Efficient Reduction Techniques for the Simulation and Optimization of Parametrized Systems: Analysis and Applications

THÈSE N^{O} 6810 (2015)

PRÉSENTÉE LE 30 OCTOBRE 2015

À LA FACULTÉ DES SCIENCES DE BASE
CHAIRE DE MODÉLISATION ET CALCUL SCIENTIFIQUE
PROGRAMME DOCTORAL EN MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Federico NEGRI

acceptée sur proposition du jury:

Prof. F. Nobile, président du jury
Prof. A. Quarteroni, Prof. G. Rozza, directeurs de thèse
Prof. Y. Maday, rapporteur
Prof. S. Volkwein, rapporteur
Prof. J. Hesthaven, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2015

to my wife, Giorgia

Acknowledgements

I am very grateful to my advisor, Prof. Alfio Quarteroni, for providing me with time, precious advices, and valuable guidance whenever I needed it, as well as the freedom to explore different research paths. I am thankful to my co-advisor, Prof. Gianluigi Rozza, for the support in the early stages of this research project and for remaining a reference point after he moved to SISSA.

I would like to thank the members of the jury, Prof. Jan Hesthaven, Prof. Yvon Maday and Prof. Stefan Volkwein, for their very insightful and constructive feedback. Thanks also to Prof. Fabio Nobile who served as president of the jury.

A special thank goes to Dr. Andrea Manzoni, for being a precious colleague, coauthor and friend. Many thanks also to Dr. Luca Dede' for the stimulating interactions on a variety of subjects; in particular, his insights and suggestions have been crucial in the development of reduced-order models for mass transfer problems in hemodynamics. I also wish to thank Dr. David Amsallem for the fruitful collaboration and the inspiring discussions on model order reduction.

A very special thank goes to my office-mate Davide Forti. This work greatly benefited from his scientific enthusiasm and criticism. Most importantly, however, I wish to thank him for his friendship, support and steadfast encouragement.

Many thanks to all my present and past colleagues at CMCS, particularly MER Simone Deparis, Dr. Simone Rossi, Dr. Claudia Colciago and Andrea Bartezzaghi.

I cannot be grateful enough to my parents and sister for their generous care and continuous support, both moral and material.

Finally, I wish to greatly thank my wife Giorgia, who first encouraged me to start the PhD, patiently coped with the distance, and was finally brave enough to marry me right before the redaction of this dissertation; you certainly deserve more than a dedication.

I gratefully acknowledge the financial support of the Swiss National Science Foundation, through the project 141034 "Model reduction strategies for control, optimization and uncertainty quantification of parametrized systems".

Lausanne, October 2015

Abstract

This thesis is concerned with the development, analysis and implementation of efficient reduced order models (ROMs) for the simulation and optimization of parametrized partial differential equations (PDEs). Indeed, since the high-fidelity approximation of many complex models easily leads to solve large-scale problems, the need to perform multiple simulations to explore different scenarios, as well as to achieve rapid responses, often requires unaffordable computational resources. Alleviating this extreme computational effort represents the main motivation for developing ROMs, i.e. low-dimensional approximations of the underlying high-fidelity problem.

Among a wide range of model order reduction approaches, here we focus on the so-called projection-based methods, in particular Galerkin and Petrov-Galerkin reduced basis methods. In this context, the goal is to generate low cost and fast, but still sufficiently accurate ROMs which characterize the system response for the whole range of input parameters we are interested in. In particular, several challenges have to be faced to ensure reliability and computational efficiency. As regards the former, this thesis presents some heuristic approaches to approximate the stability factor of parameterized nonlinear PDEs, a key ingredient of any a posteriori error estimate. Concerning computational efficiency, we propose different strategies to combine the 'Matrix Discrete Empirical Interpolation Method' (MDEIM) with a state approximation resulting either from a proper orthogonal decomposition or a greedy approach. Specifically, we exploit the MDEIM to develop fast and efficient ROMs for nonaffinely parametrized elliptic and parabolic PDEs, as well as for the time-dependent Navier-Stokes equations. The efficacy of the proposed methods is demonstrated on a variety of computationally-intensive applications, such as the shape optimization of an acoustic device, the simulation of blood flow in cerebral aneurysms and the simulation of solute dynamics in blood flow and arterial walls.

Furthermore, the above-mentioned techniques have been exploited to develop a model order reduction framework for parametrized optimization problems constrained by either linear or nonlinear stationary PDEs. In particular, among this wide class of problems, here we focus on those featuring high-dimensional control variables. To cope with this high dimensionality and complexity, we propose an all-at-once optimize-then-reduce paradigm, where a simultaneous state and control reduction is performed. This methodology is applied first to a data reconstruction problem arising in haemodynamics, and then to several optimal flow control problems.

Keywords: reduced order models, reduced basis methods, error estimates, fluid dynamics, coupled problems, hemodynamics, PDE-constrained optimization, flow control.

Résumé

Le but de cette thèse est le dévelopement, l'analyse et l'implémentation de modèles d'ordre réduit (ROMs en abrégé pour l'acronyme anglais *reduced order models*) pour la simulation numérique et l'optimisation d'équations aux dérivées partielles (EDP) contenantes des paramètres. En fait, l'approximation numérique haute-fidélité de beaucoup de systèmes complexes devient rapidement très coûteuse dès qu'il s'agit d'effectuer une analyse paramétrique. Le recours à un modèle d'ordre réduit fidèle au système initial est alors indispensable pour réduire les coûts de calcul.

Parmi de nombreuses techniques de réduction d'ordre, dans cette thèse nous nous concentrons sur les méthodes de projection, en particulier les méthodes des bases réduites. Dans ce contexte, l'objectif est de préserver les charactéristiques physiques essentielles des phénomènes complexes dans le système réduit afin que les réponses produites restent de bonne qualité malgré le très faible nombre de degrés de liberté. En particulier, on doit se confronter avec de nombreux défis pour garantir la fiabilité et l'efficacité de ces méthodes. En ce qui concerne le premier aspect, nous proposons dans cette thèse des stratégies heuristiques pour l'approximation des constantes de stabilité d'EDPs paramétrisées non-linéaires ; celles-ci jouent un rôle particulièrement important dans l'estimation à posteriori des erreurs. En ce qui concerne l'efficacité de calcul, on propose plusieurs stratégies pour combiner la version matricielle de la méthode d'interpolation empirique discrète (MDEIM en abrégé pour l'acronyme anglais *matrix discrete empirical interpolation method*) avec une réduction de modèle obtenue par décomposition orthogonale propre ou par une stratégie du type *greedy*. En particulier, nous profitons de la stratégie MDEIM pour développer des ROMs très performantes pour des EDPs du type elliptiques et paraboliques paramétrées de manière non-affine, ainsi que pour les équations de Navier-Stokes. Toutes ces méthodes sont alors appliquées à différents problèmes. On considère d'abord un problème d'optimisation de forme dans le domaine de l'acoustique, ensuite la simulation de l'écoulement sanguin dans un anévrisme cérébrale, et finalement la simulation du transport d'oxygène dans une artère.

En outre, ces techniques sont utilisées pour développer une stratégie de réduction d'ordre pour des problèmes d'optimisation paramétrisés avec des EDP stationnaires (linéaires ou non-linéaires) comme contraintes. En particulier, parmi cette grande classe de problèmes, on se concentre sur ceux ayant des variables de contrôle avec un grand nombre de degrés de liberté. Afin de réduire la taille du problème et par conséquence complexitè, nous proposons une approche monolithique où la réduction est effectuée directement sur le système d'optimalité. Au niveau de l'application, on considère d'abord un problème de reconstruction dans le cadre de l'hémodynamique. Ensuite, des problèmes de contrôle d'écoulement sont présentés.

Mots-clés : modèles d'ordre réduit, méthodes des bases réduites, estimation d'erreur, mécanique des fluides numérique, problèmes couplés, hémodynamique, contrôle optimal d'équations aux dérivées partielles, contrôle d'écoulement.

Contents

List of Figures

List of Tables

List of Acronyms

PDE	Partial Differential Equation
FOM	Full Order Model
FE	Finite Elements
BDF	Backward Differentiation Formulas
SUPG	Streamline Upwind Petrov Galerkin
AS	Additive Schwarz
WSS	Wall Shear Stress
LHS	Latin Hypercube Sampling
SCM	Successive Constraint Method
RBF	Radial Basis Functions
ROM	Reduced Order Model
HROM	Hyper Reduced Order Model
MOR	Model Order Reduction
RB	Reduced Basis
POD	Proper Orthogonal Decomposition
EIM	Empirical Interpolation Method
DEIM	Discrete Empirical Interpolation Method
MDEIM	Matrix Discrete Empirical Interpolation Method
ROMES	Reduced Order Model Error Surrogates
GP	Gaussian Process

1 Introduction

1.1 Motivations and objectives

This thesis is concerned with the development, analysis and implementation of efficient reduced order models (ROMs) for the simulation and optimization of complex parametrized systems modeled by partial differential equations (PDEs). The combination of accurate mathematical models, fast numerical algorithms and powerful computing hardware makes nowadays possible the simulation of complex phenomena by means of high-fidelity (or full-order) approximation techniques such as, e.g., the finite element method. As a result, in several fields, ranging from aerospace and mechanical engineering to medicine and finance, numerical simulations of PDEs represent a reliable tool for the prediction of input/output response, design and optimization of complex systems.

However, for many time-critical applications, high-fidelity numerical simulations are so computationally demanding that they are either too slow to satisfy the problem's time constraints or they cannot be used as often as needed. This is, e.g., the case of PDEs depending on a set of input parameters characterizing the physical and/or geometrical configuration of the underlying system. Indeed, despite the massive computer resources currently available, problems involving the repeated solution of PDEs on different data settings or requiring a numerical solution very rapidly still represent a challenge for high-fidelity numerical techniques.

Alleviating this extreme computational effort represents the main motivation for developing reduced-order models, i.e. low-dimensional approximation of the underlying high-fidelity problem. Among a wide range of model order reduction (MOR) approaches (see, e.g., [BME03, Ant05, SvdVR08, QR14]), here we focus on the so-called projection-based methods, in particular Galerkin and Petrov-Galerkin reduced basis (RB) methods. To decrease the dimension of the model, these methods first execute an offline training phase, where expensive computations are carried out to generate a low-dimensional space (the RB space) which captures the essential features of the high-fidelity parameter-to-solution map. Then, during the online phase, for a given system configuration, RB methods seek an approximate solution belonging to this low-dimensional space. By this approach, a large-scale system (either dynamical or time-independent) is replaced by a

much smaller one, potentially leading to a dramatic reduction of the computational costs and solution times.

In this context, the goal is to generate low cost and fast, yet still sufficiently accurate, reduced-order models which characterize the system response for the whole range of input parameters we are interested in. To this end, several challenges have to be faced:

1. basis computation: how to efficiently construct a basis of the RB space which properly captures the parametric dependence;
2. accuracy and reliability: how to quantify the error between the reduced and the high-fidelity approximation;
3. computational efficiency: how to ensure that, for any given value of the input parameters, forming and solving the reduced model is fast and inexpensive.

Here, we focus on the last two aspects, relying on either the proper orthogonal decomposition (POD) [Lum67, Sir87] or the greedy algorithm [PRV$^+$02, VPRP03] to compute global (over the parameter space) reduced bases.

As regards accuracy and reliability, the reduced model should guarantee a desired level of accuracy with respect to the underlying high-fidelity approximation, uniformly over the parameter space. To quantify the error between the reduced and the high-fidelity approximation, suitable residual-based posteriori error bounds and estimators have been developed in the case of elliptic [PRV$^+$02, VPRP03], parabolic [GP05], hyperbolic [HO08] and nonlinear [VPP03, VP05, YPU14, Yan14] PDEs. These error estimates are exploited for both the offline training of the reduced model and its online certification. They usually involve two main ingredients: the norm of the residual of the high-fidelity problem and its stability factor. While the former can be efficiently calculated thanks to a suitable offline-online computational splitting, the evaluation of the latter requires the solution of a large-scale eigenvalue problem. To overcome this computational bottleneck, the so-called Successive Constraint Method (SCM) has been first introduced in [HRSP07] and further developed in [CHMR08, CHMR09, HKC$^+$10, RHM13] to construct a parametric lower bound of stability factors. Here, we propose a linearized, heuristic version of this method providing a suitable estimate of the stability factor for quadratically nonlinear PDEs, such as the Navier-Stokes equations. Moreover, we develop an alternative heuristic strategy, which combines a radial basis interpolant, suitable criteria to ensure its positiveness, and an adaptive choice of interpolation points through a greedy procedure.

Concerning computational efficiency, given a value of the input parameters, the computational costs associated with assembling and solving the ROM should be independent of the dimension of the original high-fidelity model. Achieving this goal is particularly challenging when dealing with, e.g., nonlinear, nonaffinely parametrized, and multiphysics problems. Here, we propose different strategies to combine the recently proposed Matrix Discrete Empirical Interpolation Method (MDEIM) [CTB12, WSH14, BGW15, CTB15] with a state approximation resulting either from a POD or a greedy approach. Specifically, we exploit this technique to develop fast and efficient ROMs for nonaffinely parametrized elliptic and parabolic PDEs, as well as for the time-dependent Navier-Stokes equations. The efficacy of the proposed methods is demonstrated on a variety of computationally-intensive applications, including the shape optimization of an acoustic device, the simulation of blood flow in cerebral aneurysms and the simulation of solute dynamics in blood flow and arterial walls.

Reduced-order models are not only used for the numerical simulation of systems described by parametrized PDEs. Very often, the ultimate goal is not only the prediction of the input-output response of a system (i.e. the solution of a forward problem), but rather the optimization of some of its performances, or the optimal control of the underlying process in order to reach a desired state. In these cases, the goal is to minimize (or maximize) some quantities of interest related to the underlying state variable, by acting on suitable control or design variables. Remarkable instances include *(i)* optimal control problems, where we act on source/boundary terms or physical coefficients affecting the problem; *(ii)* optimal design problems, where the design variables are related with the geometrical configuration of the domain; and *(iii)* identification or data assimilation problems, where some unknown or uncertain features of the state system are estimated by exploiting the measurements of some outputs. We refer to this rather general class of problems as to PDE-constrained optimization problems [ABG+06, IK08, HPUU09, BS11].

Whenever the system configuration depends on a set of parameters, its response will be parameter dependent as well, and so will be the optimal control. For this reason, in order to possibly compare different scenarios or to evaluate the sensitivity and robustness of the optimal solution with respect to the parameters, we are required to solve the optimization problem many times. This entails large computational costs and may be very time-consuming already in the non-parametric case. Therefore, when performing the optimization process for many different parameter values or else when, for a new given configuration, the solution has to be computed in a rapid way, reducing the computational complexity is mandatory. In this thesis, we propose a general reduction framework for the efficient numerical solution of high dimension/complexity parametrized PDE-constrained optimization problems. In particular, among this wide class of problems, here we focus on those featuring high-dimensional control variables. To cope with this high dimensionality and complexity, we propose an all-at-once optimize-then-reduce paradigm, where a simultaneous state and control reduction is performed. In this context, we take advantage of both the above-mentioned techniques.

In the remainder of this chapter, we introduce in more detail the class of problems we deal with, our applications of interest, the reduction approaches we consider and the main original contributions of this work.

1.2 Forward problems

In this section we introduce a general algebraic formulation of stationary and time-dependent parametrized PDEs. Throughout this thesis, we denote by $\boldsymbol{\mu} = (\mu_1, \dots, \mu_P) \in \mathcal{D} \subset \mathbb{R}^P$ a vector of input parameters related to, e.g., boundary conditions, material properties, geometry configuration; the parameter domain $\mathcal{D}$ is assumed to be a compact subset of the Euclidean space $\mathbb{R}^P$, $P \geq 1$.

1.2.1 The steady case

Consider first a large-scale system of parametrized, nonlinear equations arising from the discretization of a parametrized PDE

$$\mathbf{E}(\mathbf{y}_h; \boldsymbol{\mu}) = \mathbf{0}, \tag{1.1}$$

where $\mathbf{y}_h = \mathbf{y}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is the state variable and $\mathbf{E} : \mathbb{R}^{N_h} \times \mathcal{D} \to \mathbb{R}^{N_h}$ is the residual vector encoding the differential operator. Equation (1.1) may result, for instance, from the finite element discretization of an advection-diffusion equation, the Helmholtz equation or the stationary Navier-Stokes equations. We will refer to (1.1) as high-fidelity or full-order model (FOM). If the governing equations are linear in the state, then the system is written as

$$\mathbf{A}(\boldsymbol{\mu})\mathbf{y}_h = \mathbf{g}(\boldsymbol{\mu}), \tag{1.2}$$

where $\mathbf{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ is a matrix that (possibly) depends on the parameters but not on the state, and the forcing vector $\mathbf{g}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ encodes the contributions of boundary conditions and source terms.

The first step in generating a projection-based ROM is to approximate the N_h-dimensional unknown $\mathbf{y}_h(\boldsymbol{\mu})$ by a linear combination of $N \ll N_h$ basis vectors,

$$\mathbf{y}_h(\boldsymbol{\mu}) \approx \mathbf{V}\mathbf{y}_N(\boldsymbol{\mu}).$$

The transformation (or projection) matrix $\mathbf{V} = [\boldsymbol{\zeta}_1, \ldots, \boldsymbol{\zeta}_N] \in \mathbb{R}^{N_h \times N}$ contains as columns the reduced basis vectors $\boldsymbol{\zeta}_i$, and the vector $\mathbf{y}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ contains the corresponding (unknown) coefficients. For the computation of the basis vectors, here we consider both proper orthogonal decomposition and the greedy algorithm. Both methods start from a set of solutions – commonly referred to as *snapshots* – computed by solving (1.1) for selected values of the parameter $\boldsymbol{\mu}$. In the POD approach [CBS99, BTDW03, BTWG08, KV07, TUV10, KV12, ADVN09, Pin08, Vol11], $n_s \geq N$ solutions corresponding to a suitable sampling of the parameter domain are collected in the columns of a matrix $\mathbf{S} \in \mathbb{R}^{N_h \times n_s}$. The POD basis vectors are given by the left singular vectors of the matrix $\mathbf{S}$ that correspond to the N largest singular values. Following this approach, a critical issue in constructing the reduced basis is the parameter space sampling. To address this challenge, the greedy algorithm was introduced in [PRV$^+$02, VPRP03] to adaptively choose samples by finding the location at which a suitable estimate of the error $\|\mathbf{y}_h(\boldsymbol{\mu}) - \mathbf{V}\mathbf{y}_N(\boldsymbol{\mu})\|$ is maximum.

Once we have built the reduced basis matrix $\mathbf{V}$, the reduced problem is generated via a projection approach. More precisely, it is obtained by enforcing the orthogonality of the residual $\mathbf{E}(\mathbf{V}\mathbf{y}_N; \boldsymbol{\mu})$ to the N-dimensional space spanned by the columns of a basis $\mathbf{W} \in \mathbb{R}^{N_h \times N}$. The latter is referred to as test (or left) basis, and the span of its columns generates the so called test subspace. This yields the following Petrov-Galerkin reduced basis problem: given $\boldsymbol{\mu} \in \mathcal{D}$, find $\mathbf{y}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ such that

$$\mathbf{W}^T\mathbf{E}(\mathbf{V}\mathbf{y}_N; \boldsymbol{\mu}) = \mathbf{0}. \tag{1.3}$$

The Galerkin reduced basis problem corresponds to choosing $\mathbf{W} = \mathbf{V}$. For the linear system (1.2), the ROM (1.3) reduces to the following linear system of dimension N

$$\mathbf{W}^T\mathbf{A}(\boldsymbol{\mu})\mathbf{V}\mathbf{y}_N = \mathbf{W}^T\mathbf{g}(\boldsymbol{\mu}). \tag{1.4}$$

In the nonlinear case, (1.3) consists instead of a set of N nonlinear equations which can be conveniently solved by means of, e.g., Newton method. Given $\boldsymbol{\mu} \in \mathcal{D}$ and an initial guess $\mathbf{y}_N^0$, for $k = 0, 1, \ldots$ until convergence, we seek $\delta\mathbf{y}_N \in \mathbb{R}^N$ such that

$$\mathbf{W}^T\mathbf{A}(\mathbf{V}\mathbf{y}_N^k; \boldsymbol{\mu})\mathbf{V}\delta\mathbf{y}_N = -\mathbf{W}^T\mathbf{E}(\mathbf{V}\mathbf{y}_N^k; \boldsymbol{\mu}), \tag{1.5}$$

and then set $\mathbf{y}_N^{k+1} = \mathbf{y}_N^k + \delta\mathbf{y}_N$. Here, the matrix $\mathbf{A}(\cdot;\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ denotes the Jacobian of $\mathbf{E}(\cdot;\boldsymbol{\mu})$ with respect to its first argument.

Broadly speaking, solving the ROM (1.3) entails the solution of one or a sequence of linear systems of the form

$$\mathbf{A}_N(\mathbf{y}_N^k;\boldsymbol{\mu})\delta\mathbf{y}_N = -\mathbf{E}_N(\mathbf{y}_N^k;\boldsymbol{\mu}), \tag{1.6}$$

where the reduced Jacobian matrix $\mathbf{A}_N \in \mathbb{R}^{N\times N}$ and the reduced residual vector $\mathbf{E}_N \in \mathbb{R}^N$ are given by

$$\mathbf{A}_N(\mathbf{y}_N^k;\boldsymbol{\mu}) = \mathbf{W}^T\mathbf{A}(\mathbf{V}\mathbf{y}_N^k;\boldsymbol{\mu})\mathbf{V}, \qquad \mathbf{E}_N(\mathbf{y}_N^k;\boldsymbol{\mu}) = \mathbf{W}^T\mathbf{E}(\mathbf{V}\mathbf{y}_N^k;\boldsymbol{\mu}). \tag{1.7}$$

System (1.6) has low dimension, since typically $N \ll N_h$, and is (in principle) much faster and less computationally intensive to solve than the original high-fidelity one. However, forming the reduced matrix $\mathbf{A}_N$ and the vector $\mathbf{E}_N$ still involves computations whose complexity depends on N_h. Indeed, given a parameter $\boldsymbol{\mu} \in \mathcal{D}$ and a state vector $\mathbf{y}_N \in \mathbb{R}^N$, forming $\mathbf{A}_N(\mathbf{y}_N;\boldsymbol{\mu})$ and $\mathbf{E}_N(\mathbf{y}_N;\boldsymbol{\mu})$ requires to: compute the full-order representation $\mathbf{V}\mathbf{y}_N \in \mathbb{R}^{N_h}$ of $\mathbf{y}_N$, assemble the high-fidelity matrix $\mathbf{A}(\mathbf{V}\mathbf{y}_N;\boldsymbol{\mu})$ and vector $\mathbf{E}_N(\mathbf{V}\mathbf{y}_N;\boldsymbol{\mu})$, and project them onto the reduced subspace.

For the linear problem (1.2), a convenient situation arises when the high-fidelity matrix (resp. vector) can be expressed as a linear combination of constant matrices (resp. vectors) weighted by suitable parameter dependent coefficients. Indeed, each $\boldsymbol{\mu}$-independent term of this weighted sum can be projected onto the reduced basis space and stored in the offline phase, thus enabling a very rapid (and N_h-independent) assembling of the ROM during the online phase. On the other hand, if $\mathbf{A}(\boldsymbol{\mu})$ and $\mathbf{g}(\boldsymbol{\mu})$ are not affine – that is, if they do not feature an affine decomposition – this computational strategy cannot be pursued. Unfortunately, many applications of interest feature a nonaffine dependency with respect to parameters. This is the case for instance when dealing with parametrized shape deformations. Furthermore, even if the problem admits an affine decomposition, exploiting this decomposition might require intrusive changes to the high-fidelity model implementation (see, e.g., [RHP08]), or even be impossible when using black-box high-fidelity solvers.

In order to recover an affine structure in those cases where the operator $\mathbf{A}(\boldsymbol{\mu})$ is nonaffine (or such a decomposition is not readily available), we must introduce a further level of reduction, called hyper-reduction or system approximation [CFCA13], employing suitable techniques such as EIM [BMNP04, DHO12], DEIM [CS10], best point interpolation method [NPP08], missing point estimation [AWWB08], and gappy POD [ES95, BTDW04, Wil06, CBMF11]. Here, we rely on the recently proposed MDEIM, which enjoys some remarkable properties: it operates at the purely algebraic level, so that it can be implemented in a non-intrusive way without requiring changes to the high-fidelity model implementation. Moreover, it can be applied either simultaneously or prior to the reduced space construction.

In Chap. 3, we show how MDEIM can be exploited to deal with complex physical and geometrical parametrizations in a non-intrusive, efficient and purely algebraic way. In particular, we propose different strategies to combine MDEIM with a state approximation resulting either from a greedy approach or proper orthogonal decomposition. A posteriori error estimates accounting for the MDEIM error are also developed. This technique will

be also exploited when dealing with nonaffine and nonlinear dynamical systems, as well as with nonaffinely parametrized PDE-constrained optimization problems in Chaps. 5 and 6. Here, instead, we do not apply it to the case of nonlinear steady problems, as our nonlinear problem of interest is represented by the Navier-Stokes equations, which feature a quadratic nonlinearity admitting an exact affine expansion.

Thanks to MDEIM, we thus achieve the goal of computational efficiency, by enabling the resulting ROMs to provide significant speedups with respect to the original high-fidelity models. However, despite the computational savings, these ROMs would be of limited practical value if they would not provide a sufficient accuracy. To quantify the error between the reduced and the high-fidelity solution, residual-based error estimates are usually employed. Indeed, for $\mathbf{V}\mathbf{y}_N(\boldsymbol{\mu})$ sufficiently close to $\mathbf{y}_h(\boldsymbol{\mu})$, the implicit function theorem (see, e.g., [CR97, Zei85]) yields the following bound

$$\|\mathbf{y}_h(\boldsymbol{\mu}) - \mathbf{V}\mathbf{y}_N(\boldsymbol{\mu})\|_{\mathbf{X}} \leq \frac{2}{\beta_h^N(\boldsymbol{\mu})} \|\mathbf{E}(\mathbf{V}\mathbf{y}_N(\boldsymbol{\mu}); \boldsymbol{\mu})\|_{\mathbf{X}^{-1}}. \tag{1.8}$$

Here, $\|\mathbf{v}\|_{\mathbf{X}} = \sqrt{\mathbf{v}^T \mathbf{X} \mathbf{v}}$ denotes a discrete norm induced by a symmetric positive definite matrix $\mathbf{X} \in \mathbb{R}^{N_h \times N_h}$, while the stability factor

$$\beta_h^N(\boldsymbol{\mu}) = \sigma_{\min}\left(\mathbf{X}^{-1/2} \mathbf{A}(\mathbf{V}\mathbf{y}_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \mathbf{X}^{-1/2}\right) \tag{1.9}$$

is the minimum generalized singular value of the matrix $\mathbf{A}(\mathbf{V}\mathbf{y}_N(\boldsymbol{\mu}); \boldsymbol{\mu})$. Whenever an (either exact or approximate) affine decomposition of the residual is available, the evaluation of its norm can be performed rapidly thanks to a suitable offline-online computational splitting; see, e.g., [PRV+02, RHP08] for further details. The key remaining problem is the computation of the stability factor $\beta_h^N(\boldsymbol{\mu})$, which must be done in an efficient way without involving the high-fidelity model.

As already anticipated, in this thesis (see Chap. 2) we propose two different approaches for constructing an approximation to $\beta_h^N(\boldsymbol{\mu})$ which can be rapidly evaluated in the online phase. First, we present a linearized, heuristic version of the natural norm SCM [SVH+06, HKC+10] algorithm providing a suitable estimate of the stability factor for quadratically nonlinear PDEs, such as the Navier-Stokes equations. Then, we develop a far more general heuristic strategy, which combines a radial basis interpolant and an adaptive choice of interpolation points through a greedy procedure. As it completely bypasses the high-fidelity problem formulation by directly seeking an approximation of the stability factor, this strategy is suitable for both linear and nonlinear, affine and nonaffine problems. We assess its accuracy, robustness and computational performances in the case of the Navier-Stokes equations. Then, we also apply this technique to the Helmholtz equation in the context of acoustics, as well as to several optimization problems in Chap. 6.

1.2.2 The unsteady case

We now consider a parameterized dynamical system arising from the spatial discretization of a time-dependent PDE:

$$\mathbf{M}(t; \boldsymbol{\mu}) \frac{d\mathbf{y}_h}{dt} + \mathbf{E}(\mathbf{y}_h; t; \boldsymbol{\mu}) = \mathbf{0}, \tag{1.10}$$

with $\mathbf{y}_h(0) = \mathbf{y}_0$. Here, $\mathbf{M} \in \mathbb{R}^{N_h \times N_h}$ denotes the mass matrix (possibly dependent on time and parameters), while $\mathbf{E} : \mathbb{R}^{N_h} \times \mathbb{R}^+ \times \mathcal{D} \to \mathbb{R}^{N_h}$ is a vector encoding the differential operator. In this thesis, equation (1.10) is derived from the finite element discretization of *(i)* the Navier-Stokes equations, and *(ii)* the advection-diffusion equations modeling, e.g., the mass transport of a substance in a fluid. Indeed, as already mentioned, we are interested in simulating mass transport phenomena in the cardiovascular system.

In this context, mass transfer refers to exchange of substances between blood and the arterial wall. Substances of interest include oxygen [ME97, PLP$^+$02, PP03, PZPQ05], low-density lipoprotein (LDL) [SE02, OKP08], as well as potential therapeutic agents designed for local delivery by intravascular infusion [CBBH08, HHD14, HHB$^+$12]. In all these cases, the solutes are passive scalars essentially convected by the blood along the vessels, while the absorption processes through the arterial wall are related to the stress induced by the blood on the vascular tissue. Therefore, simulation of solute dynamics first requires computation of the blood flow, followed by solution of the equations governing mass transport. Since the former is modeled by the Navier-Stokes equations, we are first led to solve a nonlinear system of the form (1.10). On the other hand, since the mass transport equations are linear, their finite element discretization results in a linear problem of the form

$$\mathbf{M}(t;\boldsymbol{\mu})\frac{d\mathbf{y}_h}{dt} + \mathbf{A}(t;\boldsymbol{\mu})\mathbf{y}_h = \mathbf{g}(t;\boldsymbol{\mu}), \tag{1.11}$$

in analogy with (1.2). The mass transport model is coupled in one direction: blood flow dictates the transport of the solute (thus matrix $\mathbf{A}(t;\boldsymbol{\mu})$ in (1.11) depends on the blood velocity), but solute dynamics has no effect on the blood flow solution.

