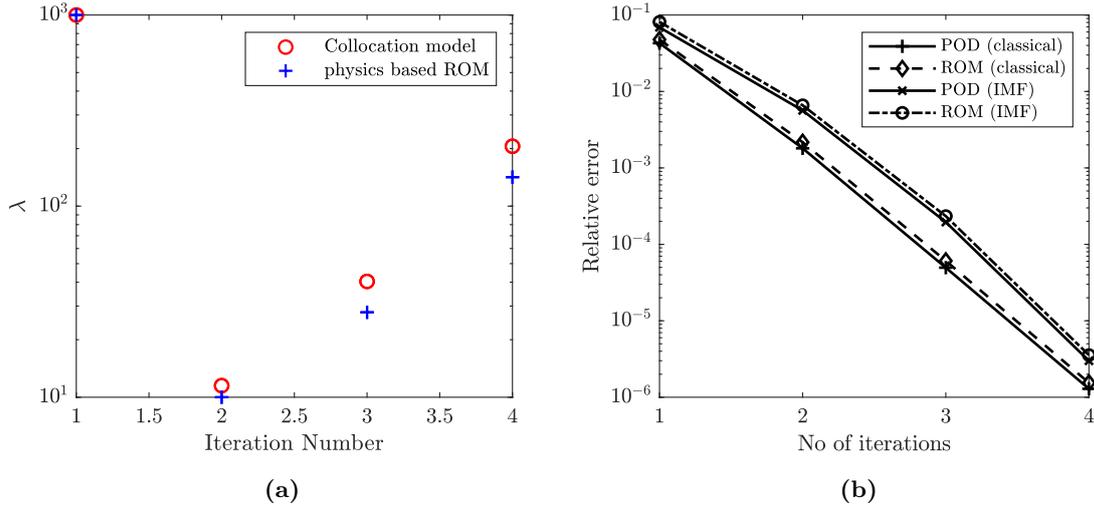
On the other hand, the ROM error is given by the  $\ell_2$ -norm error between the high-fidelity snapshots and the solution of the POD-Galerkin ROM model,

$$\epsilon_{\text{ROM}} = \frac{\sum_{i=1}^N \|\mathbf{u}_{\text{HF}}^i - \mathbf{u}^{\text{R}}(\lambda_i)\|_2^2}{\sum_{i=1}^N \|\mathbf{u}_{\text{HF}}^i\|_2^2} \quad (5.21)$$

We observe that both  $\epsilon_{\text{POD}}$  and  $\epsilon_{\text{POD}}$  have a smooth decay as the rank increases in both the classical POD-ROM without spatial adaptation and iterative multi-fidelity ROM approach with adaptation. The error decays by an order of magnitude  $\mathcal{O}(10^{-6})$  in both cases, while the convergence is attained faster in the regular grid case.

### 5.6.3 Numerical analysis on adaptive reduced basis

Next, we produce results for adaptive basis functions that are generated based on the methodology discussed in 5.5. Since four parametric points are sampled based on the targeted accuracy, we have four sets of commonly adapted meshes represented in figure 5.13. It can be inferred from the plots that each of the meshes is unique and cannot be represented by one another, hence are linearly independent. The minimum size of the element of the first mesh is  $h = 0.002$  (fig. 5.13a) and is a good enough representation for



**Figure 5.12:** (a) Sampled parametric points and (b) Error  $\epsilon_{\text{POD}}$  of the POD projection, and error  $\epsilon_{\text{ROM}}$  of the solution of the reduced order model without space adaptation (classical) and with space adaptation (IMF).

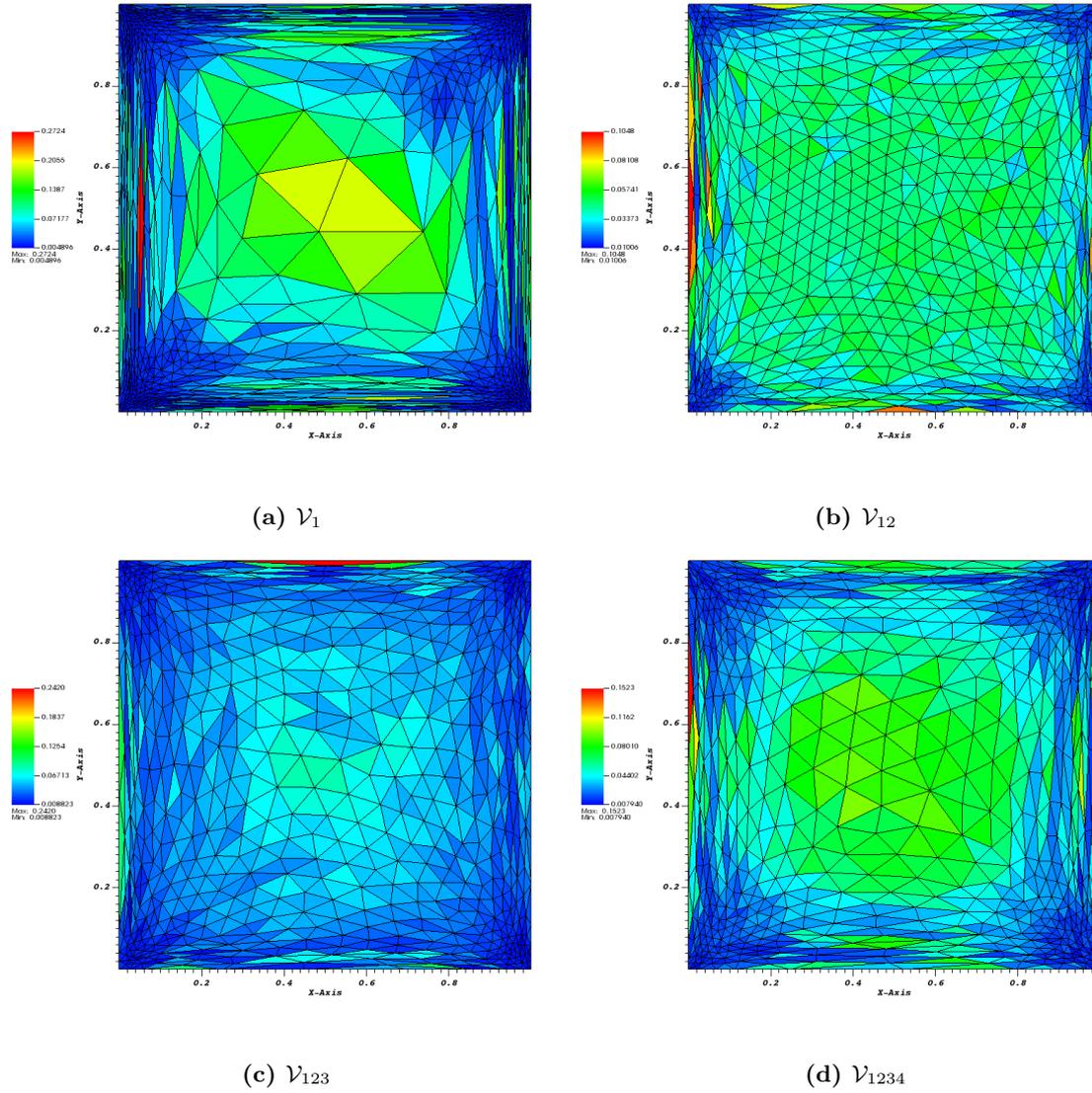
$\lambda = 1000$ . On the other hand, if we look at figure 5.13b, the mesh size is uniform across the domain with an average minimum size of 0.01, which is a good representative mesh for  $\lambda = 10$ . Similarly, if we observe figure 5.13c and 5.13d, the mesh has more elements in the boundaries and edges respectively, which are essential to capture the different slope behavior for other select lambda values. Figures 5.14a-5.17a illustrates the static POD modes that are recovered by the classical approach without mesh adaptation, and figures 5.14b-5.17b represents the POD modes that are recovered by the iterative multi-fidelity approach, and one can observe that the POD modes recovered in both the cases are identical. While figures 5.14c-5.17c represent the spatially adaptive reduced bases obtained by the procedure as explained in section 5.5 and have comparable representation with the previous two.

## 5.7 Conclusions

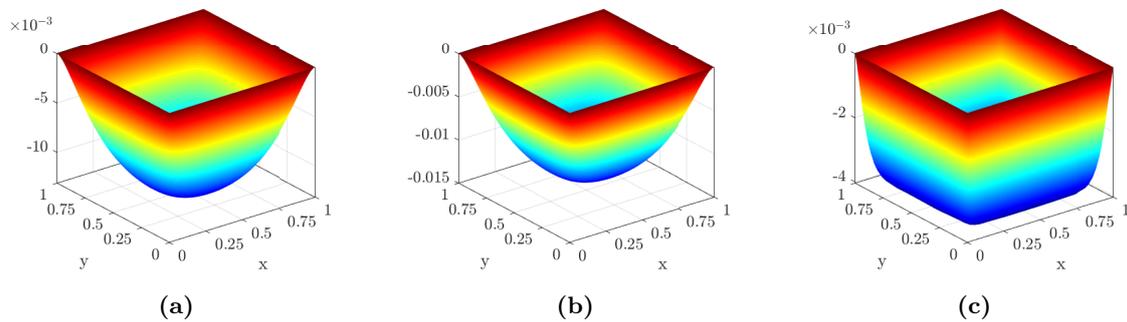
We have extended the framework of the iterative multi-fidelity reduced order modeling approach applied for the high dimensional parametric systems to snapshots that are obtained with adaptive finite elements. The proposed method allows the selection of appropriate space-adapted snapshots from the training set using a low-fidelity model approximation without the requirement of a prior error estimator, and the select snapshots are either mapped onto a fine uniform reference mesh, also termed a high-fidelity model (case 1), or we can directly solve the concerned PDE system using this high fidelity model for the select parameters (case 2).