The solute distribution and availability inside the vessels and into the vascular walls is thus strongly related to flow dynamics of blood. In particular, it has been observed that irregular flow patterns (such as flow separation, flow recirculation, low and oscillating wall shear stresses) result in disturbed mass distributions. In the case of LDL and oxygen transport, this may eventually lead to the development of atherosclerotic diseases. The numerical simulation of solutes dynamics could therefore be useful for revealing the relationships between irregular flow patterns, mass transfer, and possible pathogenesis. Moreover, numerical simulations could also serve to design more effective personalized treatments in the case of drug delivery, see e.g. [HHD14, HHB$^+$12]. However, in all these cases the predicted solute distribution is significantly affected by a number of unknown, uncertain or patient-specific parameters which enter the underlying mathematical models. In turn, these parameters may need to be either estimated (for instance to impose realistic boundary conditions), calibrated (to fit the model to physical observations) or optimized (for example to provide an effective personalized treatment in the case of drug delivery). To this end, reduced-order models could bring great advantages by enabling a more extensive parameter exploration and fast simulations.

Thanks to the one-way coupling from blood flow to the mass transport equations, the reduction process can be split into two steps: we first build a ROM for the blood flow, and then we generate a ROM to simulate the solute dynamics. In both cases, projection-based model reduction approaches reduce the dimension of the high-fidelity model by searching for solutions $\mathbf{y}_h(t;\boldsymbol{\mu}) \approx \mathbf{V}\mathbf{y}_N(t;\boldsymbol{\mu})$ belonging to a low-dimensional subspace a priori computed by expensive offline computations. Then, by a Petrov-Galerkin projection they

generate the reduced-order problem

$$\mathbf{M}_N(t;\boldsymbol{\mu})\frac{d\mathbf{y}_N}{dt} + \mathbf{E}_N(\mathbf{y}_N;t;\boldsymbol{\mu}) = \mathbf{0}, \tag{1.12}$$

where the reduced matrix and vector are given by

$$\mathbf{M}_N(t;\boldsymbol{\mu}) = \mathbf{W}^T\mathbf{M}(t;\boldsymbol{\mu})\mathbf{V}, \qquad \mathbf{E}_N(\mathbf{V}\mathbf{y}_N;t;\boldsymbol{\mu}) = \mathbf{W}^T\mathbf{E}(\mathbf{y}_N;t;\boldsymbol{\mu}).$$

The basis vectors can be computed with several techniques such as balanced truncation [Moo81, SA02, GA04], Krylov-based methods [Ant05], proper orthogonal decomposition, the greedy and POD-greedy algorithms [GP05, GMNP07, HO08, Gre12], and the space-time greedy algorithm [YPU14, Yan14]. Moreover, both global and local bases can be constructed, see e.g. [ACCF09, AF11]. Here, we always consider global bases (over time and parameters) built by means of POD. As in the case of static systems, a crucial issue is ensuring the fast assembling of the reduced vectors and matrices. We achieve this goal by employing the MDEIM.

Considering the linear problem (1.11), we propose a *simultaneous system approximation and state-space reduction* approach where a global reduced basis is constructed by means of POD and the system matrix is approximated by MDEIM. This strategy demonstrates to be particularly effective in the case of the coupled mass transport model, where the matrix $\mathbf{A}(t;\boldsymbol{\mu})$ depends in a highly nonaffine way (because of the underlying SUPG stabilized finite element approximation) on the blood velocity.

As regards the time-dependent Navier-Stokes equations, several approaches have been proposed for the construction of suitable ROMs, see, e.g., [KV03, BGL06, BBI09, WABI12, WLBI10, IW14, BCI13, XFB$^+$14, WH15] and references therein. Here, we develop a new reduction strategy which is tailored to the underlying high-fidelity approximation. For the latter, we employ equal-order SUPG stabilized finite elements for the space discretization, a BDF time discretization and a semi-implicit treatment of the convective term [FD15, GSV06]. The reduced-order model is then generated by a Galerkin projection of the resulting fully-discrete problem onto a subspace generated by POD. A hybrid approach for the treatment of the nonlinear operators is employed: we apply an exact quadratic expansion to reconstruct the convective term, while MDEIM is used to approximate the nonlinear (with respect to the convective velocity) SUPG terms.

The proposed reduction strategy is tested in Chap. 3 on a benchmark problem modeling the fluid dynamics and heat transfer around a circular obstacle. Then, in Chap. 4 it is applied to simulate blood flow in a cerebral aneurysm and solute dynamics in a femoropopliteal bypass.

1.3 PDE-constrained optimization problems

We can state the general form of a PDE-constrained optimization problem as

$$\min_{\mathbf{y}_h,\mathbf{u}_h} \mathcal{J}_h(\mathbf{y}_h,\mathbf{u}_h) \quad \text{subject to} \quad \mathbf{E}(\mathbf{y}_h,\mathbf{u}_h) = \mathbf{0}, \tag{1.13}$$

where $\mathbf{y}_h \in \mathbb{R}^{N_{h,y}}$ is the state variable, $\mathbf{u}_h \in \mathbb{R}^{N_{h,u}}$ the control (design) variable, $\mathcal{J}_h$ the objective function and $\mathbf{E} \in \mathbb{R}^{N_{h,y}}$ the residual of the state equation. Problem (1.13) can represent an optimal design, optimal control, or inverse problem, depending on the nature

of the objective function and control variable. Even though here we limit ourselves to the stationary case, a time-dependent state equation (like (1.10)) could be considered as well.

In the parametrized context, we can distinguish among two main classes of problems depending on the role played by the parameter vector $\boldsymbol{\mu} \in \mathcal{D}$:

1. *parametric optimization problems*: in this case, the control variable is a vector $\mathbf{u}_h = \boldsymbol{\mu}_c$ or a given function $\mathbf{u}_h = \mathbf{u}_h(\boldsymbol{\mu}_c)$ of control parameters $\boldsymbol{\mu}_c \in \mathcal{D}_c \subset \mathbb{R}^{P_c}$. We assume that both $\mathbf{E}$ and $\mathcal{J}$ might depend also on a set of additional scenario parameters $\boldsymbol{\mu}_s \in \mathcal{D}_s \subset \mathbb{R}^{P_s}$, that characterize the system being controlled. Consequently, $\boldsymbol{\mu} = (\boldsymbol{\mu}_s, \boldsymbol{\mu}_c)$ and $\mathcal{D} = \mathcal{D}_s \times \mathcal{D}_c$. In this case (1.13) can be more precisely formulated as: given $\boldsymbol{\mu}_s \in \mathcal{D}_s$,

$$\min_{\mathbf{y}_h, \boldsymbol{\mu}_c} \mathcal{J}_h(\mathbf{y}_h, \mathbf{u}_h(\boldsymbol{\mu}_c); \boldsymbol{\mu}_s) \quad \text{s.t.} \quad \mathbf{E}(\mathbf{y}_h, \mathbf{u}_h(\boldsymbol{\mu}_c); \boldsymbol{\mu}_s) = \mathbf{0}; \tag{1.14}$$

2. *parametrized optimization problems*: in this case, we only deal with a vector of scenario parameters $\boldsymbol{\mu}_s \in \mathcal{D}_s$, while the control variable is not a function of the parameters. Typically, the control variable is high-dimensional, as it results from the discretization of an infinite dimensional function. Here $\boldsymbol{\mu} = \boldsymbol{\mu}_s$ and $\mathcal{D} = \mathcal{D}_s$, so that in this case (1.13) reads: given $\boldsymbol{\mu}_s \in \mathcal{D}_s$,

$$\min_{\mathbf{y}_h, \mathbf{u}_h} \mathcal{J}_h(\mathbf{y}_h, \mathbf{u}_h; \boldsymbol{\mu}_s) \quad \text{s.t.} \quad \mathbf{E}(\mathbf{y}_h, \mathbf{u}_h; \boldsymbol{\mu}_s) = \mathbf{0}. \tag{1.15}$$

After the pioneering works by Ito and Ravindran in the late 90s [IR98a, IR98b, Rav00], RB methods have been extensively applied to PDE-constrained optimization problems in the past two decades. POD techniques have been originally used to deal with non-parametric time-dependent problems, see e.g. [AFS00, BC08, HV05, cSN15], and then for parametric optimization problems [BTDW04, TUV11, AZCF14, ZF15], where parameters are considered as control variables. POD has been successfully applied also in the feedback control context, see e.g. [ABK01, KVX04, GU14, AV14] and [BSV14] for a recent review. On the other hand, RB methods based on greedy algorithms have been formerly developed to deal with parametric optimization problems, see e.g. [QRQ07, LR10, AHH$^+$12, MQR12b, TUV11, DH13]), and then parametrized (linear-quadratic) optimization problems [Ded10, NRMQ13, KG14a, KG14b, NMR15]. Another rapidly growing field of interest indeed very close to PDE-constrained optimization is represented by the use of ROM for inverse identification problems, see e.g. [GNV$^+$07, GFWG10, CN12, LWG10, CMW15, MPPY15].

In the case of parametric optimization problems, a RB method operates a state reduction in order to solve the state system in a reduced state space for any new parameter vector. A simultaneous state and control reduction is instead required in the case of parametrized optimization problems, where the control variable undergoes the same procedure adopted for achieving a low-dimensional approximation of the state variable.

In this thesis, we focus on this second class of problems. In particular, we propose a model order reduction framework for parametrized quadratic optimization problems constrained by linear and nonlinear stationary PDEs. By characterizing the solutions of the optimization problem as the solutions of the corresponding optimality system, we build

a ROM following a suitable all-at-once optimize-then-reduce paradigm. Low-dimensional spaces for the state, control and adjoint variables are simultaneously constructed by means of either the greedy algorithm or proper orthogonal decomposition. By this approach, we can easily estimate the error between the high-fidelity and reduced solutions using a bound similar to (1.8); further, an estimate for the error on the cost functional is obtained. Then, for the sake of computational efficiency, we integrate into this framework the ROM Error Surrogates (ROMES) method presented in [DC15] to generate tighter error indicators. The latter models the *bound-to-error map* as a Gaussian Process [RW06] and generates (at low cost) a much sharper estimate of the error. We then embed this technique into POD and greedy strategies for the basis construction. Finally, we specify this general framework in the case of optimization problems constrained by linear elliptic, Stokes and Navier-Stokes equations. In Chap. 6, the methodology is applied first to a data reconstruction problem arising in haemodynamics, and then to several optimal flow control problems.

1.4 Thesis outline

The first part of this thesis (Chapters 2-4) is mainly focused on forward problems, while the second part (Chapters 5-6) is specifically devoted to optimization problems. In Chapter 7, we summarize some general conclusions and highlight some areas of future work. More details about the main body of the thesis are provided below.

Chapter 2 is devoted to the approximation of stability factors in nonlinear, inf-sup stable parametrized PDEs. We first propose a linearized version of the SCM and then an alternative heuristic strategy based on adaptive radial basis functions interpolation. We provide some theoretical results to support the proposed strategies, which are then applied to a set of test cases dealing with parametrized Navier-Stokes equations.

In Chapter 3 we apply MDEIM for the efficient reduction of nonaffine and nonlinear parameterized systems. Reduced-order models for nonaffinely parametrized elliptic and parabolic PDEs, as well as for the time-dependent Navier-Stokes equations, are proposed. Their efficacy is demonstrated on the solution of two computationally-intensive classes of problems occurring in engineering contexts, namely PDE-constrained shape optimization and parametrized coupled problems.

In Chapter 4 the methods developed in Chapter 3 are applied first to the simulation of blood flow in a cerebral aneurysm and then to the simulation of solute dynamics in the vessel wall of a femoropopliteal bypass.

A model order reduction framework for parametrized quadratic optimization problems constrained by linear and nonlinear stationary PDEs is presented in Chapter 5. Particular emphasis is put on to the construction of stable reduced spaces, computational efficiency and sharp error estimation. We specify this general framework in the case of optimization problems constrained by linear elliptic, Stokes and Navier-Stokes equations.

Chapter 6 shows the numerical performances of the above method dealing with an optimal heat transfer problem, a data reconstruction problem arising in haemodynamics, and several optimal flow control problems.

All the numerical results reported in this thesis have been obtained using a research code developed by the author in the MATLAB® [Mat] environment.

This thesis contains results which are already published in journal articles or have been submitted for publication in a similar form. Chapter 2 is based upon joint work with A. Manzoni which has already been published in [MN15]. Chapter 3 is partially based upon joint work with D. Amsallem and A. Manzoni which is available as submmitted pre-print [NMA15], while Chap. 5 is based upon the submitted pre-print [Neg15]. Finally, many of the numerical results presented in Chap. 6 were already reported in [RMN12, NMR15, QMN16, Neg15], partially based upon joint work with A. Manzoni, G. Rozza and A. Quarteroni. The whole presentation, however, is original.

2 Heuristic strategies for the approximation of stability factors

In this chapter we present some heuristic strategies to compute rapid and reliable approximations to stability factors in nonlinear, inf-sup stable parametrized PDEs. The efficient evaluation of these quantities is crucial for the rapid construction of a posteriori error estimates to reduced basis approximations. We first propose a linearized, heuristic version of the Successive Constraint Method (SCM), providing an approximation – rather than a lower bound as in the original SCM – of the stability factor. Moreover, for the sake of computational efficiency, we develop an alternative heuristic strategy, which combines a radial basis interpolant and an adaptive choice of interpolation points through a greedy procedure. We provide some theoretical results to support the proposed strategies, which are then applied to a set of test cases dealing with parametrized Navier-Stokes equations. Finally, we show that the interpolation strategy is inexpensive to apply and robust even in the proximity of bifurcation points, where the estimate of stability factors is particularly critical.

2.1 Introduction

Stability factors of differential operators are relevant for the well-posedness analysis of problems governed by PDEs and enter in the (a posteriori) error estimates of any numerical approximation method. Their rapid and reliable evaluation is thus crucial, especially when dealing with nonlinear parametrized PDEs.

In this chapter we focus on (quadratically) nonlinear parametrized PDEs, which can be written in the following general form:

$$\mathcal{E}(y;\boldsymbol{\mu}) = 0 \quad \text{in } V', \tag{2.1}$$

being V a suitable Hilbert space, V' its dual and $\mathcal{E} : V \times \mathcal{D} \to V'$ a nonlinear, inf-sup stable, parametrized operator. Equation (2.1) represents the continuous counterpart of the algebraic problem (1.1) introduced in Sect. 1.2.1. A meaningful example is represented by the steady Navier-Stokes equations parametrized with respect to the Reynolds number.

Our ultimate goal is to compute, in a very efficient way, a numerical approximation of the solution $y(\boldsymbol{\mu})$ for any $\boldsymbol{\mu} \in \mathcal{D}$. To this end, we rely on the reduced basis (RB) method, which allows to compute a reduced approximation $y_N(\boldsymbol{\mu}) \in V_N$ of the PDE solution $y(\boldsymbol{\mu}) \in V$, for any $\boldsymbol{\mu} \in \mathcal{D}$, as a linear combination of *snapshots* corresponding to a small set of sampled parameter values $\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N$. This can be made through a Galerkin projection in the low-dimensional subspace $V_N = \text{span}\{y_h(\boldsymbol{\mu}^1), \dots, y_h(\boldsymbol{\mu}^N)\}$, being $y_h(\boldsymbol{\mu}^i) \in V_h$, $i = 1, \dots, N$. Here V_h is a high-fidelity approximation space of dimension $N_h \gg N$ and $y_h(\boldsymbol{\mu}) \in V_h$ is the high-fidelity approximation to $u(\boldsymbol{\mu})$, obtained by any kind of numerical discretization technique. Moreover, we aim at providing an a posteriori error bound, which usually takes the form [CR97, VP05] (see also Sect. 1.2.1)

$$\|y_h(\boldsymbol{\mu}) - y_N(\boldsymbol{\mu})\|_V \leq \frac{2}{\beta_h(y_N(\boldsymbol{\mu}))} \|\mathcal{E}(y_N(\boldsymbol{\mu}), \boldsymbol{\mu})\|_{V_h'}, \tag{2.2}$$

at least for N sufficiently large, see Sect. 2.3.2. Here $\beta_h(y_N(\boldsymbol{\mu}))$ denotes the stability factor related with the (discrete, high-fidelity approximation of the) differential operator. The computation of $\beta_h(y_N(\boldsymbol{\mu}))$, for any $\boldsymbol{\mu} \in \mathcal{D}$, requires the solution of a generalized eigenproblem of dimension N_h, thus preventing both the offline and online efficiency of the RB approximation.

To overcome this computational bottleneck, the so-called Successive Constraint Method (SCM) has been first introduced in [HRSP07] (see also [CHMR08, CHMR09]). A general version using the so-called *natural norm* [SVH$^+$06] has been analyzed in [HKC$^+$10], while a recent application to Stokes equations is given in [RHM13]. This method has been developed for linear parametrized operators and provides a parametric lower bound to their stability factor. Since in the linear case the latter is independent of $y_N(\boldsymbol{\mu})$, the procedure admits an offline-online computational treatment for which the online cost is independent of N_h, and the offline computations are performed prior to the RB space construction.

In the nonlinear case, since the stability factor depends on the RB solution $y_N(\boldsymbol{\mu})$, the construction of suitable lower bounds can not be performed prior to (and independently of) the construction of the reduced space. To overcome this bottleneck, we propose to approximate $\beta_h(y_N(\boldsymbol{\mu}))$ by $\beta_h(y_h(\boldsymbol{\mu}))$, i.e. by the stability factor evaluated with respect to the high-fidelity solution $y_h(\boldsymbol{\mu})$. Indeed, thanks to the approximation property of the RB space V_N, we can prove that the error $|\beta_h(y_N(\boldsymbol{\mu})) - \beta_h(y_h(\boldsymbol{\mu}))|$ vanishes as $N \to N_h$. Then, we propose two different strategies to construct, prior to the generation of the RB space V_N, an estimate to the stability factor $\beta_h(y_h(\boldsymbol{\mu}))$.

In particular, we first develop a *linearized* version of SCM. The proposed algorithm is mechanically similar to the original SCM, but is different in spirit in that it provides an approximation, rather than a lower bound, of the stability factor. Indeed, we sacrifice the rigor of the original SCM to enhance computational efficiency. Nevertheless, although this procedure enables a very rapid online evaluation of the stability factor, it still entails a quite expensive offline stage (especially when dealing with $P \geq 3$ parameters), which may jeopardize the efficiency of the whole reduction process, as shown by the numerical test cases of Sect. 2.6.

For this reason, we then propose some inexpensive, heuristic strategies to directly approximate the stability factor. These strategies combine a radial basis interpolant [Buh03] to the stability factor and an adaptive choice of interpolation points through a

greedy procedure. In this way, it is possible to obtain a reliable approximation of the stability factor, whose offline construction and online evaluation prove to be much faster than in the case of the linearized SCM algorithm.

We test the efficacy of these procedures by considering different flow problems which involve both physical and geometrical parameters. Moreover, in order to assess the robustness of the adaptive interpolation, we also consider a numerical test case whose solution features a bifurcation point – where the estimate of stability factors is critical. Hence, we also show that our heuristic technique proves to be effective when aiming at the detection of bifurcation points.

2.2 Stability factors for nonlinear, inf-sup stable parametrized PDEs

Given a regular spatial domain $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$, let $V = V(\Omega)$ be a Hilbert space with inner product $(\cdot, \cdot)_V$ and induced norm $\|v\|_V = \sqrt{(v, v)_V}$.

Although the techniques proposed in this chapter are suitable also for more general nonlinear problems, here we restrict our analysis in the case of stationary, quadratically nonlinear parametrized operators, for which our problem of interest can be expressed as follows: given $\boldsymbol{\mu} \in \mathcal{D}$, find $y = y(\boldsymbol{\mu}) \in V$ s.t.

$$A(y(\boldsymbol{\mu}); v; \boldsymbol{\mu}) = a(y(\boldsymbol{\mu}), v; \boldsymbol{\mu}) + c(y(\boldsymbol{\mu}), y(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \qquad \forall v \in V. \tag{2.3}$$

Here $a(\cdot, \cdot; \boldsymbol{\mu})$ is a continuous bilinear form over $V \times V$ and $c(\cdot, \cdot, \cdot; \boldsymbol{\mu})$ is a continuous trilinear form over $V \times V \times V$. Moreover, the right-hand side is a parametrized linear form $f(\cdot; \boldsymbol{\mu}) : V \to \mathbb{R}$, given by

$$f(v; \boldsymbol{\mu}) = {}_{V'}\langle F(\boldsymbol{\mu}), v\rangle_V,$$

being $F(\boldsymbol{\mu}) \in V'$ and $V' = \mathcal{L}(V; \mathbb{R})$ the dual space of V. According to the general Brezzi-Rappaz-Raviart (BRR) theory [BRR80], problem (2.3) is well posed if and only if the following continuity and inf-sup conditions hold:

$$\gamma(y(\boldsymbol{\mu})) = \sup_{v \in V} \sup_{w \in V} \frac{dA[y(\boldsymbol{\mu})](v, w; \boldsymbol{\mu})}{\|v\|_V \|w\|_V} < +\infty, \quad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{2.4}$$

$$\exists\ \beta^0(\boldsymbol{\mu}) > 0 :\ \beta(y(\boldsymbol{\mu})) = \inf_{v \in V} \sup_{w \in V} \frac{dA[y(\boldsymbol{\mu}](v, w; \boldsymbol{\mu})}{\|v\|_V \|w\|_V} \geq \beta^0(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{2.5}$$

In fact, these conditions ensure the existence of a *local branch of non-singular solutions* [GR86], see Proposition 2.1. Here $dA[y(\boldsymbol{\mu})](\cdot, \cdot; \boldsymbol{\mu})$ denotes the Fréchet derivative of $A(\cdot, \cdot; \boldsymbol{\mu})$ with respect to the first variable, which is given, at $z \in V$, by

$$dA[z](w, v; \boldsymbol{\mu}) = a(w, v; \boldsymbol{\mu}) + c(z, w, v; \boldsymbol{\mu}) + c(w, z, v; \boldsymbol{\mu}) \quad \forall v, w \in V; \tag{2.6}$$

from now on, we denote $d(z; \boldsymbol{\mu})(w, v) = c(z, w, v; \boldsymbol{\mu}) + c(w, z, v; \boldsymbol{\mu})$. Furthermore,

$$\gamma^a(\boldsymbol{\mu}) = \sup_{v \in V} \sup_{w \in V} \frac{a(v, w; \boldsymbol{\mu})}{\|v\|_V \|w\|_V} < +\infty,\ \gamma^c(\boldsymbol{\mu}) = \sup_{u \in V} \sup_{v \in V} \sup_{w \in V} \frac{c(u, v, w; \boldsymbol{\mu})}{\|u\|_V \|v\|_V \|w\|_V} < +\infty$$

denote the continuity constants of $a(\cdot,\cdot;\boldsymbol{\mu})$ and $c(\cdot,\cdot,\cdot;\boldsymbol{\mu})$, respectively, while

$$\gamma^d(\boldsymbol{\mu}) = \sup_{v\in V}\sup_{w\in V}\frac{d(y(\boldsymbol{\mu});\boldsymbol{\mu})(v,w)}{\|v\|_V\|w\|_V} < +\infty$$

denotes the continuity constant of $d(y;\boldsymbol{\mu})(\cdot,\cdot)$. The stability factor $\beta(\boldsymbol{\mu})$ we want to estimate obviously depends on the nonlinear form on the left-hand side of (2.3), and thus, through $y(\boldsymbol{\mu})$, on the right-hand side, too. This makes the accurate estimate of $\beta(\boldsymbol{\mu})$ much more involved than in the linear case.

In the following subsections we introduce some definitions and provide some basic results on the continuity and the regularity of the solution map $\boldsymbol{\mu}\mapsto y(\boldsymbol{\mu})$ that will be used in the sequel.

2.2.1 Supremizer operator, norms and parametric dependence

We introduce the parametrized linear operator $T^{\boldsymbol{\mu}}: V\to V$ such that, for any $\boldsymbol{\mu}\in\mathcal{D}$, $v\in V$,

$$(T^{\boldsymbol{\mu}}v,w)_V = dA[y(\boldsymbol{\mu})](v,w;\boldsymbol{\mu}) \quad \forall w\in V; \tag{2.7}$$

equivalently, by Riesz theorem,

$$T^{\boldsymbol{\mu}}v = \arg\sup_{w\in V}\frac{dA[y(\boldsymbol{\mu})](v,w;\boldsymbol{\mu})}{\|w\|_V}, \qquad \forall v\in V. \tag{2.8}$$

Because of (2.8), $T^{\boldsymbol{\mu}}$ is called *supremizer* operator. It follows that (2.4) and (2.5) can be equivalently expressed as

$$\gamma(\boldsymbol{\mu}) = \sup_{w\in V}\frac{\|T^{\boldsymbol{\mu}}w\|_V}{\|w\|_V}, \qquad \beta(\boldsymbol{\mu}) = \inf_{w\in V}\frac{\|T^{\boldsymbol{\mu}}w\|_V}{\|w\|_V}. \tag{2.9}$$

Assuming that $0<\beta^0(\boldsymbol{\mu})\le\beta(\boldsymbol{\mu})$ and $\gamma(\boldsymbol{\mu})<\infty$ for each $\boldsymbol{\mu}\in\mathcal{D}$, implies that

$$|||w|||_{\boldsymbol{\mu}} := \|T^{\boldsymbol{\mu}}w\|_V \qquad \forall w\in V, \tag{2.10}$$

defines a norm, usually referred to as *natural norm* [SVH+06, Dep08]. Thanks to (2.9), this latter is equivalent to the V-norm,

$$\frac{1}{\gamma(\boldsymbol{\mu})}\|T^{\boldsymbol{\mu}}w\|_V \le \|w\|_V \le \frac{1}{\beta(\boldsymbol{\mu})}\|T^{\boldsymbol{\mu}}w\|_V, \quad \forall w\in V. \tag{2.11}$$

In order to develop an offline-online strategy, we assume that the forms appearing in (2.3) fulfill the following *parameter separability* – also called *affine* parameter depedence – property: for any $u,v,w\in V$,

$$a(u,v;\boldsymbol{\mu}) = \sum_{q=1}^{Q_a}\theta_q^a(\boldsymbol{\mu})a_q(u,v), \qquad c(u,v,w;\boldsymbol{\mu}) = \sum_{q=1}^{Q_c}\theta_q^c(\boldsymbol{\mu})c_q(u,v,w) \tag{2.12}$$

for some integers Q_a, Q_c, where $\theta_q^a,\theta_q^c\in C^1(\mathcal{D})$ and $a_q(\cdot,\cdot)$, $c_q(\cdot,\cdot,\cdot)$ are continuous bilinear (trilinear) forms over $V\times V$ ($V\times V\times V$), respectively; moreover, we set $Q_A = Q_a + Q_c$. The requirement that θ_a^q, θ_c^q are of class $C^1(\mathcal{D})$ is essential to ensure that $\boldsymbol{\mu}\mapsto y(\boldsymbol{\mu})$ is a regular map, as we will see in Sect. 2.2.2.

Moreover, we denote the (now $\boldsymbol{\mu}$-independent) continuity constants of $a_q(\cdot,\cdot)$ and $c_q(\cdot,\cdot,\cdot)$ by

$$\gamma_q^a = \sup_{v\in V}\sup_{w\in V}\frac{a_q(v,w)}{\|v\|_V\|w\|_V} < +\infty, \qquad \gamma_q^c = \sup_{u\in V}\sup_{v\in V}\sup_{w\in V}\frac{c_q(u,v,w)}{\|u\|_V\|v\|_V\|w\|_V} < +\infty, \tag{2.13}$$

respectively. In the same way, we set

$$d(z;\boldsymbol{\mu})(w,v) = \sum_{q=1}^{Q_c}\theta_q^c(\boldsymbol{\mu})(c_q(z,w,v)+c_q(w,z,v)) = \sum_{q=1}^{Q_c}\theta_q^c(\boldsymbol{\mu})d_q(z)(v,w),$$

and denote by

$$\gamma_q^d(y) = \sup_{v\in V}\sup_{w\in V}\frac{d_q(y)(v,w)}{\|v\|_V\|w\|_V} < +\infty \tag{2.14}$$

the (y-dependent) continuity constants of $d_q(y)(\cdot,\cdot)$.

2.2.2 Fréchet derivatives of operators and regularity of solutions

Let us show some theoretical results required to ensure the well-posedness of the linearized SCM procedure. For the sake of generality, let us cast problem (2.3) under the form (2.1), where the operator $\mathcal{E}: V\times\mathcal{D}\to V'$ at a point $z\in V$ and parameter $\boldsymbol{\mu}\in\mathcal{D}$ is defined as

$${}_{V'}\langle\mathcal{E}(z;\boldsymbol{\mu}),w\rangle_V = A(z;w;\boldsymbol{\mu}) - f(w;\boldsymbol{\mu}) \qquad \forall z,w\in V. \tag{2.15}$$

Let us denote by $d_y\mathcal{E}(z;\boldsymbol{\mu}): V\to V'$ and $d_{\boldsymbol{\mu}}\mathcal{E}(z;\boldsymbol{\mu}):\mathcal{D}\to V'$ the (partial) Fréchet derivatives of $\mathcal{E}$ at $(z,\boldsymbol{\mu})\in\mathcal{D}\times V$. Moreover, we denote by $B_r(\boldsymbol{\mu})\subset\mathcal{D}$ the open ball with radius $r>0$ and center $\boldsymbol{\mu}\in\mathcal{D}$. First of all, we can state a general result ensuring that $\boldsymbol{\mu}\mapsto y(\boldsymbol{\mu})$ is a regular map:

Proposition 2.1. *For the parametrized operator $A(\cdot,\cdot;\boldsymbol{\mu}): V\times V\to\mathbb{R}$ defined in (2.3), suppose that:*

1. *the continuity and the inf-sup conditions (2.4)–(2.5) hold;*
2. *the parameter separability assumption (2.12) holds, being $\theta_q^a,\theta_{q'}^c:\mathcal{D}\to\mathbb{R}$, $q=1,\dots,Q_a$, $q'=1,\dots,Q_c$, prescribed C^1 functions.*

Moreover assume that $\mathcal{E}(y_0;\boldsymbol{\mu}_0)=0$ for some $\boldsymbol{\mu}_0\in\mathcal{D}$, $y_0\in V$. Then, there exist $r_0,r>0$ and a unique $y(\boldsymbol{\mu})\in B_r(u_0)\cap V$ such that

$$\mathcal{E}(y(\boldsymbol{\mu});\boldsymbol{\mu}) = 0 \qquad \forall\boldsymbol{\mu}\in B_{r_0}(\boldsymbol{\mu}_0)\cap\mathcal{D}.$$

Furthermore, the map $\boldsymbol{\mu}\mapsto y(\boldsymbol{\mu})$ is Lipschitz continuous and

$$\frac{\partial y(\boldsymbol{\mu})}{\partial\boldsymbol{\mu}} = -\left(d_u\mathcal{E}(y(\boldsymbol{\mu});\boldsymbol{\mu})\right)^{-1} d_{\boldsymbol{\mu}}\mathcal{E}(y(\boldsymbol{\mu});\boldsymbol{\mu}).$$

Proof. The proof is a direct consequence of the Implicit Function Theorem: we refer here to the version stated by Hildebrandt and Graves [HG27], see also [Zei85]. A very general version providing further insights on the Lipschitz constant of the map $\boldsymbol{\mu}\mapsto y(\boldsymbol{\mu})$ can be found, e.g., in [CR97, IK08]. Provided that the continuity condition (2.4) holds, $d_y\mathcal{E}$

is continuous at each point $(y_0, \boldsymbol{\mu}_0) \in V \times \mathcal{D}$; its inverse is a continuous linear operator thanks to the inf-sup condition (2.5) – in other words, $d_y\mathcal{E}$ is an isomorphism, for any $(y_0, \boldsymbol{\mu}_0) \in V \times \mathcal{D}$. Furthermore, if the parameter separability assumption (2.12) holds, for suitable C^1 functions $\theta_q^a, \theta_{q'}^c : \mathcal{D} \to \mathbb{R}$, $q = 1, \dots, Q_a$, $q' = 1, \dots, Q_c$, then $\mathcal{E}$ is a C^1 map. Then, the Implicit Function Theorem ensures the existence of $r_0, r > 0$ and of a (unique) C^1 map $\boldsymbol{\mu} \mapsto y(\boldsymbol{\mu})$ such that, for every $\boldsymbol{\mu} \in B_{r_0}(\boldsymbol{\mu}_0) \cap \mathcal{D}$, $\mathcal{E}(y(\boldsymbol{\mu}), \boldsymbol{\mu}) = 0$. □

Exploiting the result above, we can show that also the Fréchet derivative of $A(\cdot, \cdot; \boldsymbol{\mu})$ defines a regular map, provided that suitable *a priori* (or energy) estimates on $\|y(\boldsymbol{\mu})\|_V$ hold. In the same way, also the supremizer operator defines a Lipschitz-continuous map with respect to parameter variations.

Proposition 2.2. *Under the assumptions of Proposition 2.1, and by additionally assuming that*

$$\exists K_y > 0 \quad s.t. \quad \|y(\boldsymbol{\mu})\|_V \leq K_y \qquad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{2.16}$$

there exists a positive constant $C > 0$ such that, for any $\boldsymbol{\mu}, \boldsymbol{\mu}^* \in \mathcal{D}$, $v, w \in V$

$$\Big| dA[y(\boldsymbol{\mu})](v, w; \boldsymbol{\mu}) - dA[y(\boldsymbol{\mu}^*)](v, w; \boldsymbol{\mu}^*) \Big| \leq C|\boldsymbol{\mu} - \boldsymbol{\mu}^*| \|v\|_V \|w\|_V. \tag{2.17}$$

Furthermore, the following estimate holds:

$$\|T^{\boldsymbol{\mu}} w - T^{\boldsymbol{\mu}^*} w\|_V \leq \frac{C}{\beta(\boldsymbol{\mu}^*)} \, |\boldsymbol{\mu} - \boldsymbol{\mu}^*| \, \|T^{\boldsymbol{\mu}^*} w\|_V \qquad \forall w \in V. \tag{2.18}$$

Proof. From the definition (2.6) of $dA(\cdot; \boldsymbol{\mu})(\cdot, \cdot)$ and the affine decomposition (2.12), we have that

$$\begin{aligned} &dA[y(\boldsymbol{\mu})](v, w; \boldsymbol{\mu}) - dA[y(\boldsymbol{\mu}^*](v, w; \boldsymbol{\mu}^*) \\ &= \underbrace{\sum_{q=1}^{Q_a} [\theta_q^a(\boldsymbol{\mu}) - \theta_q^a(\boldsymbol{\mu}^*)] a_q(v, w)}_{\text{(I)}} + \underbrace{\sum_{q=1}^{Q_c} \theta_q^c(\boldsymbol{\mu}) d_q(y(\boldsymbol{\mu}), v, w) - \sum_{q=1}^{Q_c} \theta_q^c(\boldsymbol{\mu}^*) d_q(y(\boldsymbol{\mu}^*), v, w)}_{\text{(II)}}. \end{aligned}$$

The first term can be easily bounded as

$$|\text{(I)}| \leq Q_a L_a |\boldsymbol{\mu} - \boldsymbol{\mu}^*| \, \bar{\gamma}_a \|v\|_V \|w\|_V, \tag{2.19}$$

where $L_a = \max_{q=1,\dots,Q_a} L_a^q$, being the L_a^q's the Lipschitz constants of the functions $\theta_q^a(\cdot)$, while $\bar{\gamma}_a = \max_{q=1,\dots,Q_a} \gamma_a^q$, being the γ_a^q's the continuity constants of the bilinear forms $a_q(\cdot, \cdot)$. Let us now rewrite the second term as

$$\text{(II)} = \sum_{q=1}^{Q_c} \Big[\theta_q^c(\boldsymbol{\mu}) - \theta_q^c(\boldsymbol{\mu}^*) \Big] d_q(y(\boldsymbol{\mu}), v, w) + \sum_{q=1}^{Q_c} \theta_q^c(\boldsymbol{\mu}^*) d_q(y(\boldsymbol{\mu}) - y(\boldsymbol{\mu}^*), v, w),$$

which can be bounded as

$$|\text{(II)}| \leq Q_c \, L_c \, |\boldsymbol{\mu} - \boldsymbol{\mu}^*| \, \bar{\gamma}_d \, \|v\|_V \|w\|_V \|u(\boldsymbol{\mu})\|_V + Q_c \, M_\theta^c \, \bar{\gamma}_d \, \|v\|_V \|w\|_V \|y(\boldsymbol{\mu}) - y(\boldsymbol{\mu}^*)\|_V.$$

Here $L_c = \max_{q=1,\dots,Q_c} L_c^q$, being the L_c^q's the Lipschitz constants of the functions $\theta_q^c(\cdot)$, $\bar{\gamma}_d$ is the larger among the continuity constants of the trilinear forms $d_q(\cdot,\cdot,\cdot)$, and

$$M_\theta^c = \max_{\boldsymbol{\mu}\in\mathcal{D}} \max_{q=1,\dots,Q_c} \theta_q^c(\boldsymbol{\mu}).$$

Since the solution $y(\boldsymbol{\mu})$ of problem (2.3) is bounded for every $\boldsymbol{\mu} \in \mathcal{D}$ – thanks to (2.16) – and Lipschitz continuous with respect to $\boldsymbol{\mu}$ (see Proposition 2.1), there exist positive constants K_y and L_y such that

$$\|y(\boldsymbol{\mu})\|_V \le K_y, \qquad \|y(\boldsymbol{\mu}) - y(\boldsymbol{\mu}^*)\|_V \le L_y|\boldsymbol{\mu} - \boldsymbol{\mu}^*|, \tag{2.20}$$

uniformly in $\mathcal{D}$. Therefore,

$$|(\mathrm{II})| \le \Big(L_c K_y + M_\theta^c L_y\Big) Q_c\, \bar{\gamma}_d |\boldsymbol{\mu} - \boldsymbol{\mu}^*| \|v\|_V \|w\|_V. \tag{2.21}$$

Combining (2.19) and (2.21), in the end we obtain (2.17) with constant

$$C = Q_a L_a \bar{\gamma}_a + Q_c \bar{\gamma}_d \,(L_c K_y + M_\theta^c L_y).$$

Furthermore, we have

$$\begin{aligned}
\|T^{\boldsymbol{\mu}} w \;-\; T^{\boldsymbol{\mu}^*} w\|_V^2 =\; & (T^{\boldsymbol{\mu}} w - T^{\boldsymbol{\mu}^*} w, T^{\boldsymbol{\mu}} w - T^{\boldsymbol{\mu}^*} w)_V \\
&= dA(y(\boldsymbol{\mu}); \boldsymbol{\mu})(w, T^{\boldsymbol{\mu}} w - T^{\boldsymbol{\mu}^*} w) - dA(y(\boldsymbol{\mu}^*); \boldsymbol{\mu}^*)(w, T^{\boldsymbol{\mu}} w - T^{\boldsymbol{\mu}^*} w) \\
&\le C|\boldsymbol{\mu} - \boldsymbol{\mu}^*|\, \|w\|_V \|T^{\boldsymbol{\mu}} w - T^{\boldsymbol{\mu}^*} w\|_V \le \frac{C}{\beta(\boldsymbol{\mu}^*)} |\boldsymbol{\mu} - \boldsymbol{\mu}^*|\, \|T^{\boldsymbol{\mu}^*} w\|_V \|T^{\boldsymbol{\mu}} w - T^{\boldsymbol{\mu}^*} w\|_V
\end{aligned}$$

by exploiting (2.17) and (2.11), from which we obtain,

$$\|T^{\boldsymbol{\mu}} w - T^{\boldsymbol{\mu}^*} w\|_V \le \frac{C}{\beta(\boldsymbol{\mu}^*)} |\boldsymbol{\mu} - \boldsymbol{\mu}^*|\, \|T^{\boldsymbol{\mu}^*} w\|_V \qquad \forall w \in V. \qquad \square$$

Remark 2.1. In the Navier-Stokes case, an *a priori* estimate like (2.16) can obtained by using the coercivity of the bilinear form $a(\cdot,\cdot;\boldsymbol{\mu})$ and the skew-symmetry (with respect to the last two arguments) of the trilinear form $c(\cdot,\cdot,\cdot;\boldsymbol{\mu})$; see e.g. [Tem01, Sect. 2.1]. •

2.3 High-fidelity and reduced approximation

In this section we introduce the high-fidelity approximation of problem (2.3), based on a Galerkin-Finite Element (FE) method, and then a lower-fidelity approximation based on the RB method. Moreover, we discuss some stability issues related with these two approximation strategies, and recall a general a posteriori error estimate, where the role of stability factors is highlighted.

2.3.1 Finite element approximation

Let us denote by $V_h \subset V$ a FE approximation space of dimension N_h, with inherited inner product $(v,w)_{V_h} = (v,w)_V$ and norm $\|v\|_{V_h} = \|v\|_V$. The Galerkin-FE approximation of (2.3) reads as follows: given $\boldsymbol{\mu} \in \mathcal{D}$, find $y_h(\boldsymbol{\mu}) \in V_h$ s.t.

$$A(y_h(\boldsymbol{\mu}); v_h; \boldsymbol{\mu}) = f(v_h; \boldsymbol{\mu}) \qquad \forall v_h \in V_h. \tag{2.22}$$

Problem (2.22) is equivalent to the following algebraic nonlinear system: given $\boldsymbol{\mu} \in \mathcal{D}$, find $\mathbf{y}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ such that

$$\Big(\mathbf{K}(\boldsymbol{\mu}) + \mathbf{C}(\mathbf{y}_h(\boldsymbol{\mu}); \boldsymbol{\mu})\Big)\mathbf{y}_h(\boldsymbol{\mu}) = \mathbf{f}(\boldsymbol{\mu}) \qquad \text{in } \mathbb{R}^{N_h}. \tag{2.23}$$

Here $\mathbf{y}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is the vector representation of $y_h(\boldsymbol{\mu}) \in V_h$ over a Lagrangian basis $\{\varphi_j^h\}_{j=1}^{N_h}$ of V_h, i.e. given a set of Lagrange nodes $\{\mathbf{x}_j\}_{j=1}^{N_h}$, $\mathbf{v}_h^{(j)} = v_h(\mathbf{x}_j)$ for any $v_h \in V_h$. Moreover,

$$(\mathbf{K}(\boldsymbol{\mu}))_{ij} = a(\varphi_j^h, \varphi_i^h; \boldsymbol{\mu}), \quad (\mathbf{C}(\mathbf{w}_h; \boldsymbol{\mu}))_{ij} = c(w_h, \varphi_i^h, \varphi_j^h; \boldsymbol{\mu}), \quad \mathbf{f}^{(j)}(\boldsymbol{\mu}) = f(\varphi_h^j; \boldsymbol{\mu})$$

are the matrices corresponding to the linear and the nonlinear term, and the vector corresponding to the source term, respectively $(i, j = 1, \ldots, N_h)$.

Concerning the stability of the approximation (2.22), we rely on the Brezzi-Rappaz-Raviart theory [BRR80, CR97]. As for the original problem, let us assume that[1]:

$$\gamma_h(y_h(\boldsymbol{\mu})) = \sup_{v_h \in V_h} \sup_{w_h \in V_h} \frac{dA[y_h(\boldsymbol{\mu})](v_h, w_h; \boldsymbol{\mu})}{\|v_h\|_V \|w_h\|_V} < +\infty, \quad \forall\, \boldsymbol{\mu} \in \mathcal{D}, \tag{2.24}$$

$$\exists\, \beta_h^0(\boldsymbol{\mu}) > 0 : \ \beta_h(y_h(\boldsymbol{\mu})) = \inf_{v_h \in V_h} \sup_{w_h \in V_h} \frac{dA[y_h(\boldsymbol{\mu})](v_h, w_h; \boldsymbol{\mu})}{\|v_h\|_V \|w_h\|_V} \geq \beta_h^0(\boldsymbol{\mu}), \quad \forall\, \boldsymbol{\mu} \in \mathcal{D}. \tag{2.25}$$

Then, if V_h is chosen so to satisfy these conditions – which are, in fact, the discrete version of (2.4)–(2.5) – problem (2.22) admits a unique solution.

Concerning the regularity of the solution with respect to $\boldsymbol{\mu}$, a result similar to that of Proposition 2.1 can be proved if we consider a Galerkin approximation, i.e. find $y_h \in V_h$ s.t.

$${}_{V'}\langle \mathcal{E}(y_h; \boldsymbol{\mu}), w_h \rangle_V = 0 \qquad \forall w_h \in V_h, \tag{2.26}$$

where V_h is such that (2.24) and (2.25) hold (see e.g. [CR97, Chapter 12 and Remark 13.2]). For instance, Taylor-Hood elements [GR86] allow to meet these requirements in the Navier-Stokes case.

By introducing the discrete supremizer operator[2] $T^{\boldsymbol{\mu}} : V_h \to V_h$ s.t.

$$(T^{\boldsymbol{\mu}} v_h, w_h)_V = dA[y_h(\boldsymbol{\mu})](v_h, w_h; \boldsymbol{\mu}) \quad \forall v_h, w_h \in V_h, \tag{2.27}$$

we have that

$$(\beta_h(y_h(\boldsymbol{\mu})))^2 = \left(\inf_{v \in V_h} \frac{dA[y_h(\boldsymbol{\mu})](v, T^{\boldsymbol{\mu}} v; \boldsymbol{\mu}))}{\|v\|_V \|T^{\boldsymbol{\mu}} v\|_V} \right)^2 = \inf_{v \in V_h} \frac{\|T^{\boldsymbol{\mu}} v\|_V^2}{\|v\|_V^2}. \tag{2.28}$$

The algebraic counterpart of (2.28) can be obtained by introducing the matrix norm

$$\mathbf{X}_{ij} = (\varphi_j^h, \varphi_i^h)_V, \qquad i, j = 1, \ldots, N_h \tag{2.29}$$

[1] In the following we denote $\beta_h(y_h(\boldsymbol{\mu}))$ by $\beta_h(\boldsymbol{\mu})$, i.e. we omit the dependence on the solution, wherever it is clear from the context.

[2] For the sake of notation, we denote by $T^{\boldsymbol{\mu}}$ the discrete supremizer operator, too.

of V_h, so that $\|v_h\|_V^2 = \mathbf{v}_h^T \mathbf{X} \mathbf{v}_h$ for any $v_h \in V_h$. Moreover, denoting by $\mathbf{t}_h$ the vector of components $\mathbf{t}_h^{(i)} = (T^{\boldsymbol{\mu}} v_h)(\mathbf{x}_i)$, we have that

$$\mathbf{w}_h^T \mathbf{X} \mathbf{t}_h = \mathbf{w}_h^T \mathbf{A}(\boldsymbol{\mu}) \mathbf{v}_h, \tag{2.30}$$

being $\mathbf{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ the matrix corresponding to the FE discretization of the Fréchet derivative, i.e.

$$(\mathbf{A}(\boldsymbol{\mu}))_{ij} = dA[y_h(\boldsymbol{\mu})](\varphi_i^h, \varphi_j^h; \boldsymbol{\mu})), \qquad i, j = 1, \ldots, N_h. \tag{2.31}$$

Then,

$$\beta_h(y_h(\boldsymbol{\mu})) = \inf_{\mathbf{v}_h \in \mathbb{R}^{N_h}} \frac{\mathbf{v}_h^T \mathbf{A}^T(\boldsymbol{\mu}) \mathbf{X}^{-1} \mathbf{A}(\boldsymbol{\mu}) \mathbf{v}_h}{\mathbf{v}_h^T \mathbf{X} \mathbf{v}_h}, \tag{2.32}$$

from (2.28) and (2.30), so that $\beta_h(y_h(\boldsymbol{\mu})) = (\lambda_{\min}(\boldsymbol{\mu}))^{1/2}$, where $\lambda_{\min}(\boldsymbol{\mu})$ is the smallest eigenvalue $\lambda(\boldsymbol{\mu})$ such that $(\lambda(\boldsymbol{\mu}), \mathbf{v}_h) \in \mathbb{R}_+ \times V_h$, $\mathbf{v}_h \neq 0$, satisfy

$$\mathbf{A}^T(\boldsymbol{\mu})\, \mathbf{X}^{-1}\, \mathbf{A}(\boldsymbol{\mu})\, \mathbf{v}_h = \lambda(\boldsymbol{\mu}) \mathbf{X}\, \mathbf{v}_h. \tag{2.33}$$

Equivalently,

$$\beta_h(y_h(\boldsymbol{\mu})) = \|\mathbf{X}^{1/2} \mathbf{A}(\boldsymbol{\mu})^{-1} \mathbf{X}^{1/2}\|_2^{-1} = \sigma_{\min}\left(\mathbf{X}^{-1/2} \mathbf{A}(\boldsymbol{\mu}) \mathbf{X}^{-1/2}\right). \tag{2.34}$$

Thus, the evaluation of the stability factor $\beta_h(y_h(\boldsymbol{\mu}))$, for any $\boldsymbol{\mu} \in \mathcal{D}$, entails the solution of both the nonlinear algebraic system (2.23) and the eigenvalue problem (2.33).

2.3.2 Reduced basis approximation

Our final goal is to compute, for any $\boldsymbol{\mu} \in \mathcal{D}$, a RB approximation $y_N(\boldsymbol{\mu}) \in V_N$ to $y_h(\boldsymbol{\mu})$, where

$$V_N = \text{span}\{y_h(\boldsymbol{\mu}^1), \ldots, y_h(\boldsymbol{\mu}^N)\} \subset V_h \tag{2.35}$$

is a reduced basis space, made by $N \ll N_h$ solutions to problem (2.22) computed for properly chosen parameter values $\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^N$. Then, we perform the Gram-Schmidt procedure on the snapshots to obtain an orthonormal basis $\{\phi_1, \ldots, \phi_N\}$, so that we have $y_N(\boldsymbol{\mu}) = \sum_{j=1}^N u_N^{(j)}(\boldsymbol{\mu}) \phi_j$, where the components $\{u_N^{(j)}\}_{j=1}^N$ are computed through a Galerkin[3] projection of (2.3) over V_N: find $y_N(\boldsymbol{\mu}) \in V_N$ s.t.

$$A(y_N(\boldsymbol{\mu}); v_N; \boldsymbol{\mu}) = f(v_N; \boldsymbol{\mu}) \qquad \forall v_N \in V_N. \tag{2.36}$$

Proving the stability of (2.36) would demand to prove that an inf-sup condition holds. This however could be challenging, see, e.g., [Man14].

Here, we rather look for an estimate of the discrete inf-sup stability factor

$$\beta_h(y_N(\boldsymbol{\mu})) = \inf_{v \in V_h} \sup_{w \in V_h} \frac{dA[y_N(\boldsymbol{\mu})](v, w; \boldsymbol{\mu})}{\|v\|_V \|w\|_V} \qquad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{2.37}$$

[3]For the sake of simplicity, here we restrict ourselves to the case of Galerkin projection, although sometimes a more general Petrov-Galerkin method is used.

which enters in the following a posteriori error bound: for any $N \geq N^*(\boldsymbol{\mu})$,

$$\|y_h(\boldsymbol{\mu}) - y_N(\boldsymbol{\mu})\|_V \leq \frac{2}{\beta_h(y_N(\boldsymbol{\mu}))}\|r(\cdot;\boldsymbol{\mu})\|_{V_h'} \qquad \forall\, \boldsymbol{\mu} \in \mathcal{D}, \tag{2.38}$$

where

$$r(w;\boldsymbol{\mu}) = A(y_N(\boldsymbol{\mu}); w; \boldsymbol{\mu}) - f(w;\boldsymbol{\mu}) \qquad \forall w \in V_h$$

is the residual, $N^*(\boldsymbol{\mu})$ the smallest N such that $\tau_N(\boldsymbol{\mu}) < 1$, for all $N \geq N^*(\boldsymbol{\mu})$, $\tau_N(\boldsymbol{\mu})$ is defined as

$$\tau_N(\boldsymbol{\mu}) = \frac{4\gamma_h^c(\boldsymbol{\mu})\|r(\cdot;\boldsymbol{\mu})\|_{V_h'}}{\beta_h(y_N(\boldsymbol{\mu}))^2},$$

and $\gamma_h^c(\boldsymbol{\mu})$ is the discrete continuity constant of $c(\cdot,\cdot,\cdot;\boldsymbol{\mu})$. See e.g. [VP05, Dep08, Man14] for further details and proofs in the Navier-Stokes case, or [CTU09, RG12] for recent applications to nonlinear advection-diffusion problems.

Since $\beta_h(y_N(\boldsymbol{\mu}))$ depends on the RB solution, computing a parametric lower bound for the stability factor *before* assembling the reduced space is *infeasible*, unless these two procedures are run simultaneously, as shown e.g. in [Yan14]. This latter option however suffers from two crucial limitations: it requires a problem-specific and intrusive implementation of the two procedures, and worsen the computational complexity of the construction and evaluation of the lower bound by making it dependent on N (see [Yan14] for further details). We avoid this extra burden by looking for a convenient approximation of the stability factor $\beta_h(y_h(\boldsymbol{\mu}))$ defined in (2.25), rather than seeking a lower bound to the stability factor $\beta_h(y_N(\boldsymbol{\mu}))$. In fact, the former quantity provides an *asymptotically* good approximation to the latter, thanks to

Proposition 2.3. *The following relation holds:*

$$|\beta_h(y_h(\boldsymbol{\mu})) - \beta_h(y_N(\boldsymbol{\mu}))| \leq 2\gamma_h^c(\boldsymbol{\mu})\|y_h(\boldsymbol{\mu}) - y_N(\boldsymbol{\mu})\|_V \qquad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{2.39}$$

Proof. By exploiting the trilinearity and the continuity of $c(\cdot,\cdot,\cdot;\boldsymbol{\mu})$, we have

$$\begin{aligned}
\beta_h(y_1) &= \inf_{v\in V_h}\sup_{w\in V_h}\left(\frac{a(v,w;\boldsymbol{\mu}) + c(y_2,v,w;\boldsymbol{\mu}) + c(v,y_2,w;\boldsymbol{\mu})}{\|v\|_V\|w\|_V}\right. \\
&\qquad\qquad \left. + \frac{c(y_1-y_2,v,w;\boldsymbol{\mu}) + c(v,y_1-y_2,w;\boldsymbol{\mu})}{\|v\|_V\|w\|_V}\right) \\
&= \inf_{v\in V_h}\sup_{w\in V_h}\frac{dA[y_2](v,w;\boldsymbol{\mu}) + c(y_1-y_2,v,w;\boldsymbol{\mu}) + c(v,y_1-y_2,w;\boldsymbol{\mu})}{\|v\|_V\|w\|_V} \\
&\leq \inf_{v\in V_h}\sup_{w\in V_h}\frac{dA[y_2](v,w;\boldsymbol{\mu})}{\|v\|_V\|w\|_V} + 2\gamma_h^c(\boldsymbol{\mu})\|y_1-y_2\|_V \\
&= \beta_h(y_2) + 2\gamma_h^c(\boldsymbol{\mu})\|y_1-y_2\|_V.
\end{aligned}$$

By considering in the previous inequality first $y_1 = y_N(\boldsymbol{\mu})$, $y_2 = y_h(\boldsymbol{\mu})$, and then $y_2 = y_N(\boldsymbol{\mu})$, $y_1 = y_h(\boldsymbol{\mu})$, (2.39) easily follows. □

Although (2.38) cannot be used to estimate $\|y_h(\boldsymbol{\mu}) - y_N(\boldsymbol{\mu})\|_V$, thanks to the approximation property of the space V_N, (2.39) can be regarded as an *a priori* convergence

result. In fact, provided the RB approximation $y_N(\boldsymbol{\mu})$ is sufficiently close to $y_h(\boldsymbol{\mu})$ (which is the case for N sufficiently large), the stability factor $\beta_h(y_N(\boldsymbol{\mu}))$ related to the former can be properly approximated by the stability factor $\beta_h(y_h(\boldsymbol{\mu}))$ related to the latter, making thus possible to estimate the stability factor before assembling the reduced space.

We also remark that a result like (2.39) holds in case of a general nonlinear operator as long as its Fréchet derivative is Lipschitz continuous, i.e. if there exist $\eta(\boldsymbol{\mu}) > 0$, $L_h^N(\boldsymbol{\mu}) > 0$ such that

$$\|dA[y_N(\boldsymbol{\mu})](\cdot,\cdot;\boldsymbol{\mu})) - dA[v](\cdot,\cdot;\boldsymbol{\mu}))\|_{\mathcal{L}(V_h,V_h')} \leq L_h^N(\boldsymbol{\mu})\|y_N(\boldsymbol{\mu}) - v\|_V,$$

holds for all $v \in B_{\eta(\boldsymbol{\mu})}(y_N(\boldsymbol{\mu})) = \{w \in V_h : \|y_N(\boldsymbol{\mu}) - w\|_V \leq \eta(\boldsymbol{\mu})\}$. Then,

$$|\beta_h(y_h(\boldsymbol{\mu})) - \beta_h(y_N(\boldsymbol{\mu}))| \leq L_h^N(\boldsymbol{\mu})\|y_h(\boldsymbol{\mu}) - y_N(\boldsymbol{\mu})\|_V \qquad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{2.40}$$

2.4 A linearized SCM for estimating the stability factor

In this section we provide a linearized version of the Successive Constraint Method (SCM) [HKC+10] to compute an estimate of the stability factor $\beta_h(y_h(\boldsymbol{\mu}))$. Following [SVH+06, HKC+10], we adopt a *natural norm* SCM procedure based on a set of local stability factors, properly computed for a (possibly small) set of J parameter values $\mathcal{S} = \{\boldsymbol{\mu}^{1*}, \ldots, \boldsymbol{\mu}^{J*}\}$ selected through a greedy procedure. The key observation is provided by the following relation:

$$\begin{aligned}
\beta_h(y_h(\boldsymbol{\mu})) &= \inf_{v\in V_h} \sup_{w\in V_h} \frac{dA[y_h(\boldsymbol{\mu})](v,w;\boldsymbol{\mu})}{\|T^{\boldsymbol{\mu}^*}w\|_V\|v\|_V} \frac{\|T^{\boldsymbol{\mu}^*}w\|_V}{\|w\|_V} \\
&\geq \inf_{v\in V_h} \sup_{w\in V_h} \frac{dA[y_h(\boldsymbol{\mu}](v,w;\boldsymbol{\mu})}{\|T^{\boldsymbol{\mu}^*}w\|_V\|v\|_V} \inf_{w\in V_h} \frac{\|T^{\boldsymbol{\mu}^*}w\|_V}{\|w\|_V} \\
&= \beta_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})\beta_h(y_h(\boldsymbol{\mu}^*)) \geq \tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})\beta_h(y_h(\boldsymbol{\mu}^*)),
\end{aligned}$$

where

$$\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) := \inf_{v\in V_h} \frac{dA[y_h(\boldsymbol{\mu})](v, T^{\boldsymbol{\mu}^*}v;\boldsymbol{\mu})}{\|T^{\boldsymbol{\mu}^*}v\|_V^2} = \inf_{v\in V_h} \frac{(T^{\boldsymbol{\mu}}v, T^{\boldsymbol{\mu}^*}v)_V}{\|T^{\boldsymbol{\mu}^*}v\|_V^2} \tag{2.41}$$

is a lower bound of $\beta_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$. As a matter of fact,

$$\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) \leq \inf_{v\in V_h} \frac{\|T^{\boldsymbol{\mu}}v\|_V}{\|T^{\boldsymbol{\mu}^*}v\|_V} = \inf_{v\in V_h} \sup_{w\in V_h} \frac{dA[y_h(\boldsymbol{\mu})](v,w;\boldsymbol{\mu})}{\|T^{\boldsymbol{\mu}^*}v\|_V\|w\|_V} =: \beta_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}), \tag{2.42}$$

thanks to Cauchy-Schwarz inequality and the definition of supremizer operator.

As in the linear case [SVH+06], we can show that, for $\boldsymbol{\mu}$ near $\boldsymbol{\mu}^* \in \mathcal{S}$, $\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ is a second-order accurate approximation to $\beta_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$.