We proposed a technique to construct an initial low-fidelity model which is a set of collocation points that are randomly drawn from the set of nodes of a given reference mesh, and approximated by mapping the entire snapshots from the training set onto this low-fidelity model. In addition, we presented two ways to update the low-fidelity model approximation.

If case 1, the methodology can be treated as a black box, and we can simply update the low-fidelity model approximation by computing new collocation points using the DEIM strategy. If case 2, we can solve a physics-based ROM with the computed basis functions that are representative of the reference mesh, and therefore update the low fidelity model

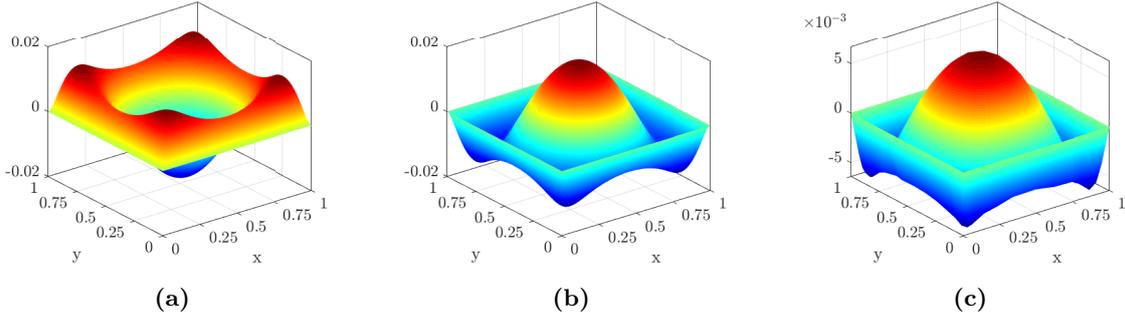


**Figure 5.13:** Illustration of commonly adapted finite element meshes that are representative of a linear combination of 4 different adaptive snapshots.

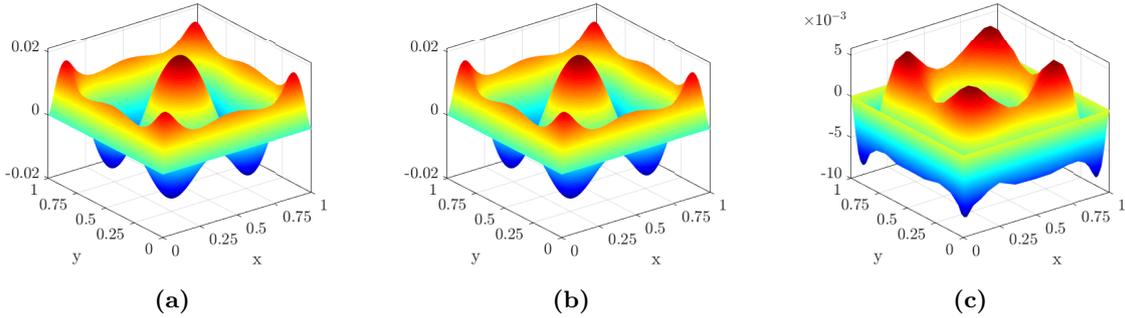


**Figure 5.14:** Mode 1 obtained by classical POD (a), and (b) by iterative multi-fidelity approach. (c) represents spatially adaptive mode 1.

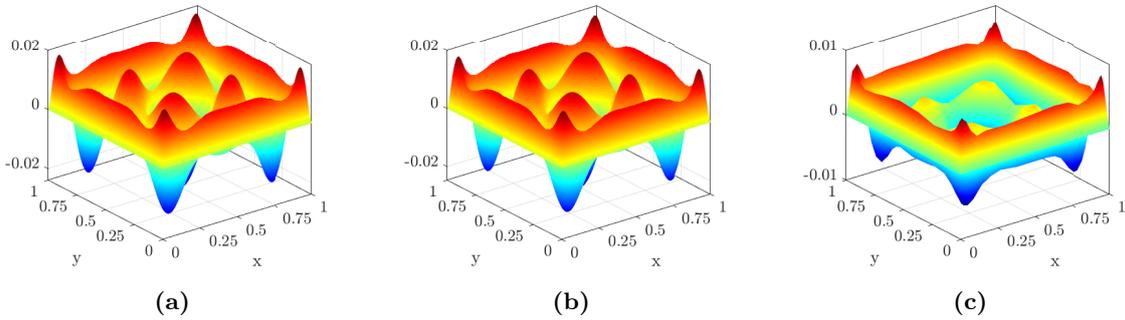
approximation with the computed reduced coefficients. In both cases, we showed the feasibility of the methodology to sample appropriate parametric points. Our findings are underlined with a numerical example of the diffusion-reaction PDE problem. The POD



**Figure 5.15:** Mode 2 obtained by classical POD (a), and (b) by iterative multi-fidelity approach. (c) represents spatially adaptive mode 2.



**Figure 5.16:** Mode 3 obtained by classical POD (a), and (b) by iterative multi-fidelity approach. (c) represents spatially adaptive mode 3.



**Figure 5.17:** Mode 4 obtained by classical POD (a), and (b) by iterative multi-fidelity approach. (c) represents spatially adaptive mode 4.

projection error of our method along with the reduced order modeling error between the high-fidelity snapshots and the POD-G reduced solution is shown to converge when the dimension of the POD is increased.

We also presented a methodology to recover adaptive basis functions that are linear combinations of the select adaptive snapshots. The advantage of this approach is that we need not map all the snapshots lying in the training set to the reference mesh, but rather a select few. Also, the initial low-fidelity model can be of very poor approximation and improve its quality as iteration proceeds until the basis functions obtained are linearly independent.

The mesh adaptation in this work is performed using the IciTech library which is based on an anisotropic framework by h-adaptation, but the proposed method is also applicable to the p-adaptive finite element discretization technique. In addition, we also presented the

benefits of using an anisotropic mesh adaptation technique which may help capture details, to perfectly describe complex issues with the smallest computational cost and computing time.

The future work is to construct a spatially adaptive ROM and combine the dynamic adaptation of the reduced basis with spatially adaptive reduced basis functions within the same framework. By doing so, we can automate the creation of a spatially adaptive ROM for an efficient model order reduction of PDE systems which are discretized using adaptive finite elements.



## Chapter 6

# Conclusions and perspectives

Three important things have been incorporated into this manuscript:

1. Advanced hyper reduction technique based on sparse approximations applied to a complex problem such as parameterized viscoelastic fluid flows of second order,
2. Sampling strategy applied to a multi-dimensional parametric elliptic PDE problem and extended to the complex fluid flow problem without the presence of error estimators,
3. Setting up a POD-ROM on a set of spatially adaptive snapshots, and recovery of spatially adaptive reduced bases.

Firstly, a hyper-reduced framework is successfully implemented to alleviate the performance of model order reduction for inertialess viscoelastic fluid flows. We presented two numerical experiments: flow past a sphere and a 4:1-step flow problem. The algebraic problem is considerably simplified in the ROM formulation. By creating divergence-free velocity approximation spaces, the incompressibility constraint is automatically satisfied in the ROM, which therefore eliminates the pressure degrees of freedom. We also created an intrinsic stabilization of the ROM by including the Schur complement of the velocity gradient unknowns into the reduced momentum balance, which entails ensuring DEVGSS-G stability in the ROM. This step removes the auxiliary velocity gradient unknown from the system. Overall, we reduce the unknowns in the ROM from  $4(\mathbf{u}, p, \mathbf{G}, \mathbf{S})$  to only  $2(\mathbf{u}, \mathbf{S})$ . Nevertheless, the field variables can be always computed in the post-processing step. We also proposed two offline stabilization strategies for online computation of ROM i.e. POD-G and POD-PG to stabilize the advection term in the constitutive equation, and we observed that POD-PG offered a stable ROM of the same accuracy as the expensive offline-online stabilization (POD-SUPG) but at a much reduced computational cost, with a gain in the computational speed-up by a factor of 10. We also noticed that the accuracy of the ROM can be consistently lost when nonlinear terms are reduced using interpolation techniques like DEIM. By adopting a sparse quadrature technique and ensuring the generated quadrature weights are non-negative, this can be prevented. Even though there are more nonlinear function evaluations overall in this approach, this strategy still achieves computational speed up by a factor of  $(10^4)$  as compared to FOM with ROM average errors in the range of  $\mathcal{O}(10^{-2} - 10^{-3})$ . Combining the POD-PG stabilizing approach and the hyper-reduction of the non-linear elements in the benchmark under consideration seemed to strike the right balance between accuracy and complexity without the requirement for online stabilization. Because stabilization concerns the offline stage only, the ROM's complexity is thus released from the constraints of the original FOM, it is plausible to anticipate even greater speedups for 3D flows when adopting the suggested method.