Proposition 2.4. *Under the assumptions of Proposition 2.2, the following relations hold:*

$$\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) - 1 = O(|\boldsymbol{\mu} - \boldsymbol{\mu}^*|) \qquad \text{as } \boldsymbol{\mu} \to \boldsymbol{\mu}^*, \tag{2.43}$$

$$\beta_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) - \tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) = O(|\boldsymbol{\mu} - \boldsymbol{\mu}^*|^2) \qquad \text{as } \boldsymbol{\mu} \to \boldsymbol{\mu}^*. \tag{2.44}$$

Proof. In order to show (2.43), we start by observing that

$$\begin{aligned}(T^{\boldsymbol{\mu}}v, T^{\boldsymbol{\mu}^*}v)_V &= (T^{\boldsymbol{\mu}}v - T^{\boldsymbol{\mu}^*}v + T^{\boldsymbol{\mu}^*}v, T^{\boldsymbol{\mu}^*}v)_V = \|T^{\boldsymbol{\mu}^*}v\|_V^2 + (T^{\boldsymbol{\mu}}v - T^{\boldsymbol{\mu}^*}v, T^{\boldsymbol{\mu}^*}v)_V \\ &= \|T^{\boldsymbol{\mu}^*}v\|_V^2 + dA[y_h(\boldsymbol{\mu})](v, T^{\boldsymbol{\mu}^*}v; \boldsymbol{\mu}) - dA[y_h(\boldsymbol{\mu}^*)](v, T^{\boldsymbol{\mu}^*}v; \boldsymbol{\mu}^*)\end{aligned}$$

so that

$$\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) \leq 1 + \inf_{v \in V_h} \frac{|dA[y_h(\boldsymbol{\mu})](v, T^{\boldsymbol{\mu}^*}v; \boldsymbol{\mu}) - dA[y_h(\boldsymbol{\mu}^*)](v, T^{\boldsymbol{\mu}^*}v; \boldsymbol{\mu}^*)|}{\|T^{\boldsymbol{\mu}^*}v\|_V^2}.$$

In order to bound this quantity, we exploit the result (2.17) of Proposition 2.2, which is valid for any $v, w \in V_h$, too (see e.g. [CR97, Remark 13.2]). Thus, for any $v \in V_h$,

$$\begin{aligned}&|dA[y_h(\boldsymbol{\mu})](v, T^{\boldsymbol{\mu}^*}v; \boldsymbol{\mu}) - dA[y_h(\boldsymbol{\mu}^*)](v, T^{\boldsymbol{\mu}^*}v; \boldsymbol{\mu}^*)| \\ &\qquad \leq C|\boldsymbol{\mu} - \boldsymbol{\mu}^*|\|v\|_V\|T^{\boldsymbol{\mu}^*}v\|_V \leq \frac{C}{\beta(\boldsymbol{\mu}^*)}|\boldsymbol{\mu} - \boldsymbol{\mu}^*|\|T^{\boldsymbol{\mu}^*}v\|_V^2,\end{aligned}$$

so that

$$\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) \leq 1 + \frac{C}{\beta(\boldsymbol{\mu}^*)}|\boldsymbol{\mu} - \boldsymbol{\mu}^*|$$

or, equivalently, $\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) - 1 = O(|\boldsymbol{\mu} - \boldsymbol{\mu}^*|)$ as $\boldsymbol{\mu} \to \boldsymbol{\mu}^*$. In order to show (2.44), we first expand

$$\begin{aligned}\beta_{\boldsymbol{\mu}^*}^2(\boldsymbol{\mu}) &= \inf_{v \in V_h} \frac{\|T^{\boldsymbol{\mu}}v\|_V^2}{\|T^{\boldsymbol{\mu}^*}v\|_V^2} = \inf_{v \in V_h} \frac{(T^{\boldsymbol{\mu}^*}v + (T^{\boldsymbol{\mu}}v - T^{\boldsymbol{\mu}^*}v), T^{\boldsymbol{\mu}^*}v + (T^{\boldsymbol{\mu}}v - T^{\boldsymbol{\mu}^*}v))_V}{\|T^{\boldsymbol{\mu}^*}v\|_V^2} \\ &= 1 + \inf_{v \in V_h} \left(2\frac{(T^{\boldsymbol{\mu}}v - T^{\boldsymbol{\mu}^*}v, T^{\boldsymbol{\mu}^*}v)_V}{\|T^{\boldsymbol{\mu}^*}v\|_V^2} + \frac{\|T^{\boldsymbol{\mu}}v - T^{\boldsymbol{\mu}^*}v\|_V^2}{\|T^{\boldsymbol{\mu}^*}v\|_V^2}\right).\end{aligned}$$

Thanks to (2.18), we have

$$\frac{\|T^{\boldsymbol{\mu}}v - T^{\boldsymbol{\mu}^*}v\|_V^2}{\|T^{\boldsymbol{\mu}^*}v\|_V^2} \leq \frac{C^2}{\beta^2(\boldsymbol{\mu}^*)}|\boldsymbol{\mu} - \boldsymbol{\mu}^*|^2, \qquad \forall v \in V_h$$

and by recognizing that

$$\inf_{v \in V_h} \frac{(T^{\boldsymbol{\mu}}v - T^{\boldsymbol{\mu}^*}v, T^{\boldsymbol{\mu}^*}v)_V}{\|T^{\boldsymbol{\mu}^*}v\|_V^2} = \tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) - 1,$$

we end up with $\beta_{\boldsymbol{\mu}^*}^2(\boldsymbol{\mu}) = 1 + 2(\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) - 1) + O(|\boldsymbol{\mu} - \boldsymbol{\mu}^*|^2)$ as $\boldsymbol{\mu} \to \boldsymbol{\mu}^*$. By taking the square root, using a Taylor series expansion and the fact that $O(\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) - 1) = O(|\boldsymbol{\mu} - \boldsymbol{\mu}^*|)$ thanks to (2.43), we obtain:

$$\begin{aligned}&\sqrt{-1 + 2\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) + O(|\boldsymbol{\mu} - \boldsymbol{\mu}^*|^2)} = \\ &\qquad = 1 + \frac{1}{2}\left(2\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) - 2 + O(|\boldsymbol{\mu} - \boldsymbol{\mu}^*|^2)\right) \\ &\qquad\qquad - \frac{1}{8}\left(2\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) - 2 + O(|\boldsymbol{\mu} - \boldsymbol{\mu}^*|^2)\right)^2 + O(|\boldsymbol{\mu} - \boldsymbol{\mu}^*|^3) \\ &\qquad = 1 + (\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) - 1) - \frac{1}{2}(\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) - 1)^2 + O(|\boldsymbol{\mu} - \boldsymbol{\mu}^*|^2),\end{aligned}$$

so that

$$\beta_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) = \tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) + O((\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) - 1)^2) + O(|\boldsymbol{\mu} - \boldsymbol{\mu}^*|^2), \qquad \text{as } \boldsymbol{\mu} \to \boldsymbol{\mu}^*.$$

Finally, by exploiting again (2.43), we end up with (2.44). □

In particular, (2.43) guarantees that, for $\boldsymbol{\mu}$ near $\boldsymbol{\mu}^*$, the bilinear form $\pi : V \times V \to \mathbb{R}$

$$\pi(u, v) = (T^{\boldsymbol{\mu}} u, T^{\boldsymbol{\mu}^*} v)_V$$

is coercive. Thus, we could compute a lower bound to $\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ by applying the SCM algorithm proposed in [HRSP07], since this surrogate problem is coercive thanks to (2.42). However, because of the parameter dependence through the solution $y_h(\boldsymbol{\mu})$, $\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ cannot be expressed as the solution of a *linear program*, which is the key ingredient of SCM in order to provide an efficient offline-online decomposition.

Therefore, we propose to approximate $\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ by the following surrogate:

$$\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) \approx \hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) = \inf_{v \in V_h} \frac{dA[y_h(\boldsymbol{\mu}^*)](v, T^{\boldsymbol{\mu}^*} v; \boldsymbol{\mu})}{\|T^{\boldsymbol{\mu}^*} v\|_V^2}. \tag{2.45}$$

In fact, by using the same argument of Proposition 2.2, it is possible to show that[4]

$$|\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) - \hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})| \leq \frac{\gamma_h^c(\boldsymbol{\mu})}{\beta_h(\boldsymbol{\mu}^*)} L_y^h |\boldsymbol{\mu} - \boldsymbol{\mu}^*|, \tag{2.46}$$

whence $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ is a reasonable approximation to $\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ for $\boldsymbol{\mu}$ sufficiently near to $\boldsymbol{\mu}^*$. The quality of this approximation depends on the ratio $\gamma_h^c(\boldsymbol{\mu})/\beta_h(\boldsymbol{\mu}^*)$, which for $\boldsymbol{\mu} = \boldsymbol{\mu}^*$ is nothing but the condition number of the problem.

Moreover, the approximation $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ can be obtained by solving a linear program amenable to a suitable offline-online decomposition. In fact, given $\boldsymbol{\mu}^* \in \mathcal{D}$,

$$\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) = \inf_{\mathbf{z} \in \mathcal{Z}_*} \mathcal{J}(\mathbf{z}; \boldsymbol{\mu}), \qquad \mathcal{J}(\mathbf{z}; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \theta_q^a(\boldsymbol{\mu}) z_q + \sum_{q'=1}^{Q_c} \theta_{q'}^c(\boldsymbol{\mu}) z_{Q_a+q'}, \tag{2.47}$$

with $\mathbf{z} = (z_1, \ldots, z_{Q_a}, z_{Q_a+1}, \ldots, z_{Q_a+Q_c})$. Here $\mathcal{Z}_* \subset \mathbb{R}^{Q_A}$ (with $Q_A = Q_a + Q_c$) is given by

$$\mathcal{Z}_* = \left\{ \mathbf{z} \in \mathbb{R}^{Q_A} : \exists\, w_h^{\mathbf{z}} \in V_h \;\middle|\; y_q = \frac{a_q(w_h^{\mathbf{z}}, T^{\boldsymbol{\mu}^*} w_h^{\mathbf{z}})}{\|T^{\boldsymbol{\mu}^*} w_h^{\mathbf{z}}\|_V^2},\ 1 \leq q \leq Q_a, \right.$$
$$\left. z_{Q_a+q'} = \frac{d_{q'}(y_h(\boldsymbol{\mu}^*))(w_h^{\mathbf{y}}, T^{\boldsymbol{\mu}^*} w_h^{\mathbf{z}})}{\|T^{\boldsymbol{\mu}^*} w_h^{\mathbf{z}}\|_V^2},\ 1 \leq q' \leq Q_c \right\}.$$

We can now use SCM to build a lower bound of $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ through a sequence of suitable relaxed problems of (2.47), by seeking the minimum of $\mathcal{J}$ on a descending sequence of

[4] Here L_y^h denotes the Lipschitz constant of the solution map $\boldsymbol{\mu} \to y_h(\boldsymbol{\mu})$. Thus L_y^h is the discrete counterpart of the Lipschitz constant L_y defined in (2.20).

larger sets, built by adding successively linear constraints. We also build an upper bound to $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$, which will serve to define a suitable *error indicator* in the greedy procedure for the construction of the local lower bound.

We underline that the original SCM would proceed by computing a local lower bound to $\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$, thus providing a global lower bound to $\beta_h(y_h(\boldsymbol{\mu}))$. Our linearized SCM computes instead a local lower bound to an approximation $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ of $\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$, in order to enable the offline-online decomposition of the whole procedure. As a result, we obtain a global approximation – rather than a lower bound – to $\beta_h(y_h(\boldsymbol{\mu}))$. We report the details of the procedure in the following subsections.

2.4.1 Construction of a local lower bound to $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$

We first remark that $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) = (\lambda_{\min}^{\boldsymbol{\mu}^*}(\boldsymbol{\mu}))^{1/2}$, where $\lambda_{\min}^{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ is the smallest eigenvalue $\lambda^{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ such that $(\lambda^{\boldsymbol{\mu}^*}(\boldsymbol{\mu}), \mathbf{v}) \in \mathbb{R}_+ \times V_h$, $\mathbf{v} \neq 0$, satisfy

$$\frac{1}{2}\left[\mathbf{A}(\boldsymbol{\mu}^*)^T\mathbf{X}^{-1}\hat{\mathbf{A}}(\boldsymbol{\mu};\boldsymbol{\mu}^*) + \hat{\mathbf{A}}(\boldsymbol{\mu};\boldsymbol{\mu}^*)^T\ \mathbf{X}^{-1}\mathbf{A}(\boldsymbol{\mu}^*)\right]\mathbf{v} = \lambda^{\boldsymbol{\mu}^*}(\boldsymbol{\mu})\mathbf{A}(\boldsymbol{\mu}^*)^T\mathbf{X}^{-1}\mathbf{A}(\boldsymbol{\mu}^*)\mathbf{v}, \tag{2.48}$$

being $\hat{\mathbf{A}}(\boldsymbol{\mu};\boldsymbol{\mu}^*)$ the matrix resulting from the discretization of $dA[y_h(\boldsymbol{\mu}^*)](\cdot,\cdot;\boldsymbol{\mu})$. By extending the procedure presented in [HRSP07], we report here the main steps required to construct a lower and an upper bound to $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$:

1. *Bounding box construction.* In order to guarantee that (2.47) is well-posed, we can construct a (continuity) bounding box $B_{\boldsymbol{\mu}^*} \subset \mathbb{R}^{Q_A}$ given by [HKC+10]

$$B_{\boldsymbol{\mu}^*} = \prod_{q=1}^{Q_a}\left[-\frac{\gamma_q^a}{\beta_h(y_h(\boldsymbol{\mu}^*))}, \frac{\gamma_q^a}{\beta_h(y_h(\boldsymbol{\mu}^*))}\right] \times \prod_{q'=1}^{Q_c}\left[-\frac{\gamma_{q'}^d(\boldsymbol{\mu}^*)}{\beta_h(y_h(\boldsymbol{\mu}^*))}, \frac{\gamma_{q'}^d(\boldsymbol{\mu}^*)}{\beta_h(y_h(\boldsymbol{\mu}^*))}\right], \tag{2.49}$$

   where $\beta_h(y_h(\boldsymbol{\mu}^*))$ is the solution of (2.33) computed for $\boldsymbol{\mu} = \boldsymbol{\mu}^*$. Alternatively, as recently proposed in [Yan14], we can consider the following bounding box,

$$B_{\boldsymbol{\mu}^*}^Y = \prod_{q=1}^{Q_a}\left[\inf_{v\in V_h}\frac{a_q(v, T^{\boldsymbol{\mu}^*}v)}{\|T^{\boldsymbol{\mu}^*}v\|_V^2}, \sup_{v\in V_h}\frac{a_q(v, T^{\boldsymbol{\mu}^*}v)}{\|T^{\boldsymbol{\mu}^*}v\|_V^2}\right] \times \prod_{q'=1}^{Q_c}\left[\inf_{v\in V_h}\frac{d_{q'}(y_h(\boldsymbol{\mu}^*))(v, T^{\boldsymbol{\mu}^*}v)}{\|T^{\boldsymbol{\mu}^*}v\|_V^2}, \sup_{v\in V_h}\frac{d_{q'}(y_h(\boldsymbol{\mu}^*))(v, T^{\boldsymbol{\mu}^*}v)}{\|T^{\boldsymbol{\mu}^*}v\|_V^2}\right], \tag{2.50}$$

   which is proved to be tighter than (2.49), i.e. $B_{\boldsymbol{\mu}^*}^Y \subset B_{\boldsymbol{\mu}^*}$. Let us remark however that the computation of $B_{\boldsymbol{\mu}^*}^Y$ requires additional operations, in particular: (i) for each $\boldsymbol{\mu}^*$ the bounding box has to be fully recomputed, while for the former we can compute the γ_q^a's once and for all, and only update the γ_q^d's at each iteration; (ii) for each $\boldsymbol{\mu}^*$, $B_{\boldsymbol{\mu}^*}^Y$ requires to compute not only the maximum but also the minimum eigenvalue of the involved bilinear forms. This is a demanding task, which can become unaffordable when Q_a and Q_c become too large. In Section 2.6 we will show a detailed comparison of these two options.

2. *Relaxed LP problem.* Given a SCM sample $\mathcal{C}_{\boldsymbol{\mu}^*} = \{\boldsymbol{\mu}_1^*, \dots, \boldsymbol{\mu}_k^*\}$ associated to $\boldsymbol{\mu}^*$, compute the corresponding lower bounds $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}')$, by solving (2.48) $\forall \boldsymbol{\mu}' \in \mathcal{C}_{\boldsymbol{\mu}^*}$; then, define the relaxation set

$$\mathcal{Z}_*^{\mathrm{LB}}(\mathcal{C}_{\boldsymbol{\mu}^*}) = \left\{ \mathbf{z} \in B_{\boldsymbol{\mu}^*} \;\middle|\; \mathcal{J}(\mathbf{z}; \boldsymbol{\mu}') \geq \hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}'), \quad \forall \boldsymbol{\mu}' \in \mathcal{C}_{\boldsymbol{\mu}^*} \right\}$$

by selecting a set of additional linear constraints associated to $\mathcal{C}_{\boldsymbol{\mu}^*}$. Let us remark that the desired *local lower bound* $\hat{\beta}_{\boldsymbol{\mu}^*}^{\mathrm{LB}}(\boldsymbol{\mu})$ is provided by the solution of the following relaxed problem:

$$\hat{\beta}_{\boldsymbol{\mu}^*}^{\mathrm{LB}}(\boldsymbol{\mu}) \equiv \hat{\beta}_{\boldsymbol{\mu}^*}^{\mathrm{LB}}(\boldsymbol{\mu}; \mathcal{C}_{\boldsymbol{\mu}^*}) = \inf_{\mathbf{z} \in \mathcal{Z}_*^{\mathrm{LB}}(\mathcal{C}_{\boldsymbol{\mu}^*})} \mathcal{J}(\mathbf{z}; \boldsymbol{\mu}), \qquad \forall \boldsymbol{\mu} \in \mathcal{D}_{\boldsymbol{\mu}^*}, \tag{2.51}$$

since $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu}) \geq \hat{\beta}_{\boldsymbol{\mu}^*}^{\mathrm{LB}}(\boldsymbol{\mu})$. In fact, $\mathcal{Z}_* \subset \mathcal{Z}_*^{\mathrm{LB}}(\mathcal{C}_{\boldsymbol{\mu}^*})$ and thus the minimum is taken over a larger set. Note that (2.51) has to be solved $\forall \boldsymbol{\mu} \in \Xi_{\mathrm{train}}$ ($\Xi_{\mathrm{train}} \subset \mathcal{D}$ being a very rich training sample), whereas the definition of $\mathcal{D}_{\boldsymbol{\mu}^*} \subset \mathcal{D}$ will be made precise later on. We can also define an upper bound to $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ as follows:

$$\hat{\beta}_{\boldsymbol{\mu}^*}^{\mathrm{UB}}(\boldsymbol{\mu}) \equiv \hat{\beta}_{\boldsymbol{\mu}^*}^{\mathrm{UB}}(\boldsymbol{\mu}; \mathcal{C}_k^*) = \inf_{\mathbf{z} \in \mathcal{Z}_*^{\mathrm{UB}}(\mathcal{C}_{\boldsymbol{\mu}^*})} \mathcal{J}(\mathbf{z}; \boldsymbol{\mu}), \qquad \forall \boldsymbol{\mu} \in \mathcal{D}_{\boldsymbol{\mu}^*}, \tag{2.52}$$

where

$$\mathcal{Z}_*^{\mathrm{UB}}(\mathcal{C}_{\boldsymbol{\mu}^*}) = \{ \tilde{\mathbf{z}} \in \mathbb{R}^{Q_A} \;:\; \tilde{\mathbf{z}} = \arg\min_{\mathbf{z} \in \mathcal{Z}_*} \mathcal{J}(\mathbf{z}; \boldsymbol{\mu}'), \;\; \forall \boldsymbol{\mu}' \in \mathcal{C}_{\boldsymbol{\mu}^*} \}.$$

Since $\mathcal{Z}_*^{\mathrm{UB}}(\mathcal{C}_{\boldsymbol{\mu}^*}) \subset \mathcal{Y}_*$ – see e.g. [HRSP07] for the proof – (2.52) is in fact an upper bound for $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$.

3. *Selection of the successive constraint.* The set $\mathcal{C}_{\boldsymbol{\mu}^*}$ is built through a (local) greedy procedure. Starting from $\mathcal{C}_{\boldsymbol{\mu}^*} = \{\boldsymbol{\mu}^*\}$, we iteratively enrich the set $\mathcal{C}_{\boldsymbol{\mu}^*}$ by adding the point $\hat{\boldsymbol{\mu}}$ such that

$$\hat{\boldsymbol{\mu}} = \arg \max_{\boldsymbol{\mu} \in E_{\boldsymbol{\mu}^*} \cap \Xi_{\mathrm{train}}} \rho(\boldsymbol{\mu}; \mathcal{C}_{\boldsymbol{\mu}^*}), \qquad \rho(\boldsymbol{\mu}; \mathcal{C}_{\boldsymbol{\mu}^*}) = \frac{\hat{\beta}_{\boldsymbol{\mu}^*}^{\mathrm{UB}}(\boldsymbol{\mu}) - \hat{\beta}_{\boldsymbol{\mu}^*}^{\mathrm{LB}}(\boldsymbol{\mu})}{\hat{\beta}_{\boldsymbol{\mu}^*}^{\mathrm{UB}}(\boldsymbol{\mu})},$$

until the largest ratio satisfies $\rho(\boldsymbol{\mu}; \mathcal{C}_{\boldsymbol{\mu}^*}) \leq \varepsilon_*$, i.e. it stands under a chosen tolerance $\varepsilon_* \in (0, 1)$. Here we restrict the search for the maximum of $\rho(\cdot;\cdot)$ to a suitable neighborhood $E_{\boldsymbol{\mu}^*}$ of $\boldsymbol{\mu}^*$, which shall represent an empirical approximation of the *coercivity region* (see Proposition 2.4) of $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$. The choice of $E_{\boldsymbol{\mu}^*}$ is problem dependent and is usually made a priori, according to physical intuition, or a posteriori once the first iterations of the algorithm have been run. Further details can be found in Sect. 2.6.

Thus, we end up with $K = |\mathcal{C}_{\boldsymbol{\mu}^*}|$ constraints and a local lower bound $\hat{\beta}_{\boldsymbol{\mu}^*}^{\mathrm{LB}}(\boldsymbol{\mu})$.

2.4.2 Computation of a global approximation

In order to turn the local lower bound $\hat{\beta}_{\boldsymbol{\mu}^*}^{LB}(\boldsymbol{\mu})$, computed upon each selected value $\boldsymbol{\mu}^*$, into a global approximation for $\beta_h(\boldsymbol{\mu})$, we consider a greedy procedure like the one addressed in [HRSP07, HKC+10] for the linear case. We remark that the output of the

Input: train sample Ξ_{train}, $J_{\max}$, $K_{\max}$, SCM tolerance ε_*, starting point $\boldsymbol{\mu}^{1*}$.
set $J = 1$, $\mathcal{C}_{\boldsymbol{\mu}^{1*}} = \{\boldsymbol{\mu}^{1*}\}$, $\mathcal{R}_J = \emptyset$
compute $\beta_h(\boldsymbol{\mu}^{1*})$ by (2.33) and the bounding box $B_{\boldsymbol{\mu}^{1*}}$
while $J < J_{\max}$, $\Xi_{\text{train}} \neq \emptyset$ and $\rho(\boldsymbol{\mu}) > \varepsilon_*$
    compute $\hat{\beta}^{\text{LB}}_{\boldsymbol{\mu}^{J*}}(\boldsymbol{\mu})$, $\hat{\beta}^{\text{UB}}_{\boldsymbol{\mu}^{J*}}(\boldsymbol{\mu})$
    construct $\mathcal{R}^*_J = \{\boldsymbol{\mu} \in \Xi_{\text{train}} \,|\, \hat{\beta}^{\text{LB}}_{\boldsymbol{\mu}^{J*}}(\boldsymbol{\mu}) > 0 \text{ and } \rho(\boldsymbol{\mu}; \mathcal{C}_{\boldsymbol{\mu}^{J*}}) \leq \varepsilon_*\}$
    compute $\beta^{\text{A}}_h(\boldsymbol{\mu})$ as in (2.53)
    if $\mathcal{R}^*_J \setminus \mathcal{R}_J = \emptyset$ or $|\mathcal{C}_{\boldsymbol{\mu}^{J*}}| = K_{max}$ **then**
        update $\Xi_{\text{train}} = \Xi_{\text{train}} \setminus \mathcal{R}_J$
        set $J = J + 1$ and select a new $\boldsymbol{\mu}^{J*}$
        compute $\beta_h(\boldsymbol{\mu}^{J*})$ by (2.33) and the bounding box $B_{\boldsymbol{\mu}^{J*}}$
        set $\mathcal{C}_{\boldsymbol{\mu}^{J*}} = \{\boldsymbol{\mu}^{J*}\}$
        construct $\mathcal{R}_J = \{\boldsymbol{\mu} \in \Xi_{\text{train}} \,|\, \hat{\beta}^{\text{LB}}_{\boldsymbol{\mu}^{J*}}(\boldsymbol{\mu}) > 0 \text{ and } \rho(\boldsymbol{\mu}; \mathcal{C}_{\boldsymbol{\mu}^{J*}}) \leq \varepsilon_*\}$
    else
        $\hat{\boldsymbol{\mu}} = \arg\max_{\boldsymbol{\mu} \in E_{\boldsymbol{\mu}^{J*}}} \rho(\boldsymbol{\mu}; \mathcal{C}_{\boldsymbol{\mu}^{J*}})$
        set $\mathcal{C}_{\boldsymbol{\mu}^{J*}} = \mathcal{C}_{\boldsymbol{\mu}^{J*}} \cup \{\hat{\boldsymbol{\mu}}\}$

        compute $\hat{\beta}_{\boldsymbol{\mu}^*}(\hat{\boldsymbol{\mu}})$ by solving (2.48)
        set $\mathcal{R}_J = \mathcal{R}^*_J$
    end if
end while

Algorithm 2.1 Linearized SCM algorithm

coverage procedure are the set $\mathcal{S} = \{\boldsymbol{\mu}^{1*}, \ldots, \boldsymbol{\mu}^{J*}\}$, $J \leq J_{\max}$ and the associated samples $\mathcal{C}_{\boldsymbol{\mu}^{j*}}$, for any $j = 1, \ldots, J$, where $K(j) = |\mathcal{C}_{\boldsymbol{\mu}^{j*}}| < K_{\max}$ is the number of constraints points related to each $\boldsymbol{\mu}^{j*} \in \mathcal{S}$. Thus, a global approximation for $\beta_h(\boldsymbol{\mu})$ is

$$\beta^{\text{A}}_h(\boldsymbol{\mu}) = \beta_h(\boldsymbol{\mu}^{\sigma*}) \hat{\beta}^{\text{LB}}_{\boldsymbol{\mu}^{\sigma*}}(\boldsymbol{\mu}), \quad \text{being} \quad \sigma \equiv \sigma(\boldsymbol{\mu}) = \arg \max_{j \in \{1,\ldots,J\}} \beta_h(\boldsymbol{\mu}^{j*})\, \hat{\beta}^{\text{LB}}_{\boldsymbol{\mu}^{j*}}(\boldsymbol{\mu}), \tag{2.53}$$

so that the subdomains $\mathcal{D}_{\boldsymbol{\mu}^{*j}}$, $j = 1, \ldots, J$, are defined as

$$\mathcal{D}_{\boldsymbol{\mu}^{*j}} = \{\boldsymbol{\mu} \in \mathcal{D} \,:\, \beta_h(\boldsymbol{\mu}^{j*}) \hat{\beta}^{\text{LB}}_{\boldsymbol{\mu}^{j*}}(\boldsymbol{\mu}) \geq \beta_h(\boldsymbol{\mu}^{j'}) \hat{\beta}^{\text{LB}}_{\boldsymbol{\mu}^{j'}}(\boldsymbol{\mu}), \quad \forall\, j' = 1, \ldots, J\}. \tag{2.54}$$

As in the original SCM, the global approximation $\beta^{\text{A}}_h(\boldsymbol{\mu})$ interpolates $\beta_h(\boldsymbol{\mu})$ at each $\boldsymbol{\mu}^* \in \mathcal{S}$, being $\beta^{\text{A}}_h(\boldsymbol{\mu}^*) = \beta_h(\boldsymbol{\mu}^*)$. The set $\mathcal{S} = \{\boldsymbol{\mu}^{1*}, \ldots, \boldsymbol{\mu}^{J*}\}$ is built through a *(global) greedy procedure*, which encapsulates the local ones used for building each sample. The whole procedure is summarized in the Algorithm 2.1.

Let us highlight which are the main computational costs of this problem. We denote by $n_{\text{train}} = |\Xi_{\text{train}}|$ and we define

$$n_\beta = \sum_{j=1}^{J} |\mathcal{C}_{\boldsymbol{\mu}^{j*}}|, \qquad n_{\mathcal{C}} = \max_{j=1,\ldots,J} |\mathcal{C}_{\boldsymbol{\mu}^{j*}}|.$$

In the offline stage we have to: (i) solve J times problem (2.23) in order to compute $y_h(\boldsymbol{\mu})$ and assemble n_β times the Fréchet derivative; (ii) solve $n_{\text{eig}}^{(1)} = n_\beta + Q_a + JQ_c$ (respectively $n_{\text{eig}}^{(2)} = n_\beta + 2JQ_a + 2JQ_c$) eigenproblems when using the bounding box (2.49) (respectively (2.50)), (iii) solve $n_{\text{train}} n_\beta$ linear programs to compute the current global lower bounds (2.53) at each iteration of the algorithm.

In the online stage, each evaluation $\boldsymbol{\mu} \to \beta_h(y_h(\boldsymbol{\mu}))$ only requires to solve J linear programs in $Q_A = Q_a + Q_c$ variables with at most $n_{\mathcal{C}} + 2Q_A$ constraints (independently of the employed bounding box).

Remark 2.2. As in the linear case, the computational complexity of the offline stage of the SCM depends inherently on $Q_A N_h^\alpha$, where the dependence on the dimension N_h is due to eigenvalues calculation (with $\alpha \in [1, 3]$). Thus, already for rather small problems, the size Q_A of the affine expansion may cause the offline stage to become potentially very expensive. A *two-level* affine decomposition strategy was proposed in [LR11, LMR12] to tackle the case of large affine operators (e.g. recovered through the empirical interpolation method). •

2.5 A heuristic strategy based on adaptive interpolation

Our numerical experience indicates a rather slow convergence of the linearized SCM procedure when dealing with many ($P \geq 3$) parameters (see also the numerical results of Sect. 2.6). This prompts us to device alternative strategies when dealing with nonlinear operators depending on many parameters.

A first, very simple approach would be to approximate the ($\boldsymbol{\mu}$-dependent) stability factor $\beta_h(\boldsymbol{\mu})$ by the constant

$$\beta_{\text{LB}} = \min_{\boldsymbol{\mu} \in \mathcal{D}} \ \beta_h(\boldsymbol{\mu}). \tag{2.55}$$

Since $\beta_h(\boldsymbol{\mu})$ might be a non-convex function of $\boldsymbol{\mu}$, finding its global minimum on $\mathcal{D}$ requires (i) to combine a local optimization solver with a suitable globalization strategy [HGT10] and possibly (ii) to provide an explicit expression for the sensitivity of $\beta_h(\boldsymbol{\mu})$ with respect to the parameters. This approach is indeed effective when the stability factor changes mildly with respect to parameters. However, as soon as the dimension of $\mathcal{D}$ increases, finding a global minimum becomes extremely expensive, not to mention that this strategy is over-conservative (thus inappropriate) when $\beta_h(\boldsymbol{\mu})$ varies significantly over $\mathcal{D}$.

For these reasons, we propose a heuristic strategy devised to meet an efficiency requirement (at both the offline and the online stages), which returns reliable and sufficiently tight approximations to parametrized stability factors.