In the problem discussed above, we considered only one input parameter to study the viscoelastic flows and also assumed the dataset to be rich enough to build a stable and accurate ROM. But in engineering applications, we have to deal with multiple parameters, such as changing shape parameters, different geometry, varying shear rates, boundary conditions, etc. It goes without saying that we have to analyze a wide range of parameters to get an accurate ROM to replicate the large-scale system. The computation of high-fidelity models for each sample point is required due to the large sample space. Despite merely having an offline cost, it is nevertheless expensive and should be maintained to a minimum. Thus, we concentrate on achieving our secondary objective, which is to develop an adaptive sampling method that will allow us to find the appropriate set of parametric points for solving complex problems effectively and robustly while greatly reducing the offline cost. So, using an iterative process, we implemented a multi-fidelity approach for reduced basis approximation of a multi-dimensional PDE system. Ad hoc parametric sampling is created from low-fidelity model approximations using the DEIM approach and is based on heuristic assumptions. The reduced bases are then reconstructed using a high-fidelity model.

In terms of implementation, we recommended using either a ROM approximation or solving the PDE with a coarse discretized model to initialize the low-fidelity model approximation. According to the numerical results, the approach has proven to be reliable and stable by converging toward the FOM approximation within the specified tolerance, regardless of the initial quality of the low-fidelity model approximation. Not to mention, this method worked well when the necessary component for effective greedy sampling—an a posteriori error estimator—was not available. The user can choose a single point or multiple points per iteration in the greedy selection. However, care must be taken because it may over-sample parametric points than necessary to obtain the same result. Overall, the proposed strategy needed less high-fidelity snapshot generation compared to classical POD to construct the reduced basis subspace. On a 2D steady-state advection-diffusion problem with 9 input parametric dimensions, this methodology is successfully shown. The suggested approach and the greedy RBM are compared qualitatively for a steady-state heat conduction problem, where both solutions exhibited comparable computational performance. Finally, the algorithm was tested on the flow past sphere benchmark problem suspended in Giesekus fluid with relaxation time as the input fluid parameter, and it was demonstrated that regardless of the PDE definition, the methodology allowed optimal parametric sampling in the absence of problem-specific error estimators.

Lastly, we have included adaptive finite element snapshots in the framework of the iterative multi-fidelity reduced order modeling approach used for the multi-dimensional parametric systems. Without the need for a prior error estimator, the suggested method enabled the selection of appropriate parameters from the training set using a low-fidelity model approximation. We suggested a novel method for creating this low-fidelity model, which is a collection of collocation points selected at random from a set of nodes or integration points of the reference mesh, and then we approximate it by interpolating all the training set’s snapshots onto it. The adaptive snapshots corresponding to the selected parameters are then either mapped onto a fine uniform reference mesh, also known as a high-fidelity model (case 1), or the relevant PDE system is solved using this high-fidelity model for the selected parameters (case 2) to recover the reduced bases. We provided two methods for updating the low-fidelity model approximation. If scenario 1, we can compute new collocation points using the DEIM strategy over the reduced bases and update the low-fidelity model approximation by interpolating all the snapshots onto these newly computed collocation points, therefore treating the methodology as a black box. If scenario 2, we can update the low fidelity model with a physics-based ROM using the computed basis functions and approximate it by solving for the reduced coefficients. In each of the

scenarios, we demonstrated the viability of the methodology to sample suitable parametric points. A numerical illustration of the diffusion-reaction PDE problem highlights our conclusions. The iteration is continued until linear independence between the generated basis functions is achieved. We also showed that the POD projection error of our method converges as the POD basis dimension is increased. The benefit of this method is that just a handful of the space-adapted snapshots from the training set need to be mapped into the reference mesh. In addition, we also presented a methodology in this manuscript for recovering spatial adaptive basis functions, which are just linear combinations of selected adaptive snapshots.

**Perspectives:** By extending the framework presented in this manuscript to three-dimensional flows, where we can also take configurational properties (i.e. position, orientation, and shape of the particle) into account as other variable parameters in the ROM, we hope to lay the groundwork for future research into integrating model order reduction in the full-scale simulations of particle suspensions in complex fluids. Full-scale simulation of DNS is currently not feasible because it requires the flow equations around the particle to be solved at each time step for which the modeling is simplified and conducted on a representative volume with few suspended particles. However, the proposed framework of ROM of viscoelastic fluid flows might allow for a full-scale simulation of the viscoelastic flow without using up all the resources of HPCs while creating a bridge for efficient multi-scale level modeling. We should be able to simplify the computations of microscale modeling for effectively computing the particle kinematics  $(\dot{\mathbf{x}}, \dot{\mathbf{p}})$  as functions of flow conditions suspended in viscoelastic medium with an online efficient ROM. From this information, the second-order orientation tensor  $\mathbf{a}^{\text{DNS}}(t_n)$  can be computed at time  $t_n$  for ' $N_p$ ' number of particles via its discrete definition,

$$\mathbf{a}^{\text{DNS}}(t_n) = \frac{1}{N_p} \sum_{i=1}^{N_p} \mathbf{p}_i^n \otimes \mathbf{p}_i^n$$

On the other hand, this can be incredibly useful for developing an HPC database by researching the fluid dynamics and particle kinematics at a significantly lower offline cost for various aspect ratios of particle shape, different particle orientations, varying flow behavior, and so on.

**Current limitations:** In the current ROM framework for viscoelastic flows, we are able to construct divergence-free approximation spaces for velocity field with inhomogeneous Dirichlet data, however, this approximation ceases to exist for cases when the mesh is not kept fixed or geometrical and shape parameterization is taken into consideration because then each snapshot would be divergence-free in its corresponding configuration.

In particular, let us assume the solution of the velocity field of eq. 3.16 now lie in different approximation space for each  $\lambda \in \Xi_{\text{train}} \subset \mathcal{D}$  parameter, i.e.  $\mathbf{u}_h^j \in \mathcal{V}_j^{N_j}, j = 1, \dots, N$ . It can fulfill divergence-free property only with respect to the corresponding pressure space  $q^j$  :

$$\langle q, \nabla \cdot \mathbf{u}_h^j \rangle = 0 \quad \forall q_h \in \mathcal{P}^j, j = 1 \dots N$$

However,  $\mathbf{u}_h^j$  is not necessarily divergence-free with respect to the other spaces  $\mathcal{P}^i$  for  $i \neq j$ . As a consequence, no divergence-free property can be ensured for arbitrary linear combinations of snapshots. There are two directions we can go: a modified velocity POD basis that is weakly divergence free with respect to a reference mesh can be built. This should enable the cancellation of the pressure term in the flow equation, and the

continuity equation is automatically fulfilled by construction. Alternatively, we can add POD pressure space with the velocity POD space which is augmented by supremizer functions in order to achieve stability in the ROM, which has been studied extensively in the past for Newtonian flows [196, 197]. Additionally, extending the recommended sampling approach to time-dependent PDE systems, which would include time as an additional input parameter, is another area of research that may prove helpful in the future.



# Appendix A

## Lower bound estimation of the coercivity constant

As mentioned in the chapter 2 subsection 2.2.4, the fundamental feature of a certified RBM is the availability of a rigorous error estimator. This step requires the computation of the residual and an estimation of the lower bound for the coercivity or inf-sup constant. Here, we discuss two approaches to estimate the lower bound using multi min- $\theta$  and min- $\theta$  methods [9] for the heat conduction problem discussed in 4.3. Before, let us first recall the definition of the discrete coercivity constant, which is given by,

$$\alpha(\mu) = \inf_{v \in \mathcal{V}^{\mathcal{N}}} \frac{a(v, v; \mu)}{\|v\|_{\mathcal{V}^{\mathcal{N}}}^2} \quad (\text{A.1})$$

The coercivity constant  $\alpha(\mu)$  can then be computed as the smallest eigenvalue of a generalized eigenvalue problem: find  $(\lambda, w) \in \mathbb{R} \times \mathcal{V}^{\mathcal{N}}$  such that,

$$a(w, v; \mu) = \lambda (w, v)_{\mathcal{V}^{\mathcal{N}}}, \quad \forall v, w \in \mathcal{V}^{\mathcal{N}} \quad (\text{A.2})$$