2.5.1 Interpolant of the stability factor

Let us denote by $\Xi_{\text{fine}} \subset \mathcal{D}$ a sample set whose dimension $n_{\text{fine}} = |\Xi_{\text{fine}}|$ is sufficiently large. We (arbitrarily and a priori) select a (possibly *small*) set of interpolation points $\Xi_I = \{\boldsymbol{\mu}^j\}_{j=1}^{n_I} \subset \Xi_{\text{fine}}$ and compute the stability factor $\beta_h(\boldsymbol{\mu})$ for each $\boldsymbol{\mu} \in \Xi_I$. Then, we compute a suitable *interpolant* $\beta_I(\boldsymbol{\mu})$ such that

$$\beta_I(\boldsymbol{\mu}) = \beta_h(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \Xi_I \qquad \text{and} \qquad \beta_I(\boldsymbol{\mu}) > 0 \quad \forall \boldsymbol{\mu} \in \Xi_{\text{fine}}.$$

For any given $\boldsymbol{\mu} \in \mathcal{D}$, the computation of $\beta_h(\boldsymbol{\mu})$ requires to solve the following eigenvalue problem: find $(\lambda(\boldsymbol{\mu}), \mathbf{v}) \in \mathbb{R}_+ \times V_h$, $\mathbf{v} \neq 0$, such that

$$\mathbf{A}(\boldsymbol{\mu})^T \mathbf{X}^{-1} \mathbf{A}(\boldsymbol{\mu}) \mathbf{v} = \lambda(\boldsymbol{\mu}) \mathbf{X} \mathbf{v}, \tag{2.56}$$

where $\mathbf{X}$ is the matrix defined by (2.29), whence $\beta_h(\boldsymbol{\mu}) = \sqrt{\lambda_{\min}(\boldsymbol{\mu})}$.

Depending on the number of parameters and their range of variation, different interpolation methods might be employed. In the two-dimensional case considered in [NRMQ13] we used a simple linear interpolant and an equally spaced grid of interpolation points. When the parameter space has higher dimension, using uniform grids would demand for $\beta_h(\boldsymbol{\mu})$ to be computed in a huge number of interpolation points. Following [Man12], we then replace Lagrange interpolation by *radial basis function* (RBF) interpolation. The latter is especially suited to interpolate scattered data in high-dimensional spaces (for a general introduction to RBF methods see, e.g., [Buh03]). We define the RBF interpolant as

$$\beta_I(\boldsymbol{\mu}) = \omega_0 + \boldsymbol{\omega}^T \boldsymbol{\mu} + \sum_{j=1}^{n_I} \gamma_j \, \phi(|\boldsymbol{\mu} - \boldsymbol{\mu}^j|),$$

where ϕ is a radial basis function[5], while the $1 + p + n_I$ interpolation weights $\{\omega_i\}_{i=0}^{p}$, $\{\gamma_j\}_{j=1}^{n_I}$ are determined by requiring the following conditions to hold:

$$\beta_I(\boldsymbol{\mu}^j) = \beta_h(\boldsymbol{\mu}^j) \qquad j = 1, \ldots, n_I, \tag{2.57a}$$

$$\sum_{j=1}^{n_I} \gamma_j = 0, \qquad \sum_{j=1}^{n_I} \gamma_j \boldsymbol{\mu}_i^j = 0 \qquad i = 1, \ldots, p. \tag{2.57b}$$

Equations (2.57a)-(2.57b) lead to the following symmetric linear system of dimension $1 + P + n_I$

$$\begin{pmatrix} \mathbb{M} & \mathbb{P}^T & \mathbf{1}^T \\ \mathbb{P} & 0 & 0 \\ \mathbf{1} & 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\omega} \\ \omega_0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{\beta} \\ 0 \\ 0 \end{pmatrix} \tag{2.58}$$

where $\mathbf{1} = [1, \ldots, 1] \in \mathbb{R}^{n_I}$, $\boldsymbol{\beta} = [\beta_h(\boldsymbol{\mu}^1), \ldots, \beta_h(\boldsymbol{\mu}^{n_I})] \in \mathbb{R}^{n_I}$ and

$$(\mathbb{M})_{ij} = \phi(|\boldsymbol{\mu}^i - \boldsymbol{\mu}^j|), \quad (\mathbb{P})_{pj} = \boldsymbol{\mu}_p^j, \qquad i, j = 1, \ldots, n_I, \;\; p = 1, \ldots, P.$$

System (2.58) is solved in the offline phase to yield the interpolation weights ω_0, $\boldsymbol{\omega}$, $\boldsymbol{\gamma}$.

In order to avoid negative values of the interpolant $\beta_I(\boldsymbol{\mu})$, we may proceed as follows. We first perform the interpolation on a starting grid Ξ_I of $n_I^{(0)}$ interpolation points and then evaluate the resulting interpolant on the fine grid. (Note that this step of the algorithm can be performed in parallel). Next, we enrich the interpolation grid Ξ_I by adding further interpolation points in those regions where the interpolant is negative. This yields a positive (provided that n_{max} is sufficiently large) interpolant $\beta_I(\boldsymbol{\mu})$. If n_{neg} denotes the number of points selected by the second step of the algorithm, we must in the offline stage (i) solve at most $n_I^{(0)} + n_{neg}$ eigenvalue problems; (ii) build $1 + n_{neg}$ times the RBF interpolant, involving $O(n_I^3 + n_I^2)$ operations (being n_I the dimension

[5] In the numerical results of Section 2.6 we employ thin plate splines RBF, i.e. $\phi(r) = r^2 \log(r)$.

of the adaptively enriched set Ξ_I); (iii) evaluate $n_I^{(0)} + n_{neg}$ times the RBF interpolant, requiring $O(n_I n_{\text{fine}})$ operations.

Alternatively, the positivity of $\beta_I(\boldsymbol{\mu})$ can be guaranteed by interpolating the logarithm of $\beta_h(\boldsymbol{\mu})$ rather than $\beta_h(\boldsymbol{\mu})$ itself. In this case, we construct the RBF interpolant

$$\widetilde{\beta}_I(\boldsymbol{\mu}) = \omega_0 + \boldsymbol{\omega}^T \boldsymbol{\mu} + \sum_{j=1}^{n_I} \gamma_j \, \phi(|\boldsymbol{\mu} - \boldsymbol{\mu}^j|) \tag{2.59}$$

by replacing condition (2.57a) with the following

$$\widetilde{\beta}_I(\boldsymbol{\mu}^j) = \log \beta_h(\boldsymbol{\mu}^j) \qquad j = 1, \dots, n_I. \tag{2.60}$$

Then, we define $\beta_I(\boldsymbol{\mu}) = \exp(\widetilde{\beta}_I(\boldsymbol{\mu}))$.

In both cases, this strategy yields a good approximation of the stability factor, with a remarkably smaller computational effort with respect to that of the linearized SCM algorithm. However, as the dimension (and extent) of $\mathcal{D}$ increases, the efficacy of this procedure highly depends on the full factorial grid Ξ_I adopted at the first step. Indeed, if Ξ_I is too coarse, most of the time is spent in the second step of the algorithm trying to ensure the positivity of the interpolant, and eventually resulting in a poor approximation. On the other hand, if Ξ_I is too fine, the additional computational effort can be often unnecessary, since too many points are added in regions where $\beta_h(\boldsymbol{\mu})$ changes mildly. We overcome these inconveniences by further improving this method in the following section.

2.5.2 RBF interpolant with adaptive sampling

In order to achieve a compromise between (i) adding new points in locations with a highly varying response, (ii) adding points in unsampled regions of the domain and (iii) ensuring the positivity of the interpolant, we propose an adaptive strategy based on a four-component criterion $C(\boldsymbol{\mu})$:

$$C(\boldsymbol{\mu}) = \Big(\|\nabla \beta_I(\boldsymbol{\mu})\| + \varepsilon\Big)\Big(|\Delta \beta_I(\boldsymbol{\mu})| + \varepsilon\Big)\left(\frac{h(\boldsymbol{\mu})}{\max h(\boldsymbol{\mu})}\right)^2 g(\beta_I(\boldsymbol{\mu})), \tag{2.61}$$

similarly to what proposed in [MA10]. Let us describe the role of each factor:

- the first two terms account for local changes of the interpolant[6]; an offset parameter ε ensures that $C(\boldsymbol{\mu}) > 0$ when $\|\nabla \beta_I(\boldsymbol{\mu})\| = 0$ or $|\Delta \beta_I(\boldsymbol{\mu})| = 0$;
- by choosing

$$h(\boldsymbol{\mu}) = \min_{\boldsymbol{\mu}^j \in \Xi_I} \|\boldsymbol{\mu} - \boldsymbol{\mu}^j\|_2, \qquad g(s) = \begin{cases} 1 & s > 0 \\ \alpha e^{-s} & s \leq 0 \end{cases}$$

  the third and the fourth terms promote the selection of space-filling points and penalize negative values of the interpolant, respectively, and $\alpha > 0$ is a tuning parameter to be prescribed.

[6]Note that the derivatives of the interpolant are available analytically and, therefore, both $\nabla \beta_I(\boldsymbol{\mu})$ and $\Delta \beta_I(\boldsymbol{\mu})$ can be computed exactly, rather than approximated numerically.

In case the interpolant is defined as in (2.59), we drop the last contribute in $C(\boldsymbol{\mu})$.

In this adaptive algorithm, the new sample locations are then selected as the ones which maximize C over Ξ_{fine}. This optimization problem is solved by enumeration, i.e. by evaluating C on the fine grid Ξ_{fine} and extracting the maximum, rather than using a global optimization algorithm. Indeed, evaluating C over Ξ_{fine} is a very fast operation (with complexity $O(n_I^2\, n_{fine})$) which can be easily performed in parallel. The loop stops when either a predetermined number of interpolation points have been added, or a desired accuracy is reached. The complete procedure[7] is reported in Algorithm 2.2.

Input: the evaluation grid Ξ_{fine}, a set of $n_I^{(0)}$ starting samples Ξ_I, $n_{\max}$

▷ Build initial coarse interpolation

for $j = 1 : n_I^{(0)}$
  set $\boldsymbol{\mu}^j = \Xi_I(j)$ and assemble $\mathbf{F}(\boldsymbol{\mu}^j)$
  compute $\beta_h(\boldsymbol{\mu}^j)$ by solving the eigenvalue problem (2.56)
end for
build the RBF interpolant $\beta_I(\boldsymbol{\mu})$

▷ Enrich interpolation with adaptive sampling

evaluate $\beta_I(\boldsymbol{\mu})$ on Ξ_{fine}
while $j < n_{\max}$ and $E_j > tol$
  compute criterion $C(\boldsymbol{\mu})$ as defined in (2.61)
  set $\boldsymbol{\mu}^j = \arg\max_{\boldsymbol{\mu}\in\Xi_{fine}} C(\boldsymbol{\mu})$ and assemble $\mathbf{F}(\boldsymbol{\mu}^j)$
  compute $\beta_h(\boldsymbol{\mu}^j)$ by solving the eigenvalue problem (2.56)
  build the RBF interpolant $\beta_{I\cup\boldsymbol{\mu}^j}(\boldsymbol{\mu})$
  evaluate $E_j = \max_{\boldsymbol{\mu}\in\Xi_{fine}} |\beta_I(\boldsymbol{\mu}) - \beta_{I\cup\boldsymbol{\mu}^j}(\boldsymbol{\mu})|/|\beta_I(\boldsymbol{\mu})|$
  update the set $\Xi_I = \Xi_I \cup \{\boldsymbol{\mu}^j\}$
end while

Algorithm 2.2 Adaptive RBF interpolant

Several techniques can be employed to assess the interpolation accuracy and possibly estimate the interpolation error, see [MA10] and references therein for further details. Here we simply require the $L^\infty(\Xi_{fine})$-norm of two consecutive iterates to be under a prescribed tolerance tol. Let us remark that the offline costs are just slightly increased with respect to the interpolation technique of Sect. 2.5.1, since the number of operations required to evaluate $C(\boldsymbol{\mu})$ only depends on n_I and n_{fine}, and is independent of N_h. Finally, we remark that the condition number of the RBF matrix arising from (2.57a)-(2.57b) rapidly increases as the number of interpolation points increase, although our greedy sampling helps in delaying this behavior. However, when a large number of interpolation points is needed, *ad hoc* preconditioning strategies [BCM99] or suitable choices of RBF shape parameters can be put in place [FZ07, DFQ14].

[7] Here, the parameters domain $\mathcal{D}$ is normalized to the unit hypercube $[0,1]^p$ for the sake of interpolation; in this way, the results of the interpolation are not affected by possible different scales and range of variations of the parameters.

2.6 Numerical results: application to a backward facing step channel

In this section we illustrate the properties and the performances of the proposed techniques. As a test case, we consider a fluid flow over a backward facing step channel [BBD04], described by the steady Navier-Stokes equations:

$$\begin{aligned} -\nu\Delta\boldsymbol{v} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + \nabla p &= \boldsymbol{0} && \text{in } \Omega_o(\boldsymbol{\mu}) \\ \operatorname{div}\boldsymbol{v} &= 0 && \text{in } \Omega_o(\boldsymbol{\mu}) \\ \boldsymbol{v} &= \boldsymbol{g} && \text{on } \Gamma_d^o \\ \boldsymbol{v} &= \boldsymbol{0} && \text{on } \Gamma_w^o(\boldsymbol{\mu}) \\ -p\boldsymbol{n} + \nu(\nabla\boldsymbol{v})\boldsymbol{n} &= \boldsymbol{0} && \text{on } \Gamma_n^o(\boldsymbol{\mu}), \end{aligned} \tag{2.62}$$

where $(\boldsymbol{v}, p)$ are the velocity and pressure defined over a parametrized domain $\Omega_o(\boldsymbol{\mu}) = \Omega_{o1} \cup \Omega_{o2}(\boldsymbol{\mu}) \cup \Omega_{o3}(\boldsymbol{\mu})$ (see Fig. 2.1). We denote by $\Gamma_D^o = \Gamma_d^o \cup \Gamma_w^o(\boldsymbol{\mu})$ the Dirichlet portion of $\partial\Omega_o$, while $\Gamma_n^o(\boldsymbol{\mu})$ denotes the outflow boundary. We define the Reynolds number as $\mathrm{Re} = D\boldsymbol{v}_b/\nu$, where ν is the kinematic viscosity, $D = 2h$ being $h = 1$ the height of the channel at the inflow, while $\boldsymbol{v}_b = 2/3 \max \boldsymbol{g} = 1$, being $\boldsymbol{g} = [6y(1-y), 0]^T$ the inflow profile. We consider $p = 3$ parameters: the Reynolds number $\mu_1 = \mathrm{Re}$ (so that $\nu = 2/\mu_1$), the step height μ_2 and the channel length μ_3 (downstream of the step).

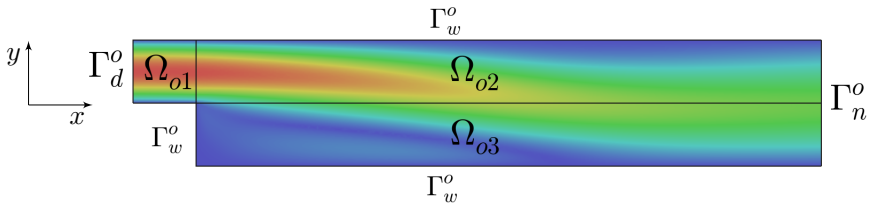


Fig. 2.1 Sketch of the channel geometry with boundaries and partition in affine subdomains. The first subdomain is $\boldsymbol{\mu}$-independent, while $\Omega_{o2} = \Omega_{o2}(\mu_3)$ and $\Omega_{o3} = \Omega_{o3}(\mu_2, \mu_3)$. Coloring is given by the velocity field magnitude obtained for $\mathrm{Re} = 250$.

Problem (2.62) can be rewritten in an affinely parametrized weak form. To do that, we first introduce a decomposition of $\Omega_o(\boldsymbol{\mu})$ into three subdomains (see Fig. 2.1) and a suitable affine geometrical transformation, then we map the problem onto a fixed, reference domain Ω; further details can be found, e.g., in [QR07, Dep08, Man12].

Once the problem has been formulated as in (2.3), we introduce its FE discretization. We use a (inf-sup stable) P_1^b-P_1 approximation for the velocity and pressure variables, i.e. continuous linear FE enriched by bubble functions for the velocity and continuous linear FE for the pressure, see e.g. [QV94]. The total number of degrees of freedom is $\mathcal{N}_h = 40\,064$, obtained using a mesh of 11 485 triangular elements. For the solution of (2.62), we employ a few Picard iterations followed by some Newton iterations, to reach a relative tolerance of 10^{-8} on the norm of the increment. To facilitate the computation of extreme eigenvalues, we consider a weighted norm on V: for any $v = (\boldsymbol{v}, q) \in V$, $\|v\|_V^2 := \tilde{a}(\boldsymbol{v}, \boldsymbol{v}; \hat{\boldsymbol{\mu}}) + \lambda\|\boldsymbol{v}\|_{\boldsymbol{L}^2}^2 + \lambda\|q\|_{L^2}^2$, where $\hat{\boldsymbol{\mu}}$ is a reference parameter value (for instance the centroid of the parameter space), $\tilde{a}(\cdot, \cdot, \boldsymbol{\mu})$ corresponds to the diffusion term

in the momentum equation, while

$$\lambda = \inf_{\boldsymbol{w}\in[H^1(\Omega)]^2} \frac{\tilde{a}(\boldsymbol{w},\boldsymbol{w};\hat{\boldsymbol{\mu}})}{\|\boldsymbol{w}\|^2_{\boldsymbol{L}^2}} > 0.$$

The code we use in this work has been developed in the MATLAB environment; all the linear systems are solved by the sparse direct solver provided by MATLAB, whereas the eigenproblems are solved using MATLAB `eigs` solver. We also take advantage of the existing SCM algorithm already developed (for linear problems) in the `rbMIT` library [HNPR12]. Parallelism is exploited to speed up the matrix assembly in the Navier-Stokes solver as well as to speed up some *embarrassingly parallel* portions of the algorithms we propose. Computations have been performed on a workstation with a Intel Core i5-2400S CPU and 16 GB of RAM. The reported computational times will mainly serve to compare the different strategies.

2.6.1 Backward-facing step channel with a physical parameter

In this first test case we only consider the Reynolds number $\mu_1 \in [20, 250]$ as varying parameter, while the geometrical parameters are frozen to $\mu_2 = 1$ and $\mu_3 = 10$. The affine decomposition (2.12) is recovered for $Q_a = 3$, $Q_c = 1$ and $Q_f = 3$.

First, we numerically verify the inequality (2.39) proved in Proposition 2.3. We build the RB space following the procedure described in [Man14]: we select $N = 11$ basis functions to obtain a maximum error $\|y_h(\boldsymbol{\mu}) - y_N(\boldsymbol{\mu})\|_V$ below 10^{-3} on the whole parameter space. In Fig. 2.2 we report the graphs of the left- and right-hand sides of (2.39) with respect to N (computed on a test sample of 20 parameter values and then averaged).

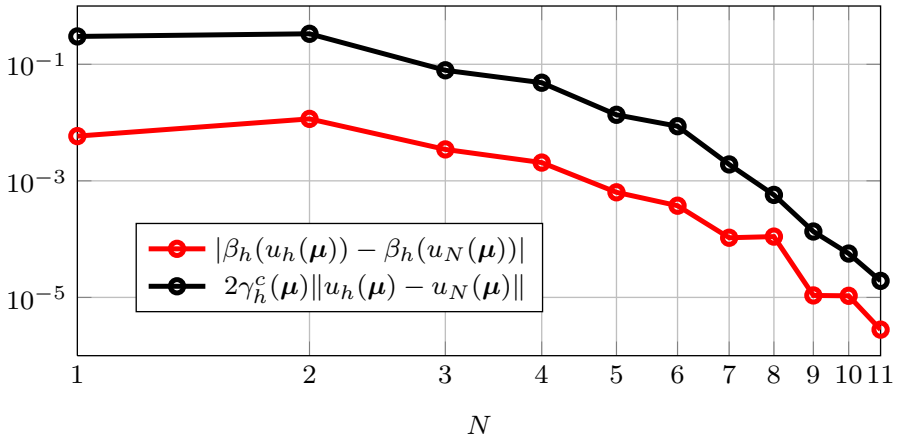


Fig. 2.2 Test 1. Numerical verification of Proposition 2.3.

Then, we numerically verify the coercivity property of the local lower bounds $\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$, i.e. that

$$\tilde{\beta}_{\boldsymbol{\mu}^*} = 1 + O(|\boldsymbol{\mu} - \boldsymbol{\mu}^*|) \qquad \text{as } \boldsymbol{\mu} \to \boldsymbol{\mu}^*$$

as shown in Proposition 2.4. For the sake of verification, we select "by hand" $J = 9$ parameter points $\boldsymbol{\mu}^{j*}$, $1 \le j \le J$, and compute the corresponding $\tilde{\beta}_{\boldsymbol{\mu}^{j*}}(\boldsymbol{\mu})$, which are reported in Fig. 2.3. As expected, for each $\boldsymbol{\mu}^{j*}$, $\tilde{\beta}_{\boldsymbol{\mu}^{j*}}(\boldsymbol{\mu})$ decreases linearly from 1. In

Fig. 2.3 we also report the approximation $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ to $\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$; as expected from estimate (2.45), the quality of the approximation deteriorates as the Reynolds number – and thus the condition number of the problem – increases. Note that for $\boldsymbol{\mu}$ sufficiently far from $\boldsymbol{\mu}^*$, both $\tilde{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ and its approximation $\hat{\beta}_{\boldsymbol{\mu}^*}(\boldsymbol{\mu})$ become negative, and are therefore useless in order to build a positive global approximation to $\beta_h(\boldsymbol{\mu})$. For this reason, we restrict the search for successive constraints $\mathcal{C}_{\boldsymbol{\mu}^*}$ to the interval $E_{\boldsymbol{\mu}^*} = [\boldsymbol{\mu}^* - 20, \boldsymbol{\mu}^* + 20]$.

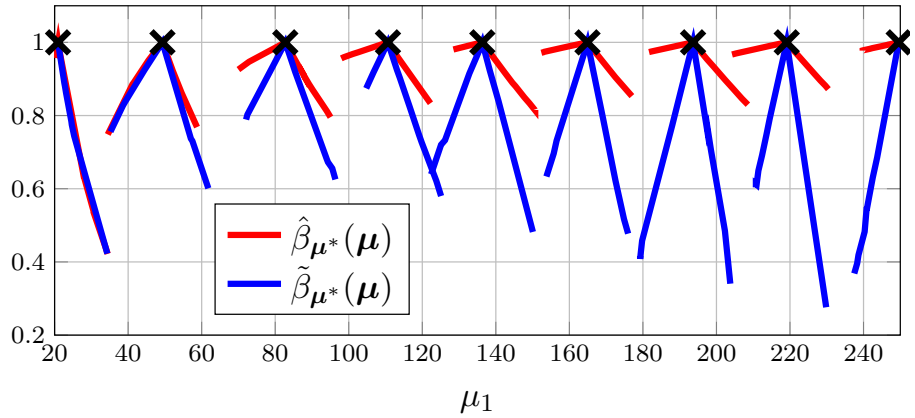


Fig. 2.3 Test 1. $\tilde{\beta}_{\boldsymbol{\mu}^{j*}}(\boldsymbol{\mu})$ and its approximation $\hat{\beta}_{\boldsymbol{\mu}^{j*}}(\boldsymbol{\mu})$, computed in the proximity of $J = 9$ (artificially imposed) parameter points $\boldsymbol{\mu}^{j*}$, $1 \le j \le J$.

Let us now apply the linearized SCM algorithm with a tolerance $\varepsilon_* = 0.7$, $n_{\text{train}} = 1000$ and using the original bounding box (2.49). The greedy procedure selects $J = 6$ anchor points $\boldsymbol{\mu}^*$ and $|\mathcal{C}_{\boldsymbol{\mu}^{j*}}|_{j=1}^6 = [3, 3, 6, 2, 2, 1]$ constraints, thus requiring to solve $n_{\text{eig}}^1 = 26$ eigenproblems. In Fig. 2.4 we report the resulting global approximation $\beta_h^{\text{A}}(\boldsymbol{\mu})$ and the subdomains partition $\mathcal{D}_{\boldsymbol{\mu}^{*j}}$ induced by the algorithm.

Then, in the same setting, we apply the linearized SCM using the tighter bounding box (2.49): we obtain $J = 4$, $|\mathcal{C}_{\boldsymbol{\mu}^{j*}}|_{j=1}^4 = [3, 3, 2, 1]$ and $n_{\text{eig}}^{(2)} = 41$. Regarding the computational performances, the two options require roughly the same time (about 20 minutes) to be performed. The (tighter) bounding box (2.50) lead to a sharper approximation (see Fig. 2.4), yet selecting a smaller number of anchor points $\boldsymbol{\mu}^*$. However, it globally requires to solve a higher number of eigenvalue problems. In particular, its computational complexity depends on the number of terms Q_a and Q_c in the affine decomposition.

Table 2.1 Test 1. Comparison of the computational cost of the linearized SCM algorithm when using the bounding box (2.49) and (2.50).

	Bounding box (2.49)	Bounding box (2.50)
J (number of selected $\boldsymbol{\mu}^*$)	6	4
$\lvert\mathcal{C}_{\boldsymbol{\mu}^{j*}}\rvert$ $j = 1, \dots, J$	[3; 3; 6; 2; 2; 1]	[3; 3; 2; 1]
Number of eigenproblems	26	41
SCM tolerance ε_*	0.7	0.7
Total time (s)	1088	1191

Let us now move to the other heuristic strategies. We first compute the *minimum*

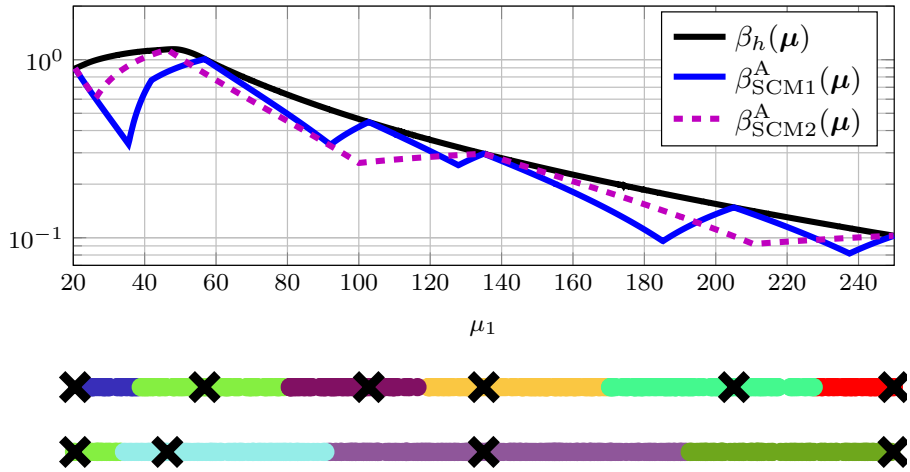


Fig. 2.4 Test 1. Comparison between the approximation of the stability factor obtained using the linearized SCM algorithm with different bounding box: $\beta^{\mathrm{A}}_{SCM1}$ refers to (2.49), while $\beta^{\mathrm{A}}_{SCM2}$ is obtained using (2.50). We also report the subdomains (2.54) induced by the algorithm, with different colors (top: $\beta^{\mathrm{A}}_{SCM1}$, bottom: $\beta^{\mathrm{A}}_{SCM2}$). The corresponding $\boldsymbol{\mu}^{j*}$, $1 \leq j \leq J$ are represented by black crosses.

stability factor by performing a multi-start optimization with three different initial points: we find a minimum stability factor $\beta_{\mathrm{LB}} = 0.1025$ (attained for $\mu_1 = 250$), which turns out to be the global minimum of $\beta_h(\boldsymbol{\mu})$ (see Fig. 2.5). In this case, the algorithm requires to solve 45 eigenproblems. We stress the importance of the multi-start strategy; indeed, if we start the optimization from $\mu_1 < 40$, the algorithm converges to the local minimum attained for $\mu_1 = 20$, thus largely overestimating the global one.

Then, we compute the adaptive RBF interpolant $\beta_I(\boldsymbol{\mu})$: starting from an initial coarse grid of 4 (uniformly distributed) interpolation points, the adaptive procedure selects 10 additional interpolation points so that $E_j < 10^{-3}$ (see Fig. 2.6). The effectiveness of the adaptivity criterion is demonstrated by the evidence that most of the interpolation points are added in the region with the highest variation of $\beta_h(\boldsymbol{\mu})$, as it can be seen in Fig. 2.5. In this case the construction of the interpolant only requires to solve 14 eigenproblems, taking about 8 minutes. Let us remark that, while the final interpolant is almost *coincident* with the exact stability factor, already the initial one (computed on the coarse grid) can be considered as a satisfactory approximation for our purposes.

In Fig. 2.6 we also report a convergence analysis of the RBF interpolation comparing the adaptive versus the uniform refinement of the interpolation grid; in particular, we show the convergence of the $L^\infty(\Xi_{test})$ relative error between the stability factor and its interpolant, where $\Xi_{test} \subset \mathcal{D}$ is a uniform grid of 1 000 points. We remark that the adaptive strategy allows to achieve the same accuracy with a considerably smaller number of interpolation points.

2.6.2 Backward-facing step channel with both physical and geometrical parameters

In the second test case, we consider as parameters both the Reynolds number μ_1 and the height of the channel step μ_2; the parameter space is now given by $\mathcal{D} = [20, 200] \times [0.5, 1.5]$.

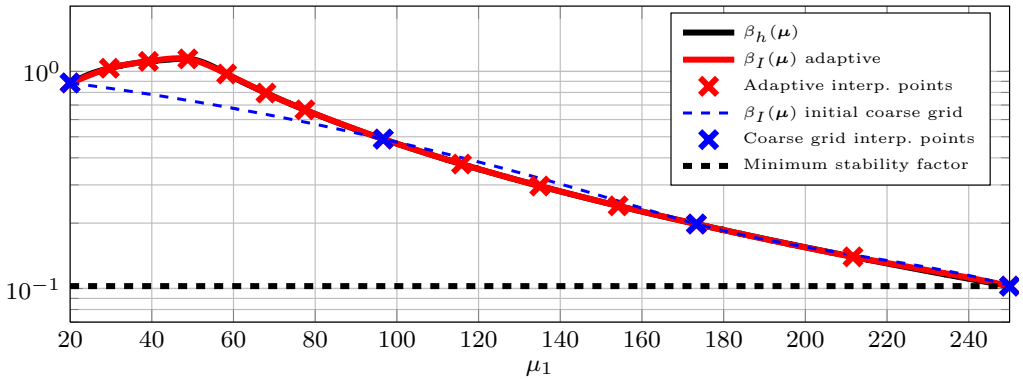


Fig. 2.5 Test 1. Comparison of the heuristic strategies. The value of the *minimum stability factor* and the RBF interpolant with respect to the true stability factor $\beta_h(\boldsymbol{\mu})$ (black line) are reported. For the latter, both the initial interpolation on a coarse grid of 4 points (blue dashed line) and the result of the adaptive strategy (red line) are shown.