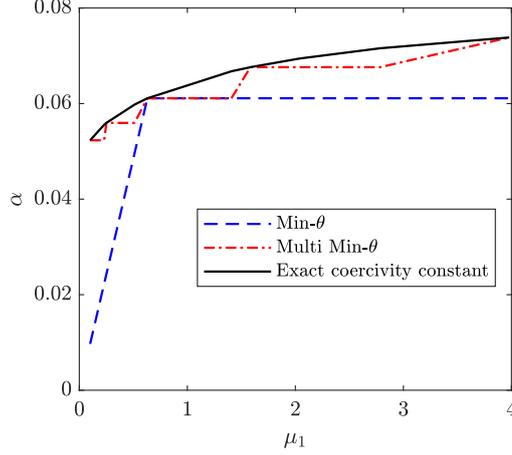
The affine decomposition of the bilinear term in the eq. 4.15 can be expressed as,

$$a(w, v; \mu) = \sum_{q=1}^{Q_a} \theta_a^q(\mu) a_q(w, v)$$

with  $Q_a = 2$  with  $\theta_a^1(\mu) = 1, \theta_a^2(\mu) = \mu_{[1]}$ . Since,  $\theta_a^q(\mu) > 0, \forall \mu \in \mathcal{D}, q = 1, \dots, Q_a$ , the PDE defined is parametrically coercive, and also,  $a_q(\cdot, \cdot) : \mathcal{V}^{\mathcal{N}} \times \mathcal{V}^{\mathcal{N}} \rightarrow \mathbb{R}$  is semi-positive definite for all  $q = 1, \dots, Q_a$ , i.e., the bilinear form is a convex combination of semi-positive bilinear forms. Under this assumption and further assuming that the stability  $\alpha(\mu')$  has been computed for a single parameter value  $\mu'$ , we observe the following identity holds true,

$$\begin{aligned} \alpha(\mu) &= \inf_{v \in \mathcal{V}^{\mathcal{N}}} \sum_{q=1}^{Q_a} \theta_a^q(\mu) \frac{a_q(w, v)}{\|v\|_{\mathcal{V}^{\mathcal{N}}}^2} \\ &= \inf_{v \in \mathcal{V}^{\mathcal{N}}} \sum_{q=1}^{Q_a} \frac{\theta_a^q(\mu)}{\theta_a^q(\mu')} \theta_a^q(\mu') \frac{a_q(w, v)}{\|v\|_{\mathcal{V}^{\mathcal{N}}}^2} \end{aligned} \quad (\text{A.3})$$

The lower bound for the min- $\theta$  approach can then be derived as,



**Figure A.1:** Illustration of the estimated lower bounds for the coercivity constant, for example presented in section 4.3.

$$\begin{aligned}
\alpha(\mu) &\geq \inf_{v \in \mathcal{V}^{\mathcal{N}}} \min_{q=1, \dots, Q_a} \frac{\theta_a^q(\mu)}{\theta_a^q(\mu')} \sum_{q=1}^{Q_a} \theta_a^q(\mu') \frac{a_q(v, v)}{\|v\|_{\mathcal{V}^{\mathcal{N}}}^2} \\
&= \min_{q=1, \dots, Q_a} \frac{\theta_a^q(\mu)}{\theta_a^q(\mu')} \underbrace{\inf_{v \in \mathcal{V}^{\mathcal{N}}} \sum_{q=1}^{Q_a} \theta_a^q(\mu') \frac{a_q(v, v)}{\|v\|_{\mathcal{V}^{\mathcal{N}}}^2}}_{\alpha(\mu')} = \alpha(\mu') \min_{q=1, \dots, Q_a} \frac{\theta_a^q(\mu)}{\theta_a^q(\mu')} =: \alpha_{\text{LB}}(\mu).
\end{aligned} \tag{A.4}$$

Note, we can efficiently compute  $\alpha(\mu')$  in the offline stage by computing the lowest eigenvalue of A.2. Although this method gives a positive lower bound for the value of  $\alpha(\mu)$ , it is often not a sharp bound, which could lead to error bounds that are too conservative.

On the other hand, in the multi min- $\theta$  technique, we discover the highest value over a range of multi-parameter values rather than estimating the lower bound based on a single parameter. By defining,  $\alpha(\mu'_p)$  for a set of ‘ $p$ ’ multi-parameter values  $\mu'_p = \{\mu_{[1]}^1, \mu_{[1]}^2, \dots, \mu_{[1]}^p\} \in \mathcal{D}$  and by following the same approach as min-theta, the lower bound is derived as,

$$\begin{aligned}
\alpha(\mu) &\geq \max_{\mu'_p} \left( \min_{q=1, \dots, Q_a} \frac{\theta_a^q(\mu)}{\theta_a^q(\mu'_p)} \underbrace{\inf_{v \in \mathcal{V}^{\mathcal{N}}} \sum_{q=1}^{Q_a} \theta_a^q(\mu'_p) \frac{a_q(w, v)}{\|v\|_{\mathcal{V}^{\mathcal{N}}}^2}}_{\alpha(\mu'_p)} \right) \\
&= \max_{\mu'_p} \left( \alpha(\mu'_p) \min_{q=1, \dots, Q_a} \frac{\theta_a^q(\mu)}{\theta_a^q(\mu'_p)} \right) =: \alpha_{\text{LB}}(\mu).
\end{aligned} \tag{A.5}$$

In the offline phase, we can easily compute the term  $\alpha(\mu'_p)$  for all  $p$  by finding the smallest eigenvalue from equation A.2. Then in the online stage, we can compute the lower bound  $\alpha_{\text{LB}}(\mu)$  for any  $\mu$ . While this approach is a relatively accurate one, but it requires more solutions to eigenvalue problems.

So, now let us assume the multi-parameter values be  $\{\mu_{[1]}^1, \dots, \mu_{[1]}^5\} = \{0.1000, 0.2508, 0.6303, 1.5837, 3.9796\}$ . Then the lower bound by multi min- $\theta$  approach is given by,

$$\alpha_{\text{LB}}(\mu) = \max_{p=1, \dots, 5} \left( \alpha(\mu'_p) \min \left( 1, \frac{\mu}{\mu'_p} \right) \right).$$

and for  $\min\text{-}\theta$ , we assume  $\mu' = 0.6303$ , then the lower bound is,

$$\alpha_{\text{LB}}(\mu) = \left( \alpha(\mu') \min \left( 1, \frac{\mu}{\mu'} \right) \right).$$

The results of the estimated lower bounds and the value of the exact coercivity constant are illustrated in figure [A.1](#). In particular, we see that the exact discrete coercivity constant is recovered by lower bounds at the sampled points.



# Bibliography

- [1] J Murali Krishnan, Abhijit P Deshpande, and PB Sunil Kumar. *Rheology of complex fluids*. Springer, 2010.
- [2] Theo van de Ven. The flow of suspensions. *Polymer Composites*, 6:209 – 214, 2004.
- [3] Neelesh A. Patankar and Howard H. Hu. Rheology of a suspension of particles in viscoelastic fluids. *Journal of Non-Newtonian Fluid Mechanics*, 96:427–443, 2001.
- [4] Eric SG Shaqfeh. On the rheology of particle suspensions in viscoelastic fluids. *AIChE Journal*, 65(5):e16575, 2019.
- [5] J. Siepmann and F. Siepmann. Mathematical modeling of drug delivery. *International Journal of Pharmaceutics*, 364:328–343, 2008.
- [6] Anjali Patil and Michael S Ferritto. Polymers for personal care and cosmetics: Overview. *Polymers for personal care and cosmetics*, pages 3–11, 2013.
- [7] R Prabhu and A Devaraju. Recent review of tribology, rheology of biodegradable and fdm compatible polymers. *Materials Today: Proceedings*, 39:781–788, 2021.
- [8] N Sakai, H Yang, and M Watanabe. Theoretical analysis of the shrinkage deformation in viscoelastic food during drying. *Japan Journal of Food Engineering (Japan)*, 2002.
- [9] Jan S Hesthaven, Gianluigi Rozza, and Benjamin Stamm. *Certified reduced basis methods for parametrized partial differential equations*, volume 590. Springer, 2016.
- [10] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2015.
- [11] Christophe Binetruy, Francisco Chinesta, and Roland Keunings. *Flows in Polymers, Reinforced Polymers and Composites*. Springer, 2015.
- [12] George Barker Jeffery. The motion of ellipsoidal particles immersed in a viscous fluid. *Proceedings of the Royal Society of London. Series A, Containing papers of a mathematical and physical character*, 102:161–179, 1922.
- [13] Christopher JS Petrie. The rheology of fibre suspensions. *Journal of Non-Newtonian Fluid Mechanics*, 87:369–402, 1999.
- [14] Rabih Mezher, M Perez, A Scheuer, E Abisset-Chavanne, Francisco Chinesta, and Roland Keunings. Analysis of the folgar & tucker model for concentrated fibre suspensions in unconfined and confined shear flows via direct numerical simulation. *Composites Part A: Applied Science and Manufacturing*, 91:388–397, 2016.

- [15] Nitin Sharma and Neelesh A. Patankar. A fast computation technique for the direct numerical simulation of rigid particulate flows. *Journal of Computational Physics*, 205:439–457, 2005.
- [16] Fanzhou Zhao and Berend G. M. van Wachem. Direct numerical simulation of ellipsoidal particles in turbulent channel flow. *Acta Mechanica*, 224:2331–2358, 2013.
- [17] G. D’Avino, P.L. Maffettone, F. Greco, and M.A. Hulsen. Viscoelasticity-induced migration of a rigid sphere in confined shear flow. *Journal of Non-Newtonian Fluid Mechanics*, 165:466–474, 2010.
- [18] Roland Glowinski, Tsorng-Whay Pan, Todd I. Hesla, and Daniel D. Joseph. A distributed lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow*, 25:755–794, 1999.
- [19] Amin Moosaie and Michael Manhart. A direct numerical simulation method for flow of brownian fiber suspensions in complex geometries. *Journal of Dispersion Science and Technology*, 34:427–440, 2013.
- [20] Aslan Nasirov, Ankit Gupta, Seymour Hasanov, and Ismail Fidan. Three-scale asymptotic homogenization of short fiber reinforced additively manufactured polymer composites. *Composites Part B: Engineering*, 202:108269, 2020.
- [21] Yohan Davit, Christopher G Bell, Helen M Byrne, Lloyd AC Chapman, Laura S Kimpton, Georgina E Lang, Katherine HL Leonard, James M Oliver, Natalie C Pearson, Rebecca J Shipley, et al. Homogenization via formal multiscale asymptotics and volume averaging: How do the two techniques compare? *Advances in Water Resources*, 62:178–206, 2013.
- [22] Robert Byron Bird, Charles F Curtiss, Robert C Armstrong, and Ole Hassager. *Dynamics of polymeric liquids, volume 2: Kinetic theory*. Wiley, 1987.
- [23] M Doi and SF Edwards. *The theory of polymer dynamics*, clarendon press, oxford, 1986.
- [24] Fransisco Folgar and Charles L Tucker III. Orientation behavior of fibers in concentrated suspensions. *Journal of reinforced plastics and composites*, 3:98–119, 1984.
- [25] Francisco Chinesta, Guillaume Chaidron, and Arnaud Poitou. On the solution of fokker–planck equations in steady recirculating flows involving short fiber suspensions. *Journal of Non-Newtonian Fluid Mechanics*, 113:97–125, 2003.
- [26] Amine Ammar and Francisco Chinesta. A particle strategy for solving the fokker–planck equation modelling the fiber orientation distribution in steady recirculating flows involving short fiber suspensions. In *Meshfree Methods for Partial Differential Equations II*, pages 1–15. Springer, 2005.
- [27] Camilo Cruz, Lounès Illoul, Francisco Chinesta, and Gilles Regnier. Effects of a bent structure on the linear viscoelastic response of diluted carbon nanotube suspensions. *Rheologica Acta*, 49:1141–1155, 2010.
- [28] Cédric Chauvière and Alexei Lozinski. Simulation of dilute polymer solutions using a fokker–planck equation. *Computers & fluids*, 33:687–696, 2004.

- [29] Alexei Lozinski and Cédric Chauvière. A fast solver for fokker–planck equation applied to viscoelastic flows calculations: 2d fene model. *Journal of Computational Physics*, 189:607–625, 2003.
- [30] J. Férec, M. Heniche, M.C. Heuzey, G. Ausias, and P.J. Carreau. Numerical solution of the fokker–planck equation for fiber suspensions: Application to the folgar–tucker–lipscomb model. *Journal of Non-Newtonian Fluid Mechanics*, 155:20–29, 2008.
- [31] Suresh G. Advani and Charles L. Tucker. The use of tensors to describe and predict fiber orientation in short fiber composites. *Journal of Rheology*, 31:751–784, 1987.
- [32] SG Advani, SF Shuler, et al. Rheology of long fiber-reinforced composites in sheet-forming. In *Composite materials series*, volume 11, pages 323–369. Elsevier, 1997.
- [33] Suresh G. Advani and Charles L. Tucker. Closure approximations for three-dimensional structure tensors. *Journal of Rheology*, 34:367–386, 1990.
- [34] Etienne Pruliere, Amine Ammar, Nadia El Kissi, and Francisco Chinesta. Recirculating flows involving short fiber suspensions: numerical difficulties and efficient advanced micro-macro solvers. *Archives of Computational Methods in Engineering*, 16:1–30, 2009.
- [35] Kunji Chiba, Amine Ammar, and Francisco Chinesta. On the fiber orientation in steady recirculating flows involving short fibers suspensions. *Rheologica acta*, 44:406–417, 2005.
- [36] Shirley J. Johnson, Andrew J. Salem, and Gerald G. Fuller. Dynamics of colloidal particles in sheared, non-newtonian fluids. *Journal of Non-Newtonian Fluid Mechanics*, 34:89–121, 1990.
- [37] G d’Avino, PL Maffettone, MA Hulsen, and GWM Peters. Numerical simulation of planar elongational flow of concentrated rigid particle suspensions in a viscoelastic fluid. *Journal of Non-Newtonian Fluid Mechanics*, 150:65–79, 2008.
- [38] Gaetano D’Avino, Pier Luca Maffettone, Martien A Hulsen, and Gerrit WM Peters. A numerical method for simulating concentrated rigid particle suspensions in an elongational flow using a fixed grid. *Journal of Computational Physics*, 226:688–711, 2007.
- [39] Yoichi Iso, Donald L Koch, and Claude Cohen. Orientation in simple shear flow of semi-dilute fiber suspensions 1. weakly elastic fluids. *Journal of Non-Newtonian Fluid Mechanics*, 62:115–134, 1996.
- [40] D.Z. Gunes, R. Scirocco, J. Mewis, and J. Vermant. Flow-induced orientation of non-spherical particles: Effect of aspect ratio and medium rheology. *Journal of Non-Newtonian Fluid Mechanics*, 155:39–50, 2008.
- [41] NO Jaensson, MA Hulsen, and PD Anderson. Direct numerical simulation of particle alignment in viscoelastic fluids. *Journal of Non-Newtonian Fluid Mechanics*, 235:125–142, 2016.
- [42] F Gauthier, HL Goldsmith, and SG Mason. Particle motions in non-newtonian media. ii. poiseuille flow. *Transactions of the Society of Rheology*, 15:297–330, 1971.

- [43] LG Leal. The slow motion of slender rod-like particles in a second-order fluid. *Journal of Fluid Mechanics*, 69:305–337, 1975.
- [44] P Brunn. The slow motion of a rigid particle in a second-order fluid. *Journal of Fluid Mechanics*, 82:529–547, 1977.
- [45] Domenico Borzacchiello, Emmanuelle Abisset-Chavanne, Francisco Chinesta, and Roland Keunings. Orientation kinematics of short fibres in a second-order viscoelastic fluid. *Rheologica Acta*, 55:397–409, 2016.
- [46] Raanan Fattal and Raz Kupferman. Constitutive laws for the matrix-logarithm of the conformation tensor. *Journal of Non-Newtonian Fluid Mechanics*, 123:281–285, 2004.
- [47] Robert Guénette, Abdelmalek Zine, André Fortin, Pierre Carreau, and Miroslav Grmela. Simulation of viscoelastic flows using a conformation tensor model. *Journal of Non-Newtonian Fluid Mechanics*, 45:187–208, 1992.
- [48] Gianluigi Rozza, DB Phuong Huynh, and Andrea Manzoni. Reduced basis approximation and a posteriori error estimation for stokes flows in parametrized geometries: roles of the inf-sup stability constants. *Numerische Mathematik*, 125:115–152, 2013.
- [49] Kazufumi Ito and Sivaguru S Ravindran. A reduced-order method for simulation and control of fluid flows. *Journal of computational physics*, 143:403–425, 1998.
- [50] Lawrence Sirovich. Turbulence and the dynamics of coherent structures. ii. symmetries and transformations. *Quarterly of Applied mathematics*, 45:573–582, 1987.
- [51] Francisco Chinesta, Roland Keunings, and Adrien Leygue. *The proper generalized decomposition for advanced numerical simulations: a primer*. Springer Science & Business Media, 2013.
- [52] Mohammad Hammoud, Marianne Beringhier, and Jean-Claude Grandidier. A reduced simulation applied to the viscoelastic fatigue of polymers. *Comptes Rendus Mécanique*, 342:671–691, 2014.
- [53] Alfio Quarteroni and Gianluigi Rozza. Numerical solution of parametrized navier–stokes equations by reduced basis methods. *Numerical Methods for Partial Differential Equations: An International Journal*, 23:923–948, 2007.
- [54] Martin A Grepl and Anthony T Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 39:157–181, 2005.
- [55] Martin A Grepl, Yvon Maday, Ngoc C Nguyen, and Anthony T Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41:575–605, 2007.
- [56] John Leask Lumley. The structure of inhomogeneous turbulent flows. *Atmospheric turbulence and radio wave propagation*, pages 166–178, 1967.
- [57] Gianluigi Rozza, Dinh Bao Phuong Huynh, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15:1, 2007.