Table 2.2 Test 1. Comparison of the computational costs. Computations have been performed using 4 cores on a desktop computer.

	# eigenproblems	Time (s)
Linearized SCM with (2.49)	26	1088
Linearized SCM with (2.50)	41	1191
Minimum	45	1320
Adaptive RBF interpolant	14	470

We have slightly restricted the range of the parameter μ_1 to avoid numerical instabilities (due to the poor convergence of the nonlinear solver) occurring for high values of μ_1 and μ_2. The affine decomposition (2.12) now holds with $Q_a = 5$ ad $Q_c = 2$.

We first run the linearized SCM algorithm with a tolerance $\varepsilon_* = 0.85$, $n_{\text{train}} = 10^4$ and using the original bounding box (2.49). The algorithm shows an extremely low convergence: $J = 195$ parameter values $\boldsymbol{\mu}^{1*}, \dots, \boldsymbol{\mu}^{J*}$ and about 750 constraints are selected, requiring to solve $O(10^3)$ eigenproblems. Its poor convergence rate is mainly due to the geometrical variation induced by the parameter μ_2. Indeed, by running the SCM with the first parameter frozen to $\mu_1 = 100$, the computation of $\beta_h^{\text{A}}(\boldsymbol{\mu})$ (shown in Fig. 2.8) required $J = 17$ parameter values $\boldsymbol{\mu}^{j*}$, a rather large number compared to the results of the previous section.

Then, we build the adaptive RBF interpolant, starting from a coarse grid of uniformly distributed 4×3 interpolation points. The algorithm stops after selecting 23 further samples, corresponding to a maximum budget of 35 interpolation points and $E_j \approx 2 \cdot 10^{-2}$. Note that this latter underestimates the interpolation error of one order of magnitude; in fact, as shown in Fig. 2.9 we are approximating $\beta_h(\boldsymbol{\mu})$ with a relative $L^\infty(\Xi_{\text{fine}})$ error of about 10^{-1}. Nevertheless, the qualitative behavior of the stability factor is well captured. Moreover, the construction of the interpolant $\beta_I(\boldsymbol{\mu})$ takes less than 20 minutes, while the linearized SCM algorithm requires many hours to build $\beta_h^{\text{A}}(\boldsymbol{\mu})$ (in both cases computations have been performed using 12 cores).

In Fig. 2.10 we also report a convergence analysis of the RBF interpolation comparing

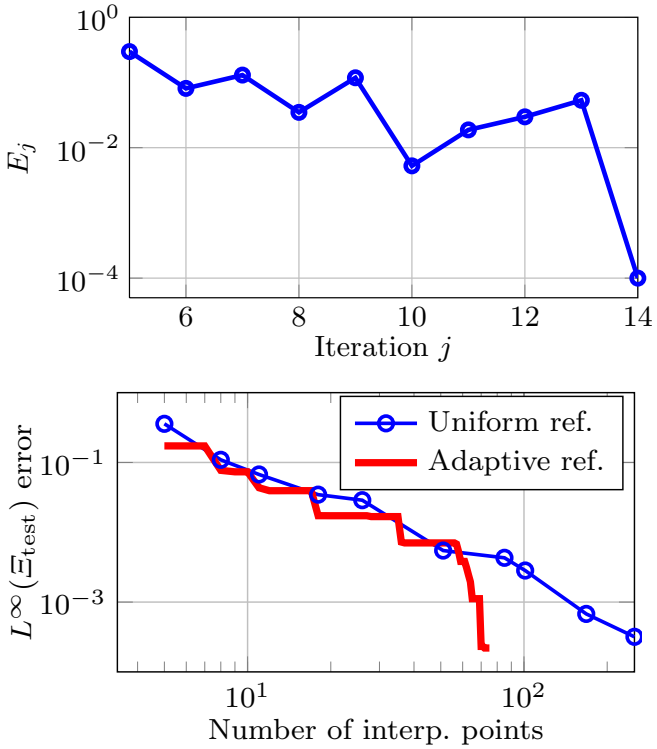


Fig. 2.6 Test 1. Top: indicator E_j (used in the adaptive algorithm to monitor the accuracy of the interpolation) versus the number of iterations j. Bottom: convergence of the $L^\infty(\Xi_{\text{test}})$ relative error between $\beta_h(\boldsymbol{\mu})$ and $\beta_I(\boldsymbol{\mu})$ with respect to the number of interpolation points (in the case of adaptive refinement, we stop the algorithm when a maximum budget of $n_{max} = 75$ points has been reached).

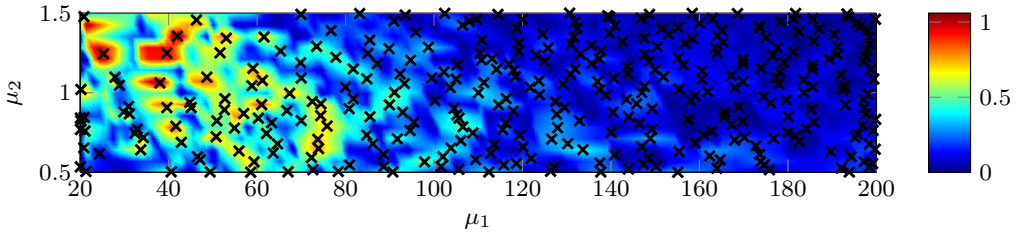


Fig. 2.7 Test 2. Approximation of the stability factor $\beta_h^{\text{A}}(\boldsymbol{\mu})$ as function of (μ_1, μ_2); black cross correspond to the parameter values $\boldsymbol{\mu}^{j*}$, $1 \leq j \leq J = 195$, selected by the linearized SCM algorithm.

the adaptive versus the uniform refinement of the interpolation grid; in this case $\Xi_{\text{test}} \subset \mathcal{D}$ is a factorial grid of 129×65 points.

2.6.3 Backward-facing step channel with three parameters

In the third test case we let all the three parameters vary; in particular the parameter domain is now given by $\mathcal{D} = [20, 200] \times [0.5, 1.5] \times [9, 12]$, while $Q_a = 9$ and $Q_c = 4$.

If follows from our previous discussion that both the linearized SCM and the minimum stability factor strategies are no longer viable in this case. The adaptive RBF interpolant

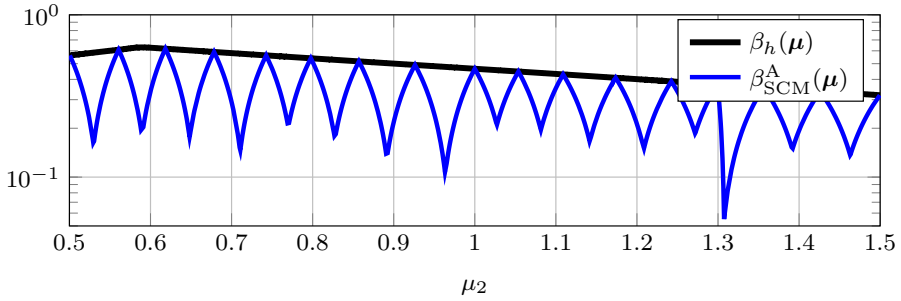


Fig. 2.8 Test 2. Approximation of the stability factor $\beta_h^{\mathrm{A}}(\boldsymbol{\mu})$ as function of μ_2, obtained running the linearized SCM algorithm with $\mu_1 = 100$ fixed. Despite the low variation of $\beta_h(\boldsymbol{\mu})$, SCM requires many iterations to converge (indeed $J = 17$).

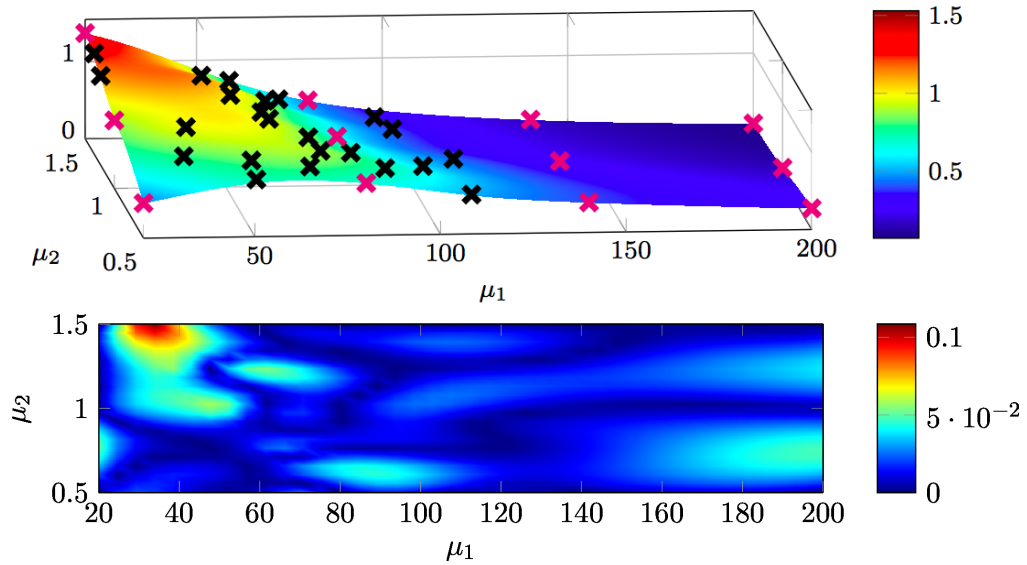


Fig. 2.9 Test 2. Top: RBF interpolant $\beta_I(\boldsymbol{\mu})$, initial coarse grid (magenta) and interpolation points (black) selected by the adaptive procedure. Bottom: relative error between the stability factor $\beta_h(\boldsymbol{\mu})$ and its RBF interpolant.

represents the only chance to obtain, with a reasonable (and somehow predictable) computational effort, a satisfactory approximation of the stability factor.

We start with a coarse grid of $3 \times 3 \times 3$ uniformly distributed interpolation points, and then we let the adaptive procedure select 18 additional samples. We report in Fig. 2.11 the resulting approximation of the stability factor; once again, the adaptive criterion promotes the selection of interpolation points in those regions featuring the highest variations of $\beta_h(\boldsymbol{\mu})$. In Fig. 2.12 we compare the stability factor and the interpolant $\beta_I(\boldsymbol{\mu})$ in the setting of the first test case, i.e. as functions of μ_1, with $\mu_2 = 1$ and $\mu_3 = 10$ fixed; in the same figure we also report the interpolant obtained using 20 more points adaptively selected.

Once again, the adaptive procedure correctly selects the interpolation points in the regions of highest variation of β_h, so that a tight approximation can be easily obtained with a moderate computational effort. As a matter of fact, we experienced that the linearized SCM algorithm tends to select control points $\boldsymbol{\mu}^*$ from subregions of $\mathcal{D}$ where the PDE solution (as well as the corresponding eigenpair) – rather than the stability

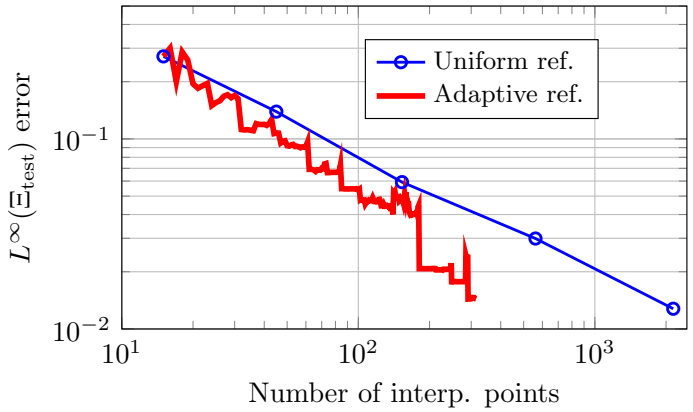


Fig. 2.10 Test 2. Convergence of the $L^\infty(\Xi_{\text{test}})$ relative error between $\beta_h(\boldsymbol{\mu})$ and $\beta_I(\boldsymbol{\mu})$ with respect to the number of interpolation points.

factor – is more sensitive to changes in the parameters. On the other hand, the adaptive interpolation is only affected by the parametric response of the stability factor, enhancing the computational efficiency of this latter.

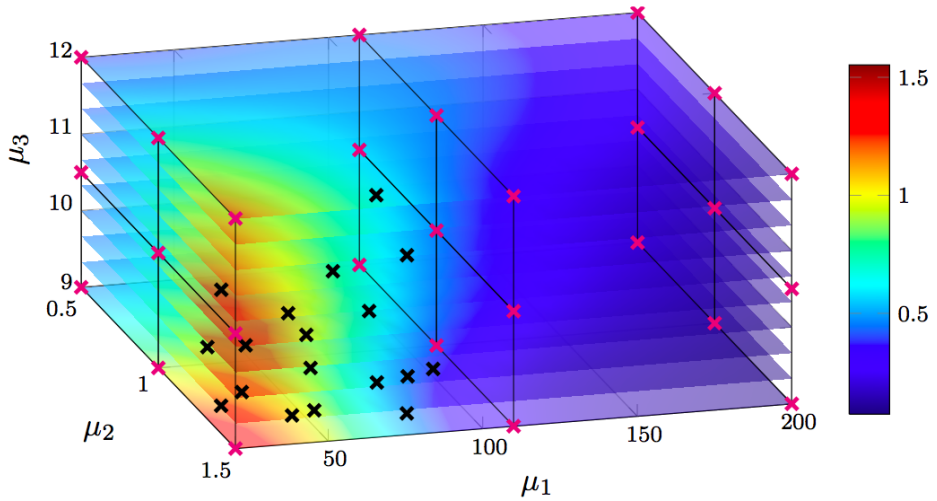


Fig. 2.11 Test 3. Slices of the adaptive RBF interpolant $\beta_I(\boldsymbol{\mu})$ for different values of μ_3; we report the initial full factorial grid (magenta) of $3 \times 3 \times 3$ points and the 18 interpolation points (black) selected by the adaptive procedure.

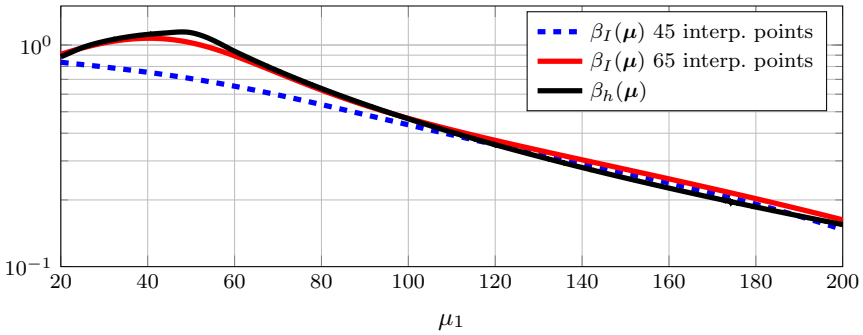


Fig. 2.12 Test 3. RBF interpolant $\beta_I(\boldsymbol{\mu})$ as a function of μ_1, with $\mu_2 = 1$ and $\mu_3 = 10$ fixed (as in the setting of the first numerical test). We compare the adaptive RBF interpolants obtained using 45 (dashed blue line) and 65 (red line) interpolations points (in the whole parameter space) w.r.t the true stability factor $\beta_h(\boldsymbol{\mu})$. Note that none of the interpolation points lies along the $(\mu_1, 1, 10)$ line.

2.7 Approaching a singular point: the channel expansion case

In order to show the robustness of the adaptive interpolation strategy, we tackle a limit-case problem where multiple steady state solutions coexist as the result of a symmetry-breaking pitchfork bifurcation. In particular, we consider a two-dimensional laminar flow through a channel featuring a sudden expansion, whose geometry and boundaries are reported in Fig. 2.13. We define the Reynolds number as $\mathrm{Re} = Uh_1/\nu$, where h_1 is the inlet height, while the characteristic velocity is $U = 2/3 \max|\boldsymbol{g}|$, being $\boldsymbol{g} = [4h_1^{-2}(h_1/2 - y)(h_1/2 + y),\, 0]^T$ the inflow profile. We consider as parameter the Reynolds number $\mu_1 = \mathrm{Re}$, so that $\nu = 2h_1/(3\mu_1)$.

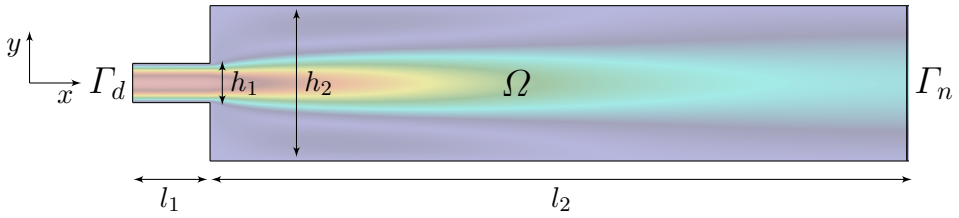


Fig. 2.13 Sketch of the expanding channel geometry: here we fix $l_1 = 1$, $l_2 = 30$, $h_1 = 0.5$, $h_2 = 2$ (yielding an expansion ratio $h_2/h_1 = 4$). Coloring is given by the velocity field magnitude obtained for $\mathrm{Re} = 60$.

At very low Reynolds numbers the flow remains symmetric with separation regions of equal length on both channel walls. Increasing the Reynolds number the separation length increases too, and at a critical value $\mu_1 = \mu_1^*$ one recirculation region grows while the other shrinks. This symmetry breaking occurs as the result of a pitchfork bifurcation in the solution of the Navier-Stokes equations [CST00, Dri97], i.e., for $\mu_1 > \mu_1^*$ two stable (asymmetric) solutions and one unstable (symmetric) solution coexist (see Fig. 2.14). In correspondence of $\mu_1 = \mu_1^*$ the problem becomes ill-posed, the tangent matrix is singular and therefore the stability factor vanishes. From the computational standpoint, since the behavior of the unstable flow may be sensitive to the discretization error, we have employed a fine mesh made of 39 222 triangular elements, obtained as the outcome of a

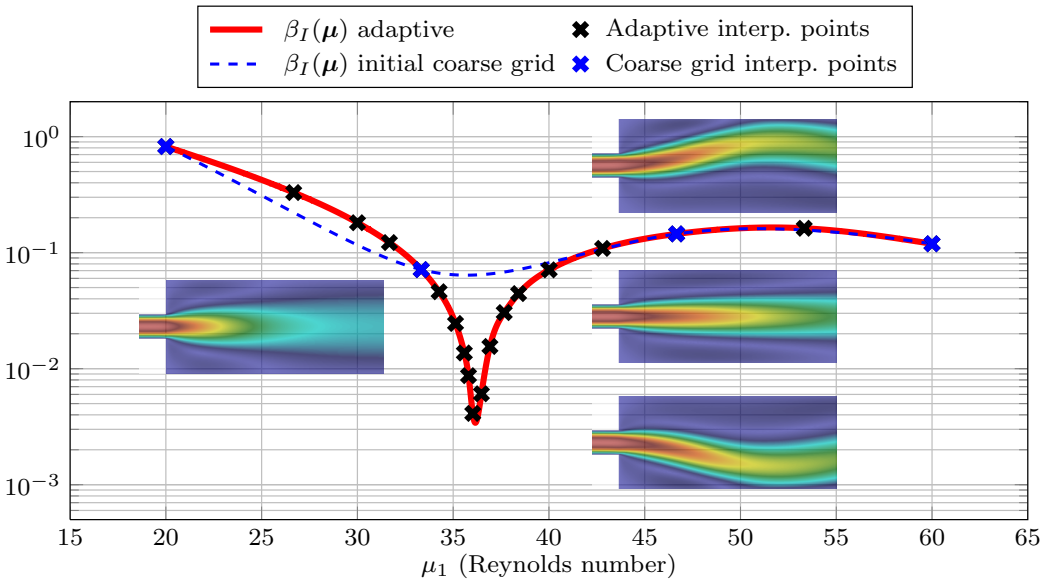


Fig. 2.14 Channel expansion case: initial interpolation on a coarse grid of 4 points (blue dashed line) and results obtained with the adaptive RBF strategy (red line).

suitable mesh convergence study.

We initialize the interpolation procedure on a coarse grid of 4 uniformly distributed interpolation points, yielding a very rough approximation of the true stability factor (that is, unaware of the bifurcation, see Fig. 2.14). Then, our adaptive algorithm selects 16 additional interpolation points in order to reach a maximum of $n_{\max} = 20$ points. Among them, the one corresponding to $\mu_1^* \approx 36.25$ is discarded since the solver does not reach convergence (thus indicating proximity to the bifurcation point). Moreover, since we use as initial guess for the Navier-Stokes solver the solution of the corresponding Stokes equations, we obtain convergence to the symmetric solution for the entire range of Reynolds numbers considered. Thus, for $\mu_1 > \mu_1^*$, the computed stability factor is the one corresponding to the unstable branch of solutions.

Once again, we highlight the efficacy of the adaptive criterion in selecting the interpolation points in the most varying region of the parameter space - that is, in the proximity of the bifurcation point. Thus, we obtain a reliable approximation of the stability factor even in this limit-case scenario, still entailing a moderate computational effort. Therefore, this heuristic strategy proves also to be viable when aiming at detecting bifurcations points. As such, it could be suitably exploited for the construction of a reduced basis approximation to problems where multiple solutions coexist [HMP13, PHV15].

We also remark that this interpolation strategy is completely independent of the high-fidelity problem formulation, as it directly seeks an approximation of the map $\boldsymbol{\mu} \to \beta_h(\boldsymbol{\mu})$. Therefore, it is suitable for both linear and nonlinear, affine and nonaffine problems. In fact, we shall apply this technique to a nonaffinely parametrized Helmholtz problem in the next chapter, as well as to several PDE-constrained optimization problems in Chap 6.

3 Hyper-reduction of parametrized systems by matrix discrete empirical interpolation

In this chapter, we apply a Matrix version of the Discrete Empirical Interpolation Method (MDEIM) for the efficient reduction of nonaffine and nonlinear parameterized systems. Specifically, reduced-order models for nonaffinely parametrized linear elliptic and parabolic PDEs, as well as for the time-dependent Navier-Stokes equations, are proposed. Their efficacy is demonstrated on the solution of two computationally-intensive classes of problems occurring in engineering contexts, namely PDE-constrained shape optimization and parametrized coupled problems.

3.1 A review of existing approaches

Projection-based model reduction techniques lower the computational costs associated with the solution of parameter-dependent high-fidelity models by replacing the solution space with a subspace of much smaller dimension. Whenever interested to solve parametrized PDEs many times, for several parametric instances, a suitable offline-online stratagem becomes mandatory to gain a strong computational speedup. Indeed, expensive computations should be carried out in the offline phase, thus leading to a much cheaper online phase. In this context however, the (possibly) complex parametric dependence and/or nonlinearity of the discretized PDE operators have a major impact on the computational efficiency. We aim at developing an offline-online procedure that alleviates the computational burden associated with such complex (and in particular nonaffine) parametric dependencies. To this end, an affine approximation of the differential operators is developed in the offline phase, leading to inexpensive evaluation of the approximated operators online. These latter are then computed in a general, black-box, purely algebraic way.

For the sake of illustration, let us consider problem (1.11) introduced in Sect. 1.2.2. In this case, the ROM (1.12) reads

$$\mathbf{M}_N(t;\boldsymbol{\mu})\frac{d\mathbf{y}_N}{dt} + \mathbf{A}_N(t;\boldsymbol{\mu})\mathbf{y}_N = \mathbf{g}_N(t;\boldsymbol{\mu}), \tag{3.1}$$

where the reduced matrices and vector are given by

$$\mathbf{M}_N(t;\boldsymbol{\mu}) = \mathbf{W}^T\mathbf{M}(t;\boldsymbol{\mu})\mathbf{V}, \qquad \mathbf{A}_N(t;\boldsymbol{\mu}) = \mathbf{W}^T\mathbf{A}(t;\boldsymbol{\mu})\mathbf{V}, \qquad \mathbf{g}_N(t;\boldsymbol{\mu}) = \mathbf{W}^T\mathbf{g}(t;\boldsymbol{\mu}).$$

The efficient evaluation of the reduced matrices $\mathbf{A}_N(t;\boldsymbol{\mu})$, $\mathbf{M}_N(t;\boldsymbol{\mu})$ and vector $\mathbf{g}_N(t;\boldsymbol{\mu})$ is one of the main challenges in order to achieve efficient offline construction and online resolution of the ROM (3.1). In the particular case that the system matrices (resp. vectors) can be expressed as an affine combination of constant matrices (resp. vectors) weighted by suitable parameter dependent coefficients, each term of the weighted sum can be projected offline onto the RB space. For instance, let us assume that the matrix $\mathbf{A}(t;\boldsymbol{\mu})$ admits an affine decomposition

$$\mathbf{A}(t;\boldsymbol{\mu}) = \sum_{q=1}^{M} \theta_q(t;\boldsymbol{\mu})\mathbf{A}_q, \tag{3.2}$$

where $\theta_q\colon [0,T]\times\mathcal{D} \mapsto \mathbb{R}$ and $\mathbf{A}_q \in \mathbb{R}^{N_h\times N_h}$ are given functions and matrices, respectively, for $q = 1,\ldots,M$. Then, we can express $\mathbf{A}_N(t;\boldsymbol{\mu})$ as

$$\mathbf{A}_N(t;\boldsymbol{\mu}) = \mathbf{W}^T\mathbf{A}(t;\boldsymbol{\mu})\mathbf{V} = \sum_{q=1}^{M} \theta_q(t;\boldsymbol{\mu})\,\mathbf{W}^T\mathbf{A}_q\mathbf{V}.$$

Since the reduced matrices $\mathbf{W}^T\mathbf{A}_q\mathbf{V} \in \mathbb{R}^{N\times N}$ can be precomputed and stored offline, the online construction of the ROM for a given $(t;\boldsymbol{\mu})$ is fast and efficient as long as $M \ll N_h$.

On the other hand, if $\mathbf{A}(t;\boldsymbol{\mu})$ is not affine, this computational strategy breaks down. Thus, the construction of the ROM for a given $(t,\boldsymbol{\mu})$ requires first to assemble the full-order matrices and vectors and then to project them onto the reduced space, thus entailing a computational complexity which scales with the dimension of the large scale system. In order to recover the affine structure (3.2) in those cases where the system is nonaffine (or when (3.2) is not readily available), we must introduce a further level of reduction, called hyper-reduction or system approximation [CFCA13], employing suitable techniques which are briefly reviewed below.

A first class of approaches aims at approximating directly the parametrized reduced operator $\mathbf{A}_N(t;\boldsymbol{\mu})$. Among them, we mention those based on interpolation on appropriate matrix manifolds [ACCF09, DVW10, AF11]. Because they do not require accessing the underlying full operator $\mathbf{A}(t;\boldsymbol{\mu})$ online, these approaches are amenable to a fully online phase that completely bypasses the high-fidelity code (see, e.g., [ADGF13] for an example of application in aeronautics). However, approximating the full operator instead of its reduced counterpart is preferable when residual-based error estimates are required for the offline construction of the ROM and/or its online certification. This is the case for instance when a greedy procedure (see, e.g., [RHP08]) is employed to construct a parametrically robust global reduced basis. As such, we will only consider computational strategies based on the approximation of the full operator $\mathbf{A}(t;\boldsymbol{\mu})$ here.

In this second class of approaches, the approximation of the full operator $\mathbf{A}(t;\boldsymbol{\mu})$ takes place prior to its reduction by Petrov-Galerkin projection. This is the case in the Empirical Interpolation Method (EIM) [BMNP04, GMNP07] (as well as in the 'best point' interpolation method [NPP08]) which was applied in [MQR12b] to shape parameterization and in [DHO12] to operator interpolation. Its discrete variant, the

Discrete Empirical Interpolation Method (DEIM) [CS10] was originally developed to efficiently deal with nonlinear problems, but was then also applied to the approximation of nonaffinely parametrized linear operators, see e.g. [AHS14].

However, in all these cases an extensive, problem-specific pre-processing phase has to be performed in order to cast the parametrized operator $\mathbf{A}(t;\boldsymbol{\mu})$ in a form suitable for the application of EIM or DEIM, see [MQR12b, AHS14]. A closely related technique is the so-called Gappy POD [ES95], which was applied in [CBMF11, CFCA13] to efficiently approximate the action of $\mathbf{A}(t;\boldsymbol{\mu})$ onto the reduced basis $\mathbf{V}$. Although this approach turns out to be less intrusive than the previous ones, it has the drawback that it can be applied only simultaneously (and not prior) to the reduced space construction.

Here, we rely instead on the recently proposed [CTB12, WSH14, BGW15, CTB15] 'Matrix version' of DEIM (MDEIM). Thanks to MDEIM, we show how to deal with complex physical and geometrical parametrizations, as well as with operator nonlinearities, in a black-box, efficient and purely algebraic way.

3.2 Model problems

In this section we introduce two relevant examples of problems which can be cast in the form (1.2) and (1.10)-(1.11): the propagation of a pressure wave into an acoustic horn and the heat transfer for a flow past a circular cylinder.

3.2.1 Helmholtz equation in a parametrized domain

Let us consider the propagation of a pressure wave $P(\mathbf{x},t)$ into the acoustic horn illustrated in Fig. 3.1. Under the assumption of time harmonic waves, the acoustic pressure P can be separated as $P(\mathbf{x},t) = \Re(p(\mathbf{x})e^{i\omega t})$, where the complex amplitude $p(\mathbf{x})$ satisfies the Helmholtz equation (see e.g. [BNB03, KWB12])

$$\begin{cases} \Delta p + \kappa^2 p = 0 & \text{in } \Omega \\ \left(i\kappa + \dfrac{1}{2R}\right) p + \nabla p \cdot \mathbf{n} = 0 & \text{on } \Gamma_o \\ i\kappa p + \nabla p \cdot \mathbf{n} = 2i\kappa A & \text{on } \Gamma_i \\ \nabla p \cdot \mathbf{n} = 0 & \text{on } \Gamma_h \cup \Gamma_s = \Gamma_n. \end{cases} \tag{3.3}$$

Here, $\kappa = \omega/c$ is the wave number, $\omega = 2\pi f$ the angular frequency and $c = 340$ cm s^{-1} the speed of sound. On the boundary Γ_i we prescribe a radiation condition which imposes an inner-going wave with amplitude $A = 1$ and absorbs the outer-going planar waves. A Neumann boundary condition is imposed on the walls Γ_h of the device as well as on the symmetry boundary Γ_s, while an absorbing condition is imposed on the far-field boundary Γ_o (with $R = 1$).

We consider as a first parameter the frequency f. Moreover, in order to describe different geometrical configurations of the horn, we introduce a suitable shape parametrization based on radial basis functions (RBF) (see e.g. [Buh03, MQR12a]). In particular, we define a set of admissible shapes as the diffeomorphic images $\Omega(\boldsymbol{\mu}_g)$ of the reference domain Ω through a parametrized map $T(\cdot;\boldsymbol{\mu}_g)$ depending on four parameters representing

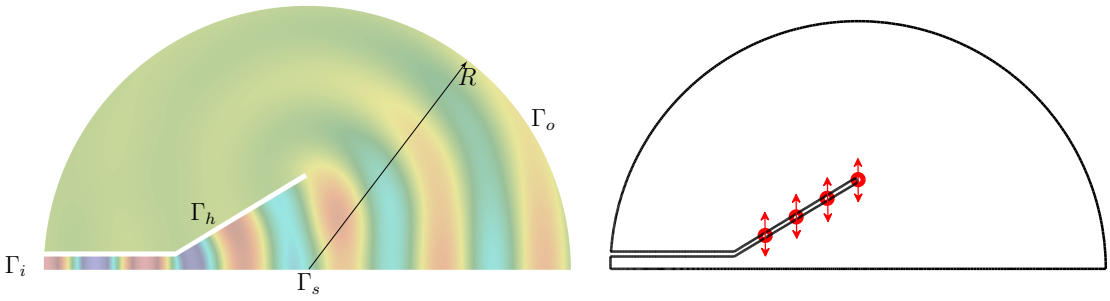


Fig. 3.1 On the left: acoustic horn domain and boundaries (background coloring given by $\Re(p)$ for $f = 900$ Hz). On the right: RBF control points (red circles) whose vertical displacement is treated as a parameter.

the vertical displacement of the control points reported in Fig. 3.1. As a result, we end up with a vector $\boldsymbol{\mu} = [f\ \boldsymbol{\mu}_g]$ of five parameters. The output of interest is the index of reflection intensity (IRI) [BNB03] defined as

$$J(\boldsymbol{\mu}) = \left| \frac{1}{|\Gamma_i|} \int_{\Gamma_i} p(\boldsymbol{\mu})\, d\Gamma - 1 \right|,$$

which measures the transmission efficiency of the device.

Let us now define $X = H^1(\Omega(\boldsymbol{\mu}_g))$ as the space of complex-valued square-integrable functions with square-integrable gradients. The weak formulation of (3.3) reads: given $\boldsymbol{\mu} \in \mathcal{D}$, find $p \in X$ such that

$$a(p, v; \boldsymbol{\mu}) = g(v; \boldsymbol{\mu}) \qquad \forall v \in X, \tag{3.4}$$

where the bilinear and linear forms $a(\cdot, \cdot; \boldsymbol{\mu})$ and $g(\cdot; \boldsymbol{\mu})$ are given, respectively, by

$$a(p, v; \boldsymbol{\mu}) = \int_{\Omega(\boldsymbol{\mu}_g)} \left\{ \nabla p \cdot \nabla \bar{v} - \kappa^2 p \bar{v} \right\} d\Omega + i\kappa \int_{\Gamma_o \cup \Gamma_i} p \bar{v}\, d\Gamma + \frac{1}{2R} \int_{\Gamma_o} p \bar{v}\, d\Gamma, \tag{3.5}$$

$$g(v; \boldsymbol{\mu}) = 2i\kappa A \int_{\Gamma_i} \bar{v}\, d\Gamma. \tag{3.6}$$

We introduce a conforming triangulation $\mathcal{K}_h = \{\Delta_k\}_{k=1}^{n_e}$ of the domain Ω and then we discretize problem (3.4) by approximating X with a finite element space X_h generated by a set of N_h piecewise polynomial nodal basis functions $\{\phi_i\}_{i=1}^{N_h}$. We end up with the following linear system of dimension N_h

$$\mathbf{A}(\boldsymbol{\mu})\mathbf{p}_h = \mathbf{g}(\boldsymbol{\mu}), \tag{3.7}$$

where

$$\mathbf{A}_{ij}(\boldsymbol{\mu}) = a(\phi_j, \phi_i; \boldsymbol{\mu}), \qquad \mathbf{g}_i(\boldsymbol{\mu}) = g(\phi_i; \boldsymbol{\mu}), \qquad 1 \le i, j \le N_h.$$

Our final goal is to use this full-order model not only to analyze the performance of different geometrical configurations, but also to find the optimal shape which maximizes the horn transmission efficiency. To this end, we are required to solve system (3.7) for many different parameter configurations, a computationally intensive task which motivates the development of a suitable inexpensive and fast reduced-order model.

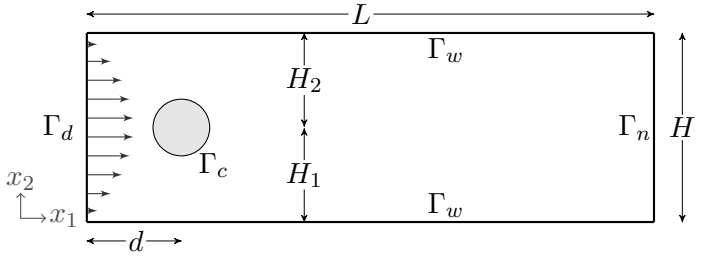


Fig. 3.2 Computational domain. The channel dimensions are: length $L = 2.2$, height $H = H_1+H_2$ with $H_1 = 0.2$, $H_2 = 0.21$. The cylinder is centered at coordinates (d, H_1) with $d = 0.2$, while its radius is $r = 0.05$.

3.2.2 Two-dimensional flow and heat transfer past a cylinder

As a second application, we consider a heat transfer problem for a flow around a circular cylinder (see Fig. 3.2). Since we are interested in a forced-convection problem, the effect of buoyancy and compressibility are neglected, and therefore only a one-way coupling from the fluid equations to the temperature equation is considered. In particular, we assume the fluid dynamics to be governed by the unsteady incompressible Navier-Stokes equations, while its temperature evolution $C = C(\mathbf{x}, t)$ is described by the following advection-diffusion equation

$$\begin{cases} \dfrac{\partial C}{\partial t} + \boldsymbol{v} \cdot \nabla C - \alpha_c \Delta C = 0 & \text{in } \Omega \times (0,T) \\ C = 0 & \text{in } \Gamma_d \times (0,T) \\ C = C_2 & \text{in } \Gamma_c \times (0,T) \\ \alpha_c \nabla C \cdot \boldsymbol{n} = 0 & \text{in } \Gamma_w \cup \Gamma_n \times (0,T) \\ C(\mathbf{x}, 0) = 0 & \text{in } \Omega, \end{cases} \tag{3.8}$$

where $\boldsymbol{v} = \boldsymbol{v}(\mathbf{x}, t)$ is the fluid velocity (solution of the Navier-Stokes equations), α_c is the thermal diffusivity and $C_2 \in \mathbb{R}$ is given. The problem is parametrized with respect to the Reynolds number (entering in the Navier-Stokes equations and thus affecting equation (3.8) through the velocity field $\boldsymbol{v}$), the temperature C_2 imposed on the cylinder, and the thermal diffusivity, so that $\boldsymbol{\mu} = [\text{Re}\ C_2\ \alpha_c]$.

Since the problem is highly transport-dominated, we discretize (3.8) using SUPG stabilized linear finite elements [QV94]. We first introduce a triangulation $\mathcal{T}_h$ of the domain, a conforming finite element approximation space X_h and a lifting operator to take into account the nonhomogeneous Dirichlet condition. Then, we end up with the following semi-discrete weak formulation: for all $t \in (0, T]$, find $C_h(t) \in X_h$ such that for all $\phi \in X_h$

$$\begin{aligned} &\left(\frac{\partial C_h}{\partial t} + \boldsymbol{v}_h(t;\boldsymbol{\mu}) \cdot \nabla C_h, \phi \right) + \left(\alpha_c \nabla C_h, \nabla \phi \right) \\ &+ \sum_{K \in \mathcal{T}_h} \left(\frac{\partial C_h}{\partial t} + \boldsymbol{v}_h(t;\boldsymbol{\mu}) \cdot \nabla C_h - \alpha_c \Delta C_h,\ \tau_K(t;\boldsymbol{\mu}) \boldsymbol{v}_h(t;\boldsymbol{\mu}) \cdot \nabla \phi \right)_K = g(\phi; t; \boldsymbol{\mu}). \end{aligned} \tag{3.9}$$

Here, $g(\cdot; t; \boldsymbol{\mu})$ encodes the action of the nonhomogeneous Dirichlet condition, while the stabilization parameter $\tau_K(t; \boldsymbol{\mu})$ is defined as in [BCTH07]

$$\tau_K(t; \boldsymbol{\mu}) = \left(\frac{4}{(\Delta t)^2} + \boldsymbol{v}_h(t; \boldsymbol{\mu}) \cdot \mathbf{G}_K \boldsymbol{v}_h(t; \boldsymbol{\mu}) + \alpha_c^2 \mathbf{G}_K : \mathbf{G}_K \right)^{-1/2}, \tag{3.10}$$

where $\mathbf{G}_K$ denotes the covariant metric tensor of the computational domain (see Sect. 3.7.2 for its definition). At the algebraic level, problem (3.9) leads to the following discrete dynamical system

$$\mathbf{M}(t; \boldsymbol{\mu}) \frac{d\mathbf{C}_h}{dt} + \mathbf{A}(t; \boldsymbol{\mu}) \mathbf{C}_h = \mathbf{g}(t; \boldsymbol{\mu}). \tag{3.11}$$

Given a value of the parameters $\boldsymbol{\mu}$, solving (3.11) requires to first solve a suitable full-order approximation of the unsteady Navier-Stokes equations to obtain the transport field $\boldsymbol{v}_h(t; \boldsymbol{\mu})$, which is then inserted into (3.11) to compute the temperature field $\mathbf{C}_h(t; \boldsymbol{\mu})$. Repeating these operations for many different system configurations requires a considerable computational effort, thus motivating the need for model order reduction. However, since the matrices $\mathbf{A}(t; \boldsymbol{\mu})$, $\mathbf{M}(t; \boldsymbol{\mu})$ and the vector $\mathbf{g}(t; \boldsymbol{\mu})$ feature a highly nonaffine and implicit (through $\boldsymbol{v}_h(t; \boldsymbol{\mu})$) dependence with respect to the parameters, any model reduction strategy would be ineffective without an accompanying system approximation technique.

Our approach in the construction of a suitable ROM for the entire problem features two levels of reduction, taking advantage of the one-way coupling from the Navier-Stokes equations to the temperature equation. Indeed, we shall first build a ROM for the fluid dynamics equations, yielding a low-dimensional approximation $\boldsymbol{v}_N(t; \boldsymbol{\mu})$ of the full-order transport field $\boldsymbol{v}_h(t; \boldsymbol{\mu})$. Then, on top of this, we construct a ROM for (3.11).

3.3 Parametrized matrix interpolation

After a brief review of the basic features of the discrete empirical interpolation method, we describe in this section a very efficient procedure to deal with the interpolation of parametrized matrices, according to the needs outlined above.

3.3.1 Review of the (discrete) empirical interpolation method

EIM [BMNP04, MNPP09] and its so-called discrete variant DEIM [CS10] are interpolation techniques for the approximation of parameter-dependent functions based on a greedy selection of interpolation points and a projection over a low-dimensional space. In more details, EIM and DEIM approximate a nonlinear function $\mathbf{f} \colon \tau \in \mathcal{T} \subset \mathbb{R}^p \to \mathbf{f}(\tau) \in \mathbb{R}^{N_h}$ (here τ could represent parameters $\boldsymbol{\mu}$, time t or both) by projection onto a low-dimensional subspace spanned by a basis $\boldsymbol{\Phi}$,

$$\mathbf{f}(\tau) \approx \mathbf{f}_m(\tau) = \boldsymbol{\Phi} \boldsymbol{\theta}(\tau),$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_M] \in \mathbb{R}^{N_h \times M}$ and $\boldsymbol{\theta}(\tau) \in \mathbb{R}^M$ is the corresponding vector of coefficients, with $M \ll N_h$. Let us recall how EIM and DEIM select offline the basis $\boldsymbol{\Phi}$ and computes online the coefficients $\boldsymbol{\theta}(\tau)$.

(i) Both methods start by constructing a set of snapshots obtained by sampling $\mathbf{f}(\tau)$ at values τ_i, $i = 1, \dots, n_s$. Then, DEIM applies POD to extract a basis from the snapshots, i.e.

$$[\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_M] = \mathrm{POD}([\mathbf{f}(\tau_1), \dots, \mathbf{f}(\tau_{n_s})], \varepsilon_{\mathrm{POD}}),$$

where $\varepsilon_{\mathrm{POD}}$ is a prescribed tolerance. The POD procedure is summarized in Algorithm 3.1. On the other hand, EIM computes a basis through a greedy algorithm, where the new basis function is a suitable scaling and shifting of the snapshot which is worst approximated by the current basis.

(ii) Given a new τ, in order to compute the coefficients vector $\boldsymbol{\theta}(\tau)$, both EIM and DEIM impose interpolation conditions at some properly selected entries $\mathcal{I} \subset \{1, \cdots, N_h\}$, $|\mathcal{I}| = M$, of the vector $\mathbf{f}(\tau)$:

$$\boldsymbol{\Phi}_{\mathcal{I}}\, \boldsymbol{\theta}(\tau) = \mathbf{f}_{\mathcal{I}}(\tau), \tag{3.12}$$

where $\boldsymbol{\Phi}_{\mathcal{I}} \in \mathbb{R}^{M \times M}$ is the matrix formed by the $\mathcal{I}$ rows of $\boldsymbol{\Phi}$. As a result,

$$\mathbf{f}_m(\tau) = \boldsymbol{\Phi} \boldsymbol{\Phi}_{\mathcal{I}}^{-1} \mathbf{f}_{\mathcal{I}}(\tau). \tag{3.13}$$

(iii) In both cases, the indices $\mathcal{I}$ are iteratively selected from the basis $\boldsymbol{\Phi}$ using a greedy procedure which minimizes the interpolation error over the snapshots set measured in the infinity norm.

The two procedures are summarized in Algorithms 3.2 and 3.3. The main difference between DEIM and EIM lies in the selection of the basis $\boldsymbol{\Phi}$, based on the POD technique rather than on a greedy algorithm. As such, the DEIM simply consists in the application of the EIM algorithm for the selection of the interpolation indices $\mathcal{I}$ to a preexisting basis obtained by POD. See, e.g., [BMS14] for further details.

As in the following we shall concentrate on DEIM, it is useful to recall that the error between $\mathbf{f}$ and its DEIM approximation $\mathbf{f}_m$ can be bounded as follows (see e.g. [CS10] for the complete derivation)

$$\|\mathbf{f}(\tau) - \mathbf{f}_m(\tau)\|_2 \leq \|\boldsymbol{\Phi}_{\mathcal{I}}^{-1}\|_2 \, \|(\mathbf{I} - \boldsymbol{\Phi}\boldsymbol{\Phi}^T)\mathbf{f}(\tau)\|_2; \tag{3.14}$$

the second factor in the right-hand side of (3.14) can be approximated as

$$\|(\mathbf{I} - \boldsymbol{\Phi}\boldsymbol{\Phi}^T)\mathbf{f}(\tau)\|_2 \approx \sigma_{M+1}, \tag{3.15}$$

where σ_{M+1} is the first discarded singular value in the POD procedure. This approximation holds for any parameter τ provided an appropriate sampling of the snapshots in the parameter space has been carried out. In that case, the predictive projection error $\|(\mathbf{I} - \boldsymbol{\Phi}\boldsymbol{\Phi}^T)\mathbf{f}(\tau)\|_2$ is comparable to the training projection error σ_{M+1}. We will employ this error bound in the following sections when dealing with the construction of a ROM for the parametrized problems at hand.

Remark 3.1. We also remark that the interpolation condition (3.12) can be generalized to the case where more sample indices ($|\mathcal{J}| > M$) than basis functions are considered, leading to a gappy POD reconstruction [ES95, CFCA13]

$$\boldsymbol{\theta}(\tau) = \arg\min_{\mathbf{x} \in \mathbb{R}^M} \|\mathbf{f}_{\mathcal{I}}(\tau) - \boldsymbol{\Phi}_{\mathcal{I}}\, \mathbf{x}\|_2. \tag{3.16}$$

The solution of the least-squares problem (3.16) yields $\mathbf{f}_m(\tau) = \boldsymbol{\Phi}\boldsymbol{\Phi}_{\mathcal{I}}^{+}\mathbf{f}_{\mathcal{I}}(\tau)$, where $\boldsymbol{\Phi}_{\mathcal{I}}^{+}$ is the Moore-Penrose pseudoinverse of the matrix $\boldsymbol{\Phi}_{\mathcal{I}}$. •

Input: Set of snapshots $\mathbf{\Lambda} = [\mathbf{f}^1, \dots, \mathbf{f}^{n_s}] \in \mathbb{R}^{N_h \times n_s}$, tolerance ε_{POD}
Output: Orthonormal basis $\mathbf{\Phi} \in \mathbb{R}^{N_h \times M}$

1: **if** $n_s \leq N_h$ **then**
2: form the correlation matrix $\mathbf{C} = \mathbf{\Lambda}^T \mathbf{\Lambda}$
3: solve the eigenvalue problem $\mathbf{C}\boldsymbol{\psi}_i = \sigma_i^2 \boldsymbol{\psi}_i, \quad i = 1, \dots, n_s$
4: set $\boldsymbol{\phi}_i = \frac{1}{\sigma_i} \mathbf{\Lambda} \boldsymbol{\psi}_i$
5: **else**
6: form the matrix $\mathbf{K} = \mathbf{\Lambda}\mathbf{\Lambda}^T$
7: solve the eigenvalue problem $\mathbf{K}\boldsymbol{\phi}_i = \sigma_i^2 \boldsymbol{\phi}_i, \quad i = 1, \dots, N_h$
8: **end if**
9: define M as the minimum integer such that

$$I(M) = \frac{\sum_{i=1}^{M} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2} \geq 1 - \varepsilon_{\text{POD}}^2, \quad \text{where } r = \text{rank}(\mathbf{\Lambda})$$

10: form the basis $\mathbf{\Phi} = [\boldsymbol{\phi}_1 \mid \dots \mid \boldsymbol{\phi}_M]$

Algorithm 3.1 POD procedure. Given a snapshots matrix $\mathbf{\Lambda} \in \mathbb{R}^{N_h \times n_s}$ and a tolerance $\varepsilon_{\text{POD}} > 0$, the algorithm returns an orthonormal basis $\mathbf{\Phi} \in \mathbb{R}^{N_h \times M}$. Alternatively, one can provide the (a priori fixed) dimension $M \leq n_s$ of the basis rather than ε_{POD}, and skip line 9. If one is interested in accurately computing the POD modes associated to the smallest singular values, the use of a thin singular value decomposition [GVL13] is recommended.

Input: Set of snapshots $\mathbf{\Lambda} = [\mathbf{f}(\tau_1), \dots, \mathbf{f}(\tau_{n_s})] \in \mathbb{R}^{N_h \times n_s}$, tolerance ε_{EIM}, maximum number of iterations $M_{\max}$
Output: Basis $\mathbf{\Phi} \in \mathbb{R}^{N_h \times M}$, set of indices $\mathcal{I} \in \mathbb{R}^M$

$M = 0$, $e_0 = \varepsilon_{\text{EIM}} + 1$
$q = \arg\max_{j=1,\dots,n_s} \|\mathbf{f}(\tau_j)\|_\infty$
$\mathbf{\Phi} = [\,]$, $\mathcal{I} = \emptyset$
$\mathbf{r} = \mathbf{f}(\tau_q)$
while $M < M_{\max}$ and $e_M > \varepsilon_{\text{EIM}}$
  $M \leftarrow M + 1$
  $i = \arg\max_{\{1,\dots,N_h\}} |\mathbf{r}|$
  $\mathcal{I} \leftarrow \mathcal{I} \cup i$
  $\boldsymbol{\phi}_M = \mathbf{r} / \mathbf{r}_i$
  $\mathbf{\Phi} \leftarrow [\mathbf{\Phi}\ \boldsymbol{\phi}_M]$
  $[e_M, q] = \max_{j=1,\dots,n_s} \|\mathbf{f}(\tau_j) - \mathbf{\Phi}\mathbf{\Phi}_{\mathcal{I}}^{-1}(\mathbf{f}(\tau_j))_{\mathcal{I}}\|_\infty$
  $\mathbf{r} = \mathbf{f}(\tau_q) - \mathbf{\Phi}\mathbf{\Phi}_{\mathcal{I}}^{-1}(\mathbf{f}(\tau_q))_{\mathcal{I}}$
end while

Algorithm 3.2 Empirical interpolation method. Here, the *max* operation returns both the maximum value and the index where the maximum occurs.

Input: Set of snapshots $\boldsymbol{\Lambda} = [\mathbf{f}(\tau_1), \dots, \mathbf{f}(\tau_{n_s})] \in \mathbb{R}^{N_h \times n_s}$, tolerance ε_{POD}
Output: Basis $\boldsymbol{\Phi} \in \mathbb{R}^{N_h \times M}$, set of indices $\mathcal{I} \in \mathbb{R}^M$

$[\boldsymbol{\phi}_1 \cdots \boldsymbol{\phi}_M] = \text{POD}(\boldsymbol{\Lambda}, \varepsilon_{\text{POD}})$
$i = \arg\max_{\{1,\dots,N_h\}} |\boldsymbol{\phi}_1|$
$\boldsymbol{\Phi} = \boldsymbol{\phi}_1$, $\mathcal{I} = \{i\}$
for $k = 2 : M$
    $\mathbf{r} = \boldsymbol{\phi}_k - \boldsymbol{\Phi}\boldsymbol{\Phi}_{\mathcal{I}}^{-1}(\boldsymbol{\phi}_k)_{\mathcal{I}}$
    $i = \arg\max_{\{1,\dots,N_h\}} |\mathbf{r}|$
    $\mathcal{I} \leftarrow \mathcal{I} \cup i$
    $\boldsymbol{\Phi} \leftarrow [\boldsymbol{\Phi}\ \boldsymbol{\phi}_k]$
end for

Algorithm 3.3 Discrete empirical interpolation method. Given a vector $\mathbf{v} \in \mathbb{R}^{N_h}$, the *arg max* operation returns the index $i \in \{1, \dots, N_h\}$ where the maximum entry occurs. Here, $|\mathbf{v}|$ denotes the vector of components $[\,|v_1| \cdots |v_{N_h}|\,]^T$.

3.3.2 Shortcomings of DEIM for nonaffinely parametrized PDEs

Employing DEIM (as well as EIM) to deal with nonaffinely parametrized PDEs is not an easy task, as it usually entails an extensive work on the continuous formulation of the problem, as well as intrusive changes to its high-fidelity implementation, very often preventing the use of existing solvers. Moreover, employing this kind of techniques to approximate parametrized functions accounting for geometrical deformations and/or physical properties easily leads to a very large number of affine terms. For the sake of illustration, we provide here two simple examples.

Following the notation of Section 3.2.1, we first consider a matrix $\mathbf{K}(\tau) \in \mathbb{R}^{N_h \times N_h}$ arising from the finite element discretization of a diffusion-reaction equation with variable coefficients $\alpha : \Omega \times \mathcal{T} \to \mathbb{R}$ and $\gamma : \Omega \times \mathcal{T} \to \mathbb{R}$,

$$\mathbf{K}_{ij}(\tau) = \int_\Omega \alpha(\mathbf{x};\tau)\, \nabla\varphi_j \cdot \nabla\varphi_i \, d\Omega + \int_\Omega \gamma(\mathbf{x};\tau)\, \varphi_j\, \varphi_i \, d\Omega, \qquad 1 \le i, j \le n. \tag{3.17}$$

If $\alpha(\mathbf{x};\tau)$ and $\gamma(\mathbf{x};\tau)$ are nonaffinely parametrized with respect to $\mathbf{x}$ and τ, we can generate an approximate affine expansion of $\mathbf{K}(\tau)$ as

$$\begin{aligned}\mathbf{K}_{ij}(\tau) &\approx \int_\Omega \alpha_m(\mathbf{x};\tau)\, \nabla\varphi_j \cdot \nabla\varphi_i \, d\Omega + \int_\Omega \gamma_m(\mathbf{x};\tau)\, \varphi_j\, \varphi_i \, d\Omega \\ &= \sum_{q=1}^{M_\alpha} \theta_q^\alpha(\tau) \int_\Omega \phi_q^\alpha(\mathbf{x})\, \nabla\varphi_j \cdot \nabla\varphi_i \, d\Omega + \sum_{q=1}^{M_\gamma} \theta_q^\gamma(\tau) \int_\Omega \phi_q^\gamma(\mathbf{x})\, \varphi_j\, \varphi_i \, d\Omega, \end{aligned} \tag{3.18}$$

where

$$\alpha_m(\mathbf{x};\tau) = \sum_{q=1}^{M_\alpha} \theta_q^\alpha(\tau)\, \phi_q^\alpha(\mathbf{x}), \qquad \gamma_m(\mathbf{x};\tau) = \sum_{q=1}^{M_\gamma} \theta_q^\gamma(\tau)\, \phi_q^\gamma(\mathbf{x})$$

are DEIM approximations of $\alpha(\mathbf{x};\tau)$ and $\gamma(\mathbf{x};\tau)$, respectively. In practice, they are computed by applying DEIM to the vector functions $\boldsymbol{\alpha}(\tau)$, $\boldsymbol{\gamma}(\tau)$ obtained by evaluating

$\alpha(\cdot;\tau)$, $\gamma(\cdot;\tau)$ in the finite element quadrature points. This way of proceeding is extremely problem-specific and often inefficient, as it completely ignores the possible common dependence from τ of the two coefficients. For instance, if $\gamma = \alpha$ we would still have $2M_\alpha$, rather than only M_α, affine terms.

The procedure is even more involved in the case of geometrical parametrizations, which require to first pull back the weak formulation of the original problem to a reference configuration, and then perform DEIM (or EIM) on each term of the tensors accounting for the geometrical deformation. Let us consider for instance the matrix

$$\mathbf{K}_{ij}(\tau) = \int_{\Omega(\tau)} \nabla\varphi_j \cdot \nabla\varphi_i \, d\Omega, \qquad 1 \le i,j \le N_h \tag{3.19}$$

resulting from the finite element discretization of the Laplace operator on a parametrized domain $\Omega(\tau) \subset \mathbb{R}^d$. This latter is obtained from a reference configuration $\widetilde{\Omega}$ through a parametric map $\mathbf{F} : \widetilde{\Omega} \times \mathcal{T} \to \mathbb{R}^d$ such that $\Omega(\tau) = \mathbf{F}(\widetilde{\Omega};\tau)$. By operating a suitable change of variables, we can rewrite the integral in (3.19) as

$$\mathbf{K}_{ij}(\tau) = \int_{\widetilde{\Omega}} \boldsymbol{\nu}(\mathbf{x};\tau)\, \nabla\varphi_j \cdot \nabla\varphi_i \, d\Omega,$$

where $\boldsymbol{\nu}(\mathbf{x};\tau) = (\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{x};\tau))^{-1}(\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{x};\tau))^{-T}|\det(\nabla_{\mathbf{x}}\mathbf{F}(\mathbf{x};\tau))|$ is a $d \times d$ matrix, usually nonlinearly parametrized with respect to τ. In order to obtain an affine expression for $\mathbf{K}(\tau)$, each component of $\boldsymbol{\nu}$ has to be approximated by DEIM, yielding

$$\mathbf{K}_{ij}(\tau) \approx \sum_{k,l=1}^{d} \int_{\widetilde{\Omega}} (\boldsymbol{\nu}_m(\mathbf{x};\tau))_{k,l}\, \nabla\varphi_j \cdot \nabla\varphi_i \, d\Omega = \sum_{k,l=1}^{d} \sum_{q=1}^{M_{kl}} \theta_q^{kl}(\tau) \int_{\widetilde{\Omega}} \phi_q^{kl}(\mathbf{x})\, \nabla\varphi_j \cdot \nabla\varphi_i \, d\Omega.$$

Therefore, to obtain an affine approximation of $\mathbf{K}(\tau)$, we are first required to approximate by DEIM (at most) d^2 functions and then to assemble $\sum_{k,l=1}^{d} M_{kl}$ τ-independent matrices. This, however, requires knowledge of the analytical expression of the geometric map and its gradient, as well as an ad-hoc implementation, thus resulting in several intrusive operations. Indeed, as long as the type of parametrization changes or another differential operator is considered, a (conceptually similar but practically) different procedure has to be put in place. We refer to, e.g., [NP08, LR10, MQR12b, AHS14] for more detailed examples of DEIM and EIM applications in this context. In order to overcome these shortcomings, we turn to a purely algebraic perspective.

3.3.3 Matrix discrete empirical interpolation method

As suggested in [CTB12, CTB15, BGW15, WSH14], we can use DEIM to address the following problem: given a parametrized matrix $\mathbf{K}(\tau)\colon \mathcal{T} \mapsto \mathbb{R}^{N_h \times N_h}$, find $M \ll N_h$ functions $\theta_q\colon \mathcal{T} \mapsto \mathbb{R}$ and parameter-independent matrices $\mathbf{K}_q \in \mathbb{R}^{N_h \times N_h}$, $1 \le q \le M$, such that

$$\mathbf{K}(\tau) \approx \mathbf{K}_m(\tau) = \sum_{q=1}^{M} \theta_q(\tau)\, \mathbf{K}_q. \tag{3.20}$$

The offline procedure consists of two main steps. First we express the matrix $\mathbf{K}(\tau)$ in vector format by defining $\mathbf{k}(\tau) = \mathrm{vec}(\mathbf{K}(\tau)) \in \mathbb{R}^{N_h^2}$ ($\mathbf{k}(\tau)$ is obtained by stacking the

columns of $\mathbf{K}(\tau)$, see, e.g., [GVL13, Chap. 1]), so that (3.20) can be reformulated as: find $\{\boldsymbol{\Phi}, \boldsymbol{\theta}(\tau)\}$ such that

$$\mathbf{k}(\tau) \approx \mathbf{k}_m(\tau) = \boldsymbol{\Phi}\boldsymbol{\theta}(\tau), \tag{3.21}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{N_h^2 \times M}$ is a τ-independent basis and $\boldsymbol{\theta}(\tau) \in \mathbb{R}^M$ the corresponding coefficients vector, that is

$$\boldsymbol{\Phi} = [\,\mathrm{vec}(\mathbf{K}_1), \ldots, \mathrm{vec}(\mathbf{K}_M)\,], \qquad \boldsymbol{\theta}(\tau) = [\theta_1(\tau), \ldots, \theta_M(\tau)]^T.$$

Then, we apply DEIM to a set of snapshots $\boldsymbol{\Lambda} = [\mathbf{k}(\tau_1), \ldots, \mathbf{k}(\tau_{n_s})]$ in order to obtain the basis $\boldsymbol{\Phi}$ and the interpolation indices $\mathcal{I} \subset \{1, \cdots, N_h^2\}$.