- [58] Anthony T Patera, Gianluigi Rozza, et al. Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations, 2007.
- [59] Immanuel Martini, Gianluigi Rozza, and Bernard Haasdonk. Reduced basis approximation and a-posteriori error estimation for the coupled stokes-darcy system. *Advances in Computational Mathematics*, 41:1131–1157, 2015.
- [60] Imran Akhtar, Jeff Borggaard, and Alexander Hay. Shape sensitivity analysis in flow models using a finite-difference approach. *Mathematical Problems in Engineering*, 2010, 2010.
- [61] Alexander Hay, Jeff Borggaard, Imran Akhtar, and Dominique Pelletier. Reduced-order models for parameter dependent geometries based on shape sensitivity analysis. *Journal of computational Physics*, 229:1327–1352, 2010.
- [62] K Aleksic, R King, BR Noack, O Lehmann, M Morzynski, and G Tadmor. Nonlinear flow control using a low dimensional galerkin model. *Facta Univ. Ser. Autom. Control Robot*, 7:63–70, 2008.
- [63] Bradley Alan Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26:17–32, 1981.
- [64] Hussam Al Daas, Laura Grigori, Pascal Hénon, and Philippe Ricoux. Recycling Krylov subspaces and reducing deflation subspaces for solving sequence of linear systems. Research report, Inria Paris, 2018.
- [65] C. Villemagne and Robert E. Skelton. Model reductions using a projection formulation. *26th IEEE Conference on Decision and Control*, 26:461–466, 1987.
- [66] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57:483–531, 2015.
- [67] Philippe G. Ciarlet. The finite element method for elliptic problems. In *Classics in applied mathematics*, 2002.
- [68] Franco Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers. *Publications mathématiques et informatique de Rennes*, pages 1–26, 1974.
- [69] Gianluigi Rozza and Karen Veroy. On the stability of reduced basis method for stokes equations in parametrized domains. *Computer Methods in Applied Mechanics and Engineering*, 196:1244–1260, 2007.
- [70] K Veroy and Anthony T Patera. Reduced-basis approximation of the viscosity-parametrized incompressible navier-stokes equation: Rigorous a posteriori error bounds. 2004.
- [71] Karen Veroy and Anthony T 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.

- [72] Yvon Maday, Anthony T. Patera, and D. V. Rovas. A blackbox reduced-basis output bound method for noncoercive linear problems. *Studies in Mathematics and Its Applications*, 31:533–569, 2002.
- [73] Gianluigi Rozza and Karen Veroy. On the stability of reduced basis method for stokes equations in parametrized domains. *Computer Methods in Applied Mechanics and Engineering*, 196:1244–1260, 2007.
- [74] Yano, Masayuki. A minimum-residual mixed reduced basis method: Exact residual certification and simultaneous finite-element reduced-basis refinement. *ESAIM: M2AN*, 50:163–185, 2016.
- [75] Wolfgang Dahmen, Christian Plesken, and Gerrit Welper. Double greedy algorithms: Reduced basis methods for transport dominated problems. *ESAIM Mathematical Modelling and Numerical Analysis*, 48, 2013.
- [76] Sebastian Ullmann, Marko Rotkvic, and Jens Lang. POD-Galerkin reduced-order modeling with adaptive finite element snapshots. *Journal of Computational Physics*, 325:244–258, 2016.
- [77] Carmen Gräßle and Michael Hinze. POD reduced-order modeling for evolution equations utilizing arbitrary finite element discretizations. *Advances in Computational Mathematics*, 44:1941–1978, 2018.
- [78] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Elsevier, 2013.
- [79] Lawrence Sirovich and M Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America. A, Optics and image science*, 4:519–24, 1987.
- [80] Rajeev Motwani and Prabhakar Raghavan. Randomized algorithms. *SIGACT News*, 26:48–50, 1995.
- [81] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *J. ACM*, 51:1025–1041, 2004.
- [82] N. Benjamin Erichson, Sergey Voronin, Steven L. Brunton, and J. Nathan Kutz. Randomized matrix decompositions using r. *Journal of Statistical Software*, 89:1–48, 2019.
- [83] Eleni Drinea, Petros Drineas, and Patrick Huggins. A randomized singular value decomposition algorithm for image processing applications. In *Proceedings of the 8th panhellenic conference on informatics*, pages 278–288. Citeseer, 2001.
- [84] Franco Woolfe, Edo Liberty, Vladimir Rokhlin, and Mark Tygert. A fast randomized algorithm for the approximation of matrices. *Applied and Computational Harmonic Analysis*, 25:335–366, 2008.
- [85] Oleg Balabanov. *Randomized linear algebra for model order reduction*. Theses, École centrale de Nantes ; Universitat politècnica de Catalunya, 2019.
- [86] T. Braconnier, M. Ferrier, J.-C. Jouhaud, M. Montagnac, and P. Sagaut. Towards an adaptive pod/svd surrogate model for aeronautic design. *Computers Fluids*, 40:195–209, 2011.

- [87] C.G. Baker, K.A. Gallivan, and P. Van Dooren. Low-rank incremental methods for computing dominant singular subspaces. *Linear Algebra and its Applications*, 436:2866–2888, 2012.
- [88] Matthew Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear algebra and its applications*, 415:20–30, 2006.
- [89] Vinita Vasudevan and M. Ramakrishna. A hierarchical singular value decomposition algorithm for low rank matrices. *ArXiv*, abs/1710.02812, 2017.
- [90] Benjamin Peherstorfer and Karen Willcox. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering*, 291:21–41, 2015.
- [91] Richard Everson and Lawrence Sirovich. Karhunen–loève procedure for gappy data. *JOSA A*, 12, 1995.
- [92] K. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers Fluids*, 35:208–226, 2006.
- [93] Nissrine Akkari, Fabien Casenave, and David Ryckelynck. A novel gappy reduced order method to capture non-parameterized geometrical variation in fluid dynamics problems. 2019.
- [94] Basang Tsering-xiao and Qinwu Xu. Gappy pod-based reconstruction of the temperature field in tibet. *Theoretical and Applied Climatology*, 138, 2019.
- [95] Christopher A. Beattie, Jeff Borggaard, Serkan Gugercin, and Traian Iliescu. A domain decomposition approach to pod. *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 6750–6756, 2006.
- [96] Zhu Wang, Brian Mcbee, and Traian Iliescu. Approximate partitioned method of snapshots for pod. *Journal of Computational and Applied Mathematics*, 307, 2015.
- [97] Annalisa Buffa, Yvon Maday, Anthony Patera, Christophe Prud’homme, and Gabriel Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46, 2012.
- [98] Yanlai Chen, Jan S Hesthaven, Yvon Maday, and Jerónimo Rodríguez. Improved successive constraint method based a posteriori error estimate for reduced basis approximation of 2d maxwell’s problem. *ESAIM: Mathematical Modelling and Numerical Analysis*, 43:1099–1116, 2009.
- [99] Yanlai Chen. A certified natural-norm successive constraint method for parametric inf-sup lower bounds. *Applied Numerical Mathematics*, 99:98–108, 2016.
- [100] Dinh Bao Phuong Huynh, Gianluigi Rozza, Sugata Sen, and Anthony T Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *Comptes Rendus Mathématique*, 345:473–478, 2007.
- [101] David Ryckelynck. Hyper-reduction of mechanical models involving internal variables. *International Journal for Numerical Methods in Engineering*, 77:75–89, 2009.
- [102] Maxime Barrault, Yvon Maday, Ngoc Cuong Nguyen, and Anthony T Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique*, 339:667–672, 2004.

- [103] Ngoc-Cuong Nguyen, Anthony T Patera, and Jaime Peraire. A ‘best points’ interpolation method for efficient approximation of parametrized functions. *International journal for numerical methods in engineering*, 73:521–543, 2008.
- [104] J.A. Hernández, 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.
- [105] Masayuki Yano and Anthony 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.
- [106] Patricia Astrid, Siep Weiland, Karen Willcox, and Ton Backx. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control*, 53:2237–2251, 2008.
- [107] Kevin Carlberg, Charbel Bou-Mosleh, and Charbel Farhat. Efficient non-linear model reduction via a least-squares petrov-galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86:155 – 181, 2011.
- [108] Saifon Chaturantabut and Danny C Sorensen. NONLINEAR MODEL REDUCTION VIA DISCRETE EMPIRICAL INTERPOLATION \*. *Society for Industrial and Applied Mathematics*, 32:2737–2764, 2010.
- [109] Charbel Farhat, Todd Chapman, and Philip 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*, 102:1077–1110, 2015.
- [110] Medhat A. Rakha. On the moore–penrose generalized inverse matrix. *Applied Mathematics and Computation*, 158:185–200, 2004.
- [111] Balas K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24:227–234, 1995.
- [112] Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73:273–282, 2011.
- [113] Mattia Manucci, Jose Vicente Aguado, and Domenico Borzacchiello. Sparse data-driven quadrature rules via lp-quasi-norm minimization. *Computational Methods in Applied Mathematics*, 22:389–411, 2022.
- [114] Irina F Gorodnitsky and Bhaskar D Rao. Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. *IEEE Transactions on signal processing*, 45:600–616, 1997.
- [115] Alexander Morozov and Saverio E. Spagnolie. *Introduction to Complex Fluids*, pages 3–52. Springer New York, New York, NY, 2015.
- [116] Marcel J Crochet, Arthur Russell Davies, and Kenneth Walters. *Numerical simulation of non-Newtonian flow*. Elsevier, 2012.

- [117] Mitsuhiro Ohta, Eiji Iwasaki, Eiji Obata, and Yutaka Yoshida. A numerical study of the motion of a spherical drop rising in shear-thinning fluid systems. *Journal of Non-Newtonian Fluid Mechanics*, 116:95–111, 2003.
- [118] Weijie K Zhu. Numerical analysis of shear thickening fluids for blast mitigation applications. Technical report, NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 2011.
- [119] Alexis Aposporidis. *Numerical Analysis of Mixed Formulations for Bingham Fluids*. PhD thesis, Emory University, 2012.
- [120] Rod Cross. Elastic and viscous properties of silly putty. *American Journal of Physics*, 80:870–875, 2012.
- [121] R.R. Huilgol. On the concept of the Deborah number. *Journal of Rheology*, 19:297–306, 1975.
- [122] Robert Poole. The Deborah and Weissenberg numbers. *The British Society of Rheology - Rheology Bulletin*, 53:32–39, 2012.
- [123] Wilco Verbeeten, Arjen Bogaerds, Gerrit Peters, and Frank Baaijens. Viscoelastic analysis of polymer melts in complex flows. *ACM Journal of Experimental Algorithms - JEA*, 1999.
- [124] Frank P.T. Baaijens, Sjaak H.A. Selen, Hans P.W. Baaijens, Gerrit W.M. Peters, and Han E.H. Meijer. Viscoelastic flow past a confined cylinder of a low density polyethylene melt. *Journal of Non-Newtonian Fluid Mechanics*, 68:173–203, 1997.
- [125] Kristian Ejlebjerg Jensen, Peter Szabó, and Fridolin Okkels. Implementation of the log-conformation formulation for two-dimensional viscoelastic flow. *arXiv preprint arXiv:1508.01041*, 2015.
- [126] Yue Mu, Guoqun Zhao, Xianghong Wu, and Jiqiang Zhai. Modeling and simulation of three-dimensional planar contraction flow of viscoelastic fluids with ptt, Giesekus and FENE-P constitutive models. *Applied Mathematics and Computation*, 218(17):8429–8443, 2012.
- [127] B Purnode and MJ Crochet. Polymer solution characterization with the FENE-P model. *Journal of non-newtonian fluid mechanics*, 77(1-2):1–20, 1998.
- [128] Hanswalter Giesekus. A simple constitutive equation for polymer fluids based on the concept of deformation-dependent tensorial mobility. *Journal of Non-Newtonian Fluid Mechanics*, 11(1-2):69–109, 1982.
- [129] Nhan Phan-Thien. A nonlinear network viscoelastic model. *Journal of Rheology*, 22:259–283, 1978.
- [130] Jingjing Chen, Dongxu Han, Bo Yu, Dongliang Sun, and Jinjia Wei. A pod-galerkin reduced-order model for isotropic viscoelastic turbulent flow. *International Communications in Heat and Mass Transfer*, 84:121–133, 2017.
- [131] Yue Wang, Hanghang Ma, Weihua Cai, Hongna Zhang, Jianping Cheng, and Xin Zheng. A pod-galerkin reduced-order model for two-dimensional Rayleigh-Bénard convection with viscoelastic fluid. *International Communications in Heat and Mass Transfer*, 117:104747, 2020.

- [132] Dilip Rajagopalan, Robert C. Armstrong, and Robert A. Brown. Finite element methods for calculation of steady, viscoelastic flow using constitutive equations with a newtonian viscosity. *Journal of Non-Newtonian Fluid Mechanics*, 36:159 – 192, 1990.
- [133] Robert Guénette and Michel Fortin. A new mixed finite element method for computing viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, 60:27 – 52, 1995.
- [134] Michel Fortin and André Fortin. A new approach for the fem simulation of viscoelastic flows. *Journal of Non-Newtonian Fluid Mechanics*, 32:295 – 310, 1989.
- [135] G. D’Avino and M. A. Hulsen. Decoupled second-order transient schemes for the flow of viscoelastic fluids without a viscous solvent contribution. *Journal of Non-Newtonian Fluid Mechanics*, 165:1602–1612, 2010.
- [136] Alexander N. Brooks and Thomas J.R. Hughes. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199 – 259, 1982.
- [137] MA Hulsen, APG Van Heel, and BHAA Van Den Brule. Simulation of viscoelastic flows using brownian configuration fields. *Journal of Non-Newtonian Fluid Mechanics*, 70:79–101, 1997.
- [138] Timothy Nigel Phillips and AJ Williams. Comparison of creeping and inertial flow of an oldroyd b fluid through planar and axisymmetric contractions. *Journal of Non-Newtonian Fluid Mechanics*, 108:25–47, 2002.
- [139] Mostafa Abbaszadeh and Mehdi Dehghan. Reduced order modeling of time-dependent incompressible navier–stokes equation with variable density based on a local radial basis functions-finite difference (lrbf-fd) technique and the pod/deim method. *Computer Methods in Applied Mechanics and Engineering*, 364:112914, 2020.
- [140] D. Xiao, F. Fang, A.G. Buchan, C.C. Pain, I.M. Navon, J. Du, and G. Hu. Non-linear model reduction for the navier–stokes equations using residual deim method. *Journal of Computational Physics*, 263:1–18, 2014.
- [141] Martin Isoz. Pod-deim based model order reduction for speed-up of flow parametric studies. *Ocean Engineering*, 186:106083, 2019.
- [142] Miguel Fosas de Pando, Peter J. Schmid, and Denis Sipp. Nonlinear model-order reduction for compressible flow solvers using the discrete empirical interpolation method. *Journal of Computational Physics*, 324:194–209, 2016.
- [143] Annika Radermacher and Stefanie Reese. Pod-based model reduction with empirical interpolation applied to nonlinear elasticity. *International Journal for Numerical Methods in Engineering*, 107:477 – 495, 2016.
- [144] Paolo Pacciarini and Gianluigi Rozza. Stabilized reduced basis method for parametrized advection–diffusion pdes. *Computer Methods in Applied Mechanics and Engineering*, 274:1–18, 2014.
- [145] SHAFQAT ALI, Francesco Ballarin, and Gianluigi Rozza. Stabilized reduced basis methods for parametrized steady stokes and navier–stokes equations. *Computers Mathematics with Applications*, 80, 2020.

- [146] Arjen C.B. Bogaerds, Anne M. Grillet, Gerrit W.M. Peters, and Frank P.T. Baaijens. Stability analysis of polymer shear flows using the extended pom–pom constitutive equations. *Journal of Non-Newtonian Fluid Mechanics*, 108:187 – 208, 2002. Numerical Methods Workshop S.I.
- [147] Finite Element Method, Weighted Residuals, Galerkin Method, and Weighted Residuals. Chapter 2 Method of Weighted Residuals. pages 1–12.
- [148] Martien a Hulsen. TFEM A toolkit for the finite element method User ’ s Guide. 2009.
- [149] Lawrence F Shampine. Computer solution of ordinary differential equations. *The initial value problem*, 1975.
- [150] Lawrence F Shampine and Mark W Reichelt. The matlab ode suite. *SIAM journal on scientific computing*, 18:1–22, 1997.
- [151] Geuzaine, Christophe and Remacle, Jean-Francois. Gmsh.
- [152] Annalisa Buffa, Yvon Maday, Anthony Patera, Christophe Prud’homme, and Gabriel Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46, 2012.
- [153] Jon C Helton and Freddie Joe Davis. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety*, 81:23–69, 2003.
- [154] Jon C Helton, FJ Davis, and Jay D Johnson. A comparison of uncertainty and sensitivity analysis results obtained with random and latin hypercube sampling. *Reliability Engineering & System Safety*, 89:305–330, 2005.
- [155] James E Gentle. *Random number generation and Monte Carlo methods*. Springer Science & Business Media, 2006.
- [156] Martin Alexander Grepl. *Reduced-basis approximation a posteriori error estimation for parabolic partial differential equations*. PhD thesis, Massachusetts Institute of Technology, 2005.
- [157] Luca Dede. Reduced basis method and a posteriori error estimation for parametrized linear-quadratic optimal control problems. *SIAM Journal on Scientific Computing*, 32:997–1019, 2010.
- [158] NC Nguyen, Gianluigi Rozza, DB Phuong Huynh, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for parametrized parabolic pdes: Application to real-time bayesian parameter estimation. *Large-Scale Inverse Problems and Quantification of Uncertainty*, pages 151–177, 2010.
- [159] Cohen, Albert, Dahmen, Wolfgang, DeVore, Ronald, and Nichols, James. Reduced basis greedy selection using random training sets. *ESAIM: M2AN*, 54:1509–1524, 2020.
- [160] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 30:3270–3288, 2008.