During the online phase, given a new $\tau \in \mathcal{T}$, we can compute $\mathbf{k}_m(\tau)$ as

$$\mathbf{k}_m(\tau) = \boldsymbol{\Phi}\boldsymbol{\theta}(\tau), \qquad \text{with} \quad \boldsymbol{\theta}(\tau) = \boldsymbol{\Phi}_{\mathcal{I}}^{-1}\mathbf{k}_{\mathcal{I}}(\tau). \tag{3.22}$$

Then, reversing the vec operation, we get the MDEIM approximation $\mathbf{K}_m(\tau)$ to the matrix $\mathbf{K}(\tau)$. We point out that, for the sake of model order reduction, the crucial step in the online evaluation of $\mathbf{K}_m(\tau)$ is the computation of $\mathbf{k}_{\mathcal{I}}(\tau)$. The following subsection is devoted to the illustration of the details related with this procedure.

Remark 3.2. In the finite element context, the entire procedure is implemented using a suitable sparse format for the matrix $\mathbf{K}(\tau)$. Therefore, the actual dimension of the vectorized matrices is n_z rather than N_h^2, where n_z denotes the number of nonzero entries of the matrix $\mathbf{K}(\tau)$. As such, one could reuse standard POD and DEIM routines for generating the POD basis $\boldsymbol{\Phi}$ and the interpolation indices $\mathcal{I}$. However, depending on the software which is used and the actual implementation, these operations can be quite demanding in terms of time and memory resources (see also [SS14]). For instance, using the compressed-column storage format [GVL13] available in MATLAB, computing the singular value decomposition of the sparse snapshots matrix $\boldsymbol{\Lambda}$ is quite inefficient. To this end, it is more convenient to define an auxiliary, dense snapshots matrix $\widetilde{\boldsymbol{\Lambda}} \in \mathbb{R}^{n_z \times M}$ such that

$$\widetilde{\boldsymbol{\Lambda}}_{i,j} = \boldsymbol{\Lambda}_{Z(i),j}, \qquad i = 1, \ldots, n_z,\ j = 1, \ldots, M,$$

where $Z \subset \{1, \ldots, N_h^2\}$ is the set of indices corresponding to the rows of the matrix $\boldsymbol{\Lambda}$ having at least a nonzero entry and $n_z = |Z|$. Running standard POD and DEIM routines on $\widetilde{\boldsymbol{\Lambda}}$ we obtain a dense POD basis $\widetilde{\boldsymbol{\Phi}} \in \mathbb{R}^{n_z \times M}$ and a set of interpolation indices $\widetilde{\mathcal{I}} \subset \{1, \ldots, n_z\}$. Then, the nonzero entries of the sparse POD basis $\boldsymbol{\Phi} \in \mathbb{R}^{N_h^2 \times M}$ are given by

$$\boldsymbol{\Phi}_{Z(i),j} = \widetilde{\boldsymbol{\Phi}}_{i,j}, \qquad i = 1, \ldots, n_z,\ j = 1, \ldots, M,$$

while $\mathcal{I} = Z(\widetilde{\mathcal{I}}) \subset \{1, \ldots, n^2\}$. •

Remark 3.3. A similar Matrix EIM (MEIM) procedure could have been obtained by employing EIM rather than DEIM to approximate the vectorized matrix $\mathbf{k}(\tau)$. As already mentioned, the main difference lies in the selection of the basis, based on a greedy algorithm rather than on the POD technique, see e.g. [BMS14] for further details. In this respect, we note that a "multi-component EIM" has been proposed and analyzed in [Ton12]. •

3.3.4 Efficient evaluation of $\mathbf{k}_{\mathcal{I}}(\tau)$ in the finite element context

The evaluation of $\mathbf{K}_m(\tau)$ requires the computation of the values of the entries $\mathcal{I}$ of the vectorized matrix $\mathbf{k}(\tau)$. If the matrix $\mathbf{K}(\tau)$ results from the discretization of a PDE operator, the efficient implementation of this operation requires ad-hoc techniques depending on the underlying high-fidelity approximation. In the finite element (FE) context, efficiency is achieved thanks to the local support of the basis functions, which delivers the usual element-wise approach for the matrix assembly. Indeed, to compute $\mathbf{k}_{\mathcal{I}}(\tau)$, we can reuse the available high-fidelity matrix assemblers by simply restricting the *loop over the elements* to those elements which provide a nonzero contribution to the entries $\mathcal{I}$ of $\mathbf{k}(\tau)$. This requires to perform some pre-processing operations in the offline phase (once and for all). Starting from the data structures defining a finite element mesh (i.e. *nodes, (internal) elements* and *boundary elements*), we need to:

1. detect the *reduced degrees of freedom (dofs)*: to each index $i \in \mathcal{I}$ in the vector format corresponds a pair of row-column indexes $(l, j) \in I \times J$ in the matrix format (with $I, J \subset \{1, \cdots, N_h\}$). We define the *reduced dofs* as the union of the sets I and J;
2. define the *reduced nodes* of the mesh as the set of nodes associated to the *reduced dofs* (in the case of vectorial problems, see e.g. Sect. 6.2, we could have more than one reduced dof corresponding to the same reduced node);
3. detect the *reduced elements* of the mesh, which are defined as the elements containing at least one *reduced node*. Similarly define the *reduced boundary elements* as the set of boundary elements containing at least one *reduced node*.

These new data structures identify the *reduced mesh* (see Fig. 3.3), also called *sample mesh* [CFCA13] or *reduced integration domain* [Ryc09]. Once the reduced mesh has been generated and stored in the offline phase, in the online phase we just have to compute the entries of the matrix $\mathbf{K}(\tau)$ corresponding to the reduced elements. As anticipated, this operation can be performed at very low cost by exploiting the same assembler routine used for the high-fidelity simulations. The resulting matrix $\widehat{\mathbf{K}}(\tau)$ has still dimension $N_h \times N_h$, but it is extremely sparse since only the entries associated to the reduced elements are actually nonzero. Then, we obtain $\mathbf{k}_{\mathcal{I}}(\tau)$ by vectorizing $\widehat{\mathbf{K}}(\tau)$ and extracting the entries $\mathcal{I}$, i.e.

$$\mathbf{k}_{\mathcal{I}}(\tau) = \Big(\mathrm{vec}(\widehat{\mathbf{K}}(\tau))\Big)_{\mathcal{I}}.$$

Given $\mathbf{k}_{\mathcal{I}}(\tau)$, we can finally compute $\boldsymbol{\theta}(\tau)$ and thus $\mathbf{k}_m(\tau)$ by (3.22).

Remark 3.4. In the context of the finite element method, an unassembled variant of DEIM was developed in [Ded12, TR13, AHS14]. The Unassembled DEIM (UDEIM) differs from the DEIM in the sense that unassembled quantities are approximated. This results in a larger number of rows in the vector-valued function approximated by UDEIM but, when a reduced node is selected, only one attached reduced element is associated, resulting in a sparser reduced mesh in the online phase. A detailed comparison between DEIM and UDEIM is reported in [AHS14]. The main drawbacks associated with the UDEIM are (i) the larger dimension of vectors and matrices one has to deal with during

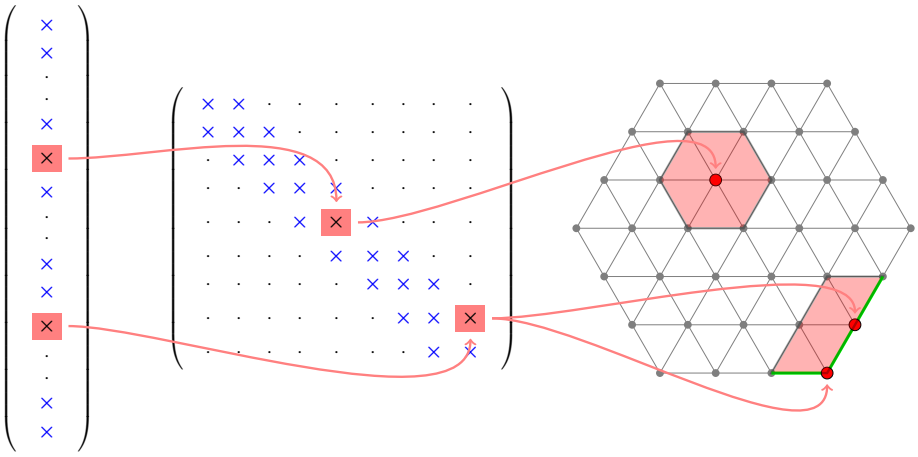


Fig. 3.3 Reduced mesh concept in the case of $\mathbb{P}_1$ finite elements. On the left: sparsity pattern of the vectorized matrix $\mathbf{K}$; blue crosses identify the nonzero entries, while the red boxes correspond to the DEIM entries $\mathcal{I}$. In the middle: red boxes correspond to the reduced dofs (in matrix format). On the right: underlying FE mesh with red circles/triangles denoting the reduced dofs/elements and green lines corresponding to the reduced boundaries.

the offline stage, and (ii) possible substantive modifications to the original high-fidelity code which are required to return unassembled quantities associated with vectors or matrices, rendering it less amenable to black-box approaches. Hence, DEIM is preferred here, but all the developments would be equally applicable with UDEIM. •

3.3.5 Preservation of matrix properties

We finally highlight some properties fulfilled by the reduced matrix $\mathbf{K}_m(\tau)$ with respect to the corresponding original matrix $\mathbf{K}(\tau)$. First of all, thanks to a general result of perturbation theory for eigenvalue problems, the singular values of the reduced matrix approach the singular values of the original matrix as m increases. This is ensured by Weyl-Mirsky theorem (see, e.g., [SS90]) for a general nonsingular matrix: if $\sigma_i(\tau)$, $\sigma_i^m(\tau)$ for $1, \dots, N_h$ denote the singular values of $\mathbf{K}(\tau)$ and $\mathbf{K}_m(\tau)$, respectively, then

$$|\sigma_i(\tau) - \sigma_i^m(\tau)| \leq \|\mathbf{K}(\tau) - \mathbf{K}_m(\tau)\|_2. \tag{3.23}$$

Further, the matrix error $\|\mathbf{K}(\tau) - \mathbf{K}_m(\tau)\|_2$ can be bounded as follows

$$\|\mathbf{K}(\tau) - \mathbf{K}_m(\tau)\|_2 \leq \|\mathbf{K}(\tau) - \mathbf{K}_m(\tau)\|_F = \|\mathbf{k}(\tau) - \mathbf{k}_m(\tau)\|_2 \leq \|\mathbf{\Phi}_{\mathcal{I}}^{-1}\|_2 \, \|(\mathbf{I} - \mathbf{\Phi}\mathbf{\Phi}^T)\mathbf{k}(\tau)\|_2 \tag{3.24}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The first term $\|\mathbf{\Phi}_{\mathcal{I}}^{-1}\|_2$ does not depend on $\mathbf{K}(\tau)$, and thus can be computed just once for all $\tau \in \mathcal{T}$. On the other hand, the second term depends on $\mathbf{K}(\tau)$ and therefore changes for each new $\tau \in \mathcal{T}$, thus being too expensive to compute. Nevertheless, a good approximation for this quantity is given by the first discarded singular value in the POD basis computation, i.e.

$$\|(\mathbf{I} - \mathbf{\Phi}\mathbf{\Phi}^T)\mathbf{k}(\tau)\|_2 \approx \sigma_{M+1}.$$

As already noted in Sect. 3.3.1, this approximation holds for any τ provided that an appropriate sampling of the parameter domain has been operated to construct $\mathbf{\Phi}$.

While a possible symmetry of the original matrix is automatically inherited by its approximation, this is not the case of positive definiteness. Although the relation (3.23) between the singular values of $\mathbf{K}(\tau)$ and $\mathbf{K}_m(\tau)$ ensures that the spectra of these two matrices are close to each other, the positivity of the reduced matrix could be properly enforced (if not automatically satisfied) for those cases requiring this assumption to be fulfilled. For instance, in [CTB15] the authors propose to augment the least-squares problem (3.16) with a generalized linear constraint:

$$\boldsymbol{\theta}(\tau) = \arg\min_{\mathbf{x}\in\mathbb{R}^M} \|\mathbf{k}_{\mathcal{I}}(\tau) - \boldsymbol{\Phi}_{\mathcal{I}}\,\mathbf{x}\|_2 \quad \text{subject to} \quad \sum_{q=1}^{M} x_q \mathbf{V}^T \mathbf{K}_q \mathbf{V} > 0. \tag{3.25}$$

The generalized linear constraint is a classic linear matrix inequality in the variable $\mathbf{x}$ that leads to a convex optimization problem. In the numerical experiments reported in [CTB15] the unconstrained solution always satisfies the coercivity condition. In fact, this is also confirmed by our numerical tests.

An alternative approach would be to perform the interpolation on the manifold of symmetric positive definite (SPD) matrices directly. This guarantees that the resulting matrix is itself SPD. This approach is followed for instance in [ACCF09], when dealing with the interpolation of reduced matrices arising in structural mechanics. However, when applied to full-order matrices this approach is no more viable, since it involves the computation of matrix logarithm and exponential, which destroy the sparsity pattern of the original matrix.

More generally, in the non-symmetric case, we have observed numerically that the MDEIM approximation preserves the non-singularity of the full operator. This observation is very important, as it supports the derivation of the a posteriori error bound in Sect. 3.4.2.

3.4 Hyper-reduction of parametrized elliptic equations

Let us consider the linear, stationary problem (1.2) already introduced in Sect. 1.2.1: given $\boldsymbol{\mu} \in \mathcal{D}$, find $\mathbf{y}_h \in \mathbb{R}^{N_h}$ such that

$$\mathbf{A}(\boldsymbol{\mu})\,\mathbf{y}_h = \mathbf{g}(\boldsymbol{\mu}), \tag{3.26}$$

resulting for instance from the finite element discretization of an elliptic problem, such as the one of Sect. 3.2.1. We assume the matrix $\mathbf{A}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ to be nonsingular for all $\boldsymbol{\mu} \in \mathcal{D}$, so that

$$\beta_h(\boldsymbol{\mu}) = \sigma_{\min}(\mathbf{X}^{-\frac{1}{2}}\mathbf{A}(\boldsymbol{\mu})\mathbf{X}^{-\frac{1}{2}}) > 0 \qquad \forall \boldsymbol{\mu} \in \mathcal{D}, \tag{3.27}$$

where $\beta_h(\boldsymbol{\mu})$ denotes the stability factor with respect to a symmetric positive definite matrix $\mathbf{X} \in \mathbb{R}^{N_h \times N_h}$ defining a suitable vectorial norm for the solution. For instance, in the case of a second-order elliptic problem the matrix $\mathbf{X}$ results from the discretization of the $H^1(\Omega)$ inner product.

When considering the corresponding ROM, we seek $\mathbf{y}_N \in \mathbb{R}^N$ such that

$$\mathbf{A}_N(\boldsymbol{\mu})\,\mathbf{y}_N = \mathbf{g}_N(\boldsymbol{\mu}). \tag{3.28}$$

Our goal is to evaluate efficiently (i.e. with complexity independent of N_h) the reduced operators $\mathbf{A}_N(\boldsymbol{\mu}) = \mathbf{W}^T\mathbf{A}(\boldsymbol{\mu})\mathbf{V}$, $\mathbf{g}_N(\boldsymbol{\mu}) = \mathbf{W}^T\mathbf{g}(\boldsymbol{\mu})$ when $\mathbf{A}(\boldsymbol{\mu})$ and $\mathbf{g}(\boldsymbol{\mu})$ are nonaffine functions of $\boldsymbol{\mu}$. Thanks to MDEIM, we can approximate $\mathbf{A}(\boldsymbol{\mu})$ as

$$\mathbf{A}(\boldsymbol{\mu}) \approx \mathbf{A}_m(\boldsymbol{\mu}) = \sum_{q=1}^{M_a} \theta_q^a(\boldsymbol{\mu})\, \mathbf{A}_q, \tag{3.29}$$

so that

$$\mathbf{A}_N(\boldsymbol{\mu}) \approx \mathbf{A}_N^m(\boldsymbol{\mu}) = \sum_{q=1}^{M_a} \theta_q^a(\boldsymbol{\mu})\, \mathbf{A}_N^q. \tag{3.30}$$

Here, $\mathbf{A}_N^q = \mathbf{W}^T\mathbf{A}_q\mathbf{V} \in \mathbb{R}^{N\times N}$, $q = 1, \dots, M_a$, are precomputable matrices of small dimension. The corresponding weights $\boldsymbol{\theta}^a(\boldsymbol{\mu}) = [\theta_1^a(\boldsymbol{\mu}) \cdots \theta_{M_a}^a(\boldsymbol{\mu})]$ are given by

$$\boldsymbol{\theta}^a(\boldsymbol{\mu}) = (\boldsymbol{\Phi}_{\mathcal{I}}^a)^{-1}\mathbf{a}_{\mathcal{I}}(\boldsymbol{\mu}),$$

where $\mathbf{a}(\boldsymbol{\mu}) = \mathrm{vec}(\mathbf{A}(\boldsymbol{\mu}))$, while $\boldsymbol{\Phi}^a \in \mathbb{R}^{N_h^2\times M_a}$ is a basis for $\mathbf{a}(\boldsymbol{\mu})$. Similarly, we can employ DEIM to obtain an approximate affine decomposition for the right-hand side

$$\mathbf{g}(\boldsymbol{\mu}) \approx \mathbf{g}_m(\boldsymbol{\mu}) = \sum_{q=1}^{M_g} \theta_q^g(\boldsymbol{\mu})\, \mathbf{g}_q, \tag{3.31}$$

so that

$$\mathbf{g}_N(\boldsymbol{\mu}) \approx \mathbf{g}_N^m(\boldsymbol{\mu}) = \sum_{q=1}^{M_g} \theta_q^g(\boldsymbol{\mu})\, \mathbf{g}_N^q.$$

Therefore, the ROM with system approximation (or hyper-ROM) reads: find $\mathbf{y}_N^m \in \mathbb{R}^N$ such that

$$\mathbf{A}_N^m(\boldsymbol{\mu})\, \mathbf{y}_N^m = \mathbf{g}_N^m(\boldsymbol{\mu}). \tag{3.32}$$

Note that (3.32) can be obtained by Petrov-Galerkin projection using the reduced bases $\mathbf{W}$ and $\mathbf{V}$ of the following full-order model with system approximation

$$\mathbf{A}_m(\boldsymbol{\mu})\, \mathbf{y}_h^m = \mathbf{g}_m(\boldsymbol{\mu}). \tag{3.33}$$

Following the discussion in Sect. 3.3.5, we assume that the approximate matrix $\mathbf{A}_m(\boldsymbol{\mu})$ is nonsingular for all $\boldsymbol{\mu} \in \mathcal{D}$, that is

$$\beta_h^m(\boldsymbol{\mu}) = \sigma_{\min}(\mathbf{X}^{-\frac{1}{2}}\mathbf{A}_m(\boldsymbol{\mu})\mathbf{X}^{-\frac{1}{2}}) > 0 \qquad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{3.34}$$

Remark 3.5. Using the notation of Sect. 3.2.1, at the continuous level problem (3.26) reads: find $y_h \in X_h$ such that

$$a(y_h, v; \boldsymbol{\mu}) = g(v; \boldsymbol{\mu}) \qquad \forall v \in X_h, \tag{3.35}$$

where $X_h \subset X$, $\dim(X_h) = N_h$, $a(\cdot,\cdot;\boldsymbol{\mu})$ and $g(\cdot;\boldsymbol{\mu})$ are the bilinear and linear forms that give rise to the matrix $\mathbf{A}(\boldsymbol{\mu})$ and vector $\mathbf{g}(\boldsymbol{\mu})$. Problem (3.33) can thus be interpreted as a generalized Galerkin approximation: find $y_h^m \in X_h$ such that

$$a_m(y_h^m, v; \boldsymbol{\mu}) = g_m(v; \boldsymbol{\mu}) \qquad \forall v \in X_h, \tag{3.36}$$

where $a_m(\cdot,\cdot;\boldsymbol{\mu}) : X_h \times X_h \to \mathbb{R}$ and $g_m(\cdot;\boldsymbol{\mu}) : X_h \to \mathbb{R}$ are the forms associated to $\mathbf{A}_m(\boldsymbol{\mu})$ and $\mathbf{g}_m(\boldsymbol{\mu})$, respectively. Indeed, for all $w, v \in X_h$,

$$a_m(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{M_a} \theta_q^a(\boldsymbol{\mu})\, \mathbf{v}^T \mathbf{A}_q \mathbf{w}, \qquad g_m(v; \boldsymbol{\mu}) = \sum_{q=1}^{M_g} \theta_q^g(\boldsymbol{\mu})\, \mathbf{v}^T \mathbf{g}_q,$$

thanks to the bijection between the spaces X_h and $\mathbb{R}^{N_h}$ defined by the correspondence

$$v = \sum_{j=1}^{N_h} v_j \phi_j \in X_h \quad \leftrightarrow \quad \mathbf{v} = (v_1, \dots, v_{N_h})^T \in \mathbb{R}^{N_h}. \qquad \bullet$$

3.4.1 Generation of the reduced spaces

We still need to specify how to construct the reduced spaces, i.e. how to build the left and right bases $\mathbf{W}$ and $\mathbf{V}$. We first focus on the right basis $\mathbf{V}$, which is responsible for the approximation properties of the ROM. The greedy algorithm [PRV+02, VPRP03] and POD [Sir87] are probably the two most common approaches for the construction of $\mathbf{V}$. Both of them can be used in combination with DEIM and MDEIM techniques following a *system approximation then state-space reduction* approach, whose main steps are reported in Algorithm 3.4.

(1) compute a set of matrix and vector snapshots of (3.26);

(2) perform MDEIM and DEIM to obtain affine expansions of matrices and vectors;

(3a) run a greedy procedure to generate the reduced space, using (3.33) as high-fidelity model, or

(3b) generate a set of snapshots solution of (3.33), then use POD to generate the reduced space.

Algorithm 3.4 system approximation then state-space reduction approach for problem (3.26).

If the greedy algorithm is used, a suitable estimate for the norm of the error between the full and reduced-order solutions has to be provided. Moreover, the latter can also serve to quantify the solution accuracy in the online phase, and possibly guide a basis enrichment if the POD approach is employed. We postpone the derivation of a posteriori error estimates to Section 3.4.2.

We remark that, in the case when POD is used to build the reduced space, an alternative approach would be to generate the solution snapshots by solving (3.26) rather than (3.33). Since in this case steps (1) and (3b) would be run simultaneously, we refer to this approach as *simultaneous system approximation and state-space reduction*. We will concentrate on this paradigm in Sect. 3.6 when dealing with time-dependent problems.

The whole framework above is independent of the choice of the left basis $\mathbf{W}$. We only mention the two most popular options: (i) $\mathbf{W} = \mathbf{V}$, corresponding to a Galerkin projection, which is known to be optimal for symmetric positive definite problems; (ii) $\mathbf{W} = \mathbf{X}^{-1}\mathbf{A}_m(\boldsymbol{\mu})\mathbf{V}$, corresponding to a least-squares projection (also called minimum-residual method), which is suitable also for nonsymmetric, indefinite problems. In the

latter case, the matrix $\mathbf{X}$ is the one already introduced in (3.27) and results from the minimization of the dual norm of the residual $\|\mathbf{A}_m(\boldsymbol{\mu})\mathbf{V}\mathbf{u}_N^m - \mathbf{g}_m(\boldsymbol{\mu})\|^2_{\mathbf{X}^{-1}}$.

3.4.2 A posteriori error estimates

The goal of this section is to derive an a posteriori estimate for the norm of the error $\mathbf{e}_N^m(\boldsymbol{\mu}) = \mathbf{y}_h(\boldsymbol{\mu}) - \mathbf{V}\mathbf{y}_N^m(\boldsymbol{\mu})$ between the high-fidelity solution (3.26) and the reduced-order solution (3.32). To this end, let us split the error into a contribution $\mathbf{e}_m(\boldsymbol{\mu})$ due to *system approximation*,

$$\mathbf{e}_m(\boldsymbol{\mu}) = \mathbf{y}_h(\boldsymbol{\mu}) - \mathbf{y}_h^m(\boldsymbol{\mu}), \tag{3.37}$$

and a contribution $\mathbf{e}_N(\boldsymbol{\mu})$ due to *state-space reduction*,

$$\mathbf{e}_N(\boldsymbol{\mu}) = \mathbf{y}_h^m(\boldsymbol{\mu}) - \mathbf{V}\mathbf{y}_N^m(\boldsymbol{\mu}). \tag{3.38}$$

Since we want to find an estimate for the $\mathbf{X}$-norm of the error, it is useful to first define the $\mathbf{X}^{-1}$ vectorial norm

$$\|\mathbf{v}\|_{\mathbf{X}^{-1}} = \sqrt{(\mathbf{v}, \mathbf{X}^{-1}\mathbf{v})_2} = \|\mathbf{X}^{-\frac{1}{2}}\mathbf{v}\|_2 \qquad \forall \mathbf{v} \in \mathbb{R}^n,$$

and the associated $(\mathbf{X}, \mathbf{X}^{-1})$ matrix norm

$$\|\mathbf{B}\|_{\mathbf{X},\mathbf{X}^{-1}} = \sup_{\mathbf{v}\in\mathbb{R}^n} \frac{\|\mathbf{B}\mathbf{v}\|_{\mathbf{X}^{-1}}}{\|\mathbf{v}\|_{\mathbf{X}}} = \sup_{\mathbf{v}\in\mathbb{R}^n} \frac{\|\mathbf{X}^{-\frac{1}{2}}\mathbf{B}\mathbf{X}^{-\frac{1}{2}}\mathbf{v}\|_2}{\|\mathbf{v}\|_2} \qquad \forall\, \mathbf{B} \in \mathbb{R}^{n\times n}.$$

Note that the $\|\cdot\|_{\mathbf{X},\mathbf{X}^{-1}}$ norm is a consistent matrix norm, i.e.

$$\|\mathbf{B}\mathbf{v}\|_{\mathbf{X}^{-1}} \leq \|\mathbf{B}\|_{\mathbf{X},\mathbf{X}^{-1}}\|\mathbf{v}\|_{\mathbf{X}} \qquad \forall \mathbf{B} \in \mathbb{R}^{n\times n}, \mathbf{v} \in \mathbb{R}^n.$$

A first error bound can be obtained by estimating separately the two error components and then using the triangular inequality.

Proposition 3.1. *If $M_a \in \mathbb{N}^+$ and $\{\theta_q(\boldsymbol{\mu})\}_{q=1}^{M_a}$ are such that the matrix $\mathbf{A}_m(\boldsymbol{\mu})$ is nonsingular for all $\boldsymbol{\mu} \in \mathcal{D}$, then the norm of the error $\mathbf{e}_N^m(\boldsymbol{\mu})$ can be bounded by*

$$\begin{aligned}\|\mathbf{y}_h(\boldsymbol{\mu}) - \mathbf{V}\mathbf{y}_N^m(\boldsymbol{\mu})\|_{\mathbf{X}} \leq{}& \frac{1}{\beta_h^m(\boldsymbol{\mu})}\|\mathbf{A}_m(\boldsymbol{\mu})\mathbf{V}\mathbf{y}_N^m - \mathbf{g}_m(\boldsymbol{\mu})\|_{\mathbf{X}^{-1}} \\ &+ \frac{1}{\beta_h(\boldsymbol{\mu})}\left(\|\mathbf{g}(\boldsymbol{\mu}) - \mathbf{g}_m(\boldsymbol{\mu})\|_{\mathbf{X}^{-1}} + \|\mathbf{A}(\boldsymbol{\mu}) - \mathbf{A}_m(\boldsymbol{\mu})\|_{\mathbf{X},\mathbf{X}^{-1}}\|\mathbf{y}_h^m(\boldsymbol{\mu})\|_{\mathbf{X}}\right). \end{aligned} \tag{3.39}$$

Proof. We first derive an estimate for the error $\mathbf{e}_m(\boldsymbol{\mu})$. From (3.26) and (3.33), we have that

$$\mathbf{A}(\boldsymbol{\mu})\mathbf{y}_h - \mathbf{A}(\boldsymbol{\mu})\mathbf{y}_h^m + \mathbf{A}(\boldsymbol{\mu})\mathbf{y}_h^m - \mathbf{A}_m(\boldsymbol{\mu})\mathbf{y}_h^m = \mathbf{g}(\boldsymbol{\mu}) - \mathbf{g}_m(\boldsymbol{\mu}).$$

Rearranging the terms we obtain

$$\mathbf{A}(\boldsymbol{\mu})(\mathbf{y}_h - \mathbf{y}_h^m) = \mathbf{g}(\boldsymbol{\mu}) - \mathbf{g}_m(\boldsymbol{\mu}) + (\mathbf{A}_m(\boldsymbol{\mu}) - \mathbf{A}(\boldsymbol{\mu}))\mathbf{y}_h^m,$$

so that

$$\mathbf{e}_m(\boldsymbol{\mu}) = \mathbf{A}^{-1}(\boldsymbol{\mu})\left(\mathbf{g}(\boldsymbol{\mu}) - \mathbf{g}_m(\boldsymbol{\mu}) + (\mathbf{A}_m(\boldsymbol{\mu}) - \mathbf{A}(\boldsymbol{\mu}))\mathbf{y}_h^m\right).$$

Left multiplying by $\mathbf{X}^{\frac{1}{2}}$, exploiting the identity $\mathbf{X}^{\frac{1}{2}}\mathbf{X}^{-\frac{1}{2}} = \mathbf{I}$ at the right-hand side and taking the 2-norm we then obtain

$$\|\mathbf{e}_m(\boldsymbol{\mu})\|_{\mathbf{X}} \leq \|\mathbf{X}^{\frac{1}{2}}\mathbf{A}^{-1}(\boldsymbol{\mu})\mathbf{X}^{\frac{1}{2}}\|_2 \Big(\|\mathbf{g}(\boldsymbol{\mu}) - \mathbf{g}_m(\boldsymbol{\mu})\|_{\mathbf{X}^{-1}} + \|(\mathbf{A}_m(\boldsymbol{\mu