- [161] T. Bui-Thanh, K. Willcox, O. Ghattas, and B. van Bloemen Waanders. Goal-oriented, model-constrained optimization for reduction of large-scale systems. *Journal of Computational Physics*, 224:880–896, 2007.
- [162] K. Willcox, O. Ghattas, B. van Bloemen Waanders, and B. Bader. An optimization frame work for goal-oriented, model-based reduction of large-scale systems. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 2265–2271, 2005.
- [163] Khac Chi Hoang, Pierre Kerfriden, Boo Cheong Khoo, and S.P.A. Bordas. An efficient goal-oriented sampling strategy using reduced basis method for parametrized elastodynamic problems. *Numerical Methods for Partial Differential Equations*, 31, 2015.
- [164] M. A. Bazaz, S. A. Nahve, M. Nabi, S. Janardhanan, and M. U. Rehman. Adaptive parameter space sampling in matrix interpolatory pmor. In *2015 International Conference on Recent Developments in Control, Automation and Power Engineering (RDCAPE)*, pages 83–89, 2015.
- [165] Maria Cruz Varona, Boris Lohmann, et al. Automatic adaptive sampling in parametric model order reduction by matrix interpolation. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 472–477. IEEE, 2017.
- [166] David Amsallem. *Interpolation on manifolds of CFD-based fluid and finite element-based structural reduced-order models for on-line aeroelastic predictions*. Stanford University, 2010.
- [167] Bradley N Bond and Luca Daniel. A piecewise-linear moment-matching approach to parameterized model-order reduction for highly nonlinear systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26:2116–2129, 2007.
- [168] Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization, 2018.
- [169] Alexander Forrester, Andras Sobester, and Andy Keane. Multi-fidelity optimization via surrogate modelling. *Proc. R. Soc. A*, 463:3251–3269, 2007.
- [170] Leo W. T. Ng and Karen E. Willcox. Multifidelity approaches for optimization under uncertainty. *International Journal for Numerical Methods in Engineering*, 100:746–772, 2014.
- [171] X.Q. Wang, P. Song, and M.P. Mignolet. Applications of multifidelity reduced order modeling to single and multiphysics nonlinear structural problems. *Applications in Engineering Science*, 5:100035, 2021.
- [172] Stefan Heinrich. Multilevel monte carlo methods. In Svetozar Margenov, Jerzy Waśniewski, and Plamen Yalamov, editors, *Large-Scale Scientific Computing*, pages 58–67, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
- [173] Michael Penwarden, Shandian Zhe, Akil Narayan, and Robert M. Kirby. Multifidelity modeling for physics-informed neural networks (pinns). *Journal of Computational Physics*, 451:110844, 2022.

- [174] Mengwu Guo, Andrea Manzoni, Maurice Amendt, Paolo Conti, and Jan S. Hesthaven. Multi-fidelity regression using artificial neural networks: Efficient approximation of parameter-dependent output quantities. *Computer Methods in Applied Mechanics and Engineering*, 389:114378, 2022.
- [175] Mariella Kast, Mengwu Guo, and Jan S. Hesthaven. A non-intrusive multifidelity method for the reduced order modeling of nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, 364:112947, 2020.
- [176] Tianshu Wen and Matthew J Zahr. A globally convergent method to accelerate large-scale optimization using on-the-fly model hyperreduction: application to shape optimization. *arXiv preprint arXiv:2206.09942*, 2022.
- [177] Alessandro Alla, Angela Monti, and Ivonne Sgura. Adaptive pod-deim correction for turing pattern approximation in reaction-diffusion pde systems. *arXiv preprint arXiv:2203.05998*, 2022.
- [178] M.J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82:64–84, 1989.
- [179] John Trangenstein. Multi-scale iterative techniques and adaptive mesh refinement for flow in porous media. *Advances in Water Resources*, 25:1175–1213, 2002.
- [180] C.C. Pain, A.P. Umpleby, C.R.E. de Oliveira, and A.J.H. Goddard. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. *Computer Methods in Applied Mechanics and Engineering*, 190:3771–3796, 2001.
- [181] Andreas Papoutsakis, Sergei S. Sazhin, Steven Begg, Ionut Danaïla, and Francky Luddens. An efficient adaptive mesh refinement (amr) algorithm for the discontinuous galerkin method: Applications for the computation of compressible two-phase flows. *Journal of Computational Physics*, 363:399–427, 2018.
- [182] Darrell Pepper and Xiuling Wang. Comparison of h -, p - and hp -adaptation for convective heat transfer. pages 495–504, 2007.
- [183] Eric de Sturler, Glaucio PAULINO, and Shun WANG. Topology optimization with adaptive mesh refinement. 2012.
- [184] T. Coupez, H. Dignonnet, and R. Ducloux. Parallel meshing and remeshing. *Applied Mathematical Modelling*, 25:153–175, 2000. Dynamic load balancing of mesh-based applications on parallel.
- [185] Mazen Ali, Kristina Steih, and Karsten Urban. Reduced basis methods with adaptive snapshot computations. *Advances in Computational Mathematics*, 43:257–294, 2017.
- [186] Carmen Gräßle, Michael Hinze, Jens Lang, and Sebastian Ullmann. Pod model order reduction with space-adapted snapshots for incompressible flows. *Advances in Computational Mathematics*, 45, 2019.
- [187] Mazen Ali and Karsten Urban. *Reduced Basis Exact Error Estimates with Wavelets*, volume 112, pages 359–367. 2016.
- [188] Fangxin Fang, Christopher Pain, Ionel Navon, Matthew Piggott, G. Gorman, P. Alison, and Antony Goddard. Reduced-order modelling of an adaptive mesh ocean model. *International Journal for Numerical Methods in Fluids*, 59:827 – 851, 2009.

- [189] Oliver Lass. Reduced order modeling and parameter identification for coupled nonlinear PDE systems. 2014.
- [190] T. Coupez. Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing. *Journal of Computational Physics*, 230:2391–2405, 2011.
- [191] Youssef Mesri, Walid Zerguine, Hugues Digonnet, Luisa Silva, and Thierry Coupez. *Dynamic Parallel Adaption for Three Dimensional Unstructured Meshes: Application to Interface Tracking*, pages 195–212. 2008.
- [192] Cyril Gruau and Thierry Coupez. 3d tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric. *Computer Methods in Applied Mechanics and Engineering*, 194:4951–4976, 2005.
- [193] Thierry Coupez. Génération de maillage et adaptation de maillage par optimisation locale. *Revue Européenne des Éléments Finis*, 9:403 – 423, 2000.
- [194] Thierry Coupez. Grandes transformations et remaillage automatique. 1991.
- [195] Frédéric Alauzet, J. Freya, and Bijan Mohammadi. Adaptation de maillages non structurés pour des problèmes instationnaires. *Comptes Rendus Mathématique*, 335, 2002.
- [196] Francesco Ballarin, Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Supremizer stabilization of pod–galerkin approximation of parametrized steady incompressible navier–stokes equations. *International Journal for Numerical Methods in Engineering*, 102:1136–1161, 2015.
- [197] Carmen Gräßle, Michael Hinze, Jens Lang, and Sebastian Ullmann. Pod model order reduction with space-adapted snapshots for incompressible flows. *Advances in Computational Mathematics*, 45:2401–2428, 2019.

**Titre :** Modélisation avancée d'ordre réduit et échantillonnage paramétrique pour les écoulements de fluides non-Newtonian

**Mots clés :** hyper-réduction, stabilisation hors ligne , modèles multi-fidélité, snapshots adaptés à l'espace

**Résumé :** Le sujet de cette thèse porte sur la réduction d'ordre de modèle (MOR) de problèmes d'écoulement non-Newtonian paramétrés qui ont des applications industrielles importantes. Les méthodes traditionnelles de réduction de l'ordre des modèles limitent les performances de calcul de ces problèmes hautement non linéaires, nous suggérons donc une technique d'hyper-réduction avancée basée sur une approximation sparse de l'évaluation des termes non linéaire à complexité réduite. Nous proposons également une stratégie de stabilisation hors ligne pour stabiliser le modèle constitutif dans le modèle d'ordre réduit qui est moins cher à calculer tout en maintenant la précision du modèle d'ordre complet. La combinaison des deux réduit drastiquement le coût du processeur, augmentant

inévitavelmente les performances du MOR. Ce travail est validé sur deux problèmes de benchmark. En outre, une stratégie d'échantillonnage adaptatif est également présentée dans ce manuscrit, qui est réalisée en tirant parti de l'approximation des modèles multi-fidélité. Vers la fin de la thèse, nous abordons un autre problème qui est généralement observé dans les cas où des maillages d'éléments finis adaptatifs sont déployés. Dans de tels cas, les méthodes MOR ne parviennent pas à produire une représentation de faible dimension car les snapshots ne sont pas des vecteurs de même longueur. Par conséquent, nous suggérons une méthodologie qui peut générer des fonctions de base réduites pour des snapshots adaptative.

**Title :** Advanced reduced-order modeling and parametric sampling for non-Newtonian fluid flows

**Keywords :** hyper-reduction, offline stabilization, multi-fidelity models, adaptive parametric sampling, adaptive snapshots

**Abstract :** The subject of this thesis concerns model-order reduction (MOR) of parameterized non-Newtonian flow problems that have significant industrial applications. Traditional MOR methods constrain the computational performance of such highly nonlinear problems, so we suggest a state-of-the-art hyper-reduction technique based on a sparse approximation to tackle the evaluation of nonlinear terms at much reduced complexity. We also provide offline stabilization strategy for stabilizing the constitutive model in the reduced order model framework that is less expensive to compute while maintaining the full order model's (FOM) accuracy. Combining the two significantly lowers the CPU cost as compared to the FOM

evaluation which inevitably boosts MOR performance. This work is validated on two benchmark flow problems. Additionally, an adaptive sampling strategy is also presented in this manuscript which is achieved by leveraging multi-fidelity model approximation. Towards the end of the thesis, we address another issue that is typically observed for cases when adaptive finite element meshes are deployed. In such cases, MOR methods fail to produce a low-dimensional representation since the snapshots are not vectors of same length. We therefore, suggest an alternate method that can generate reduced basis functions for database of space-adapted snapshots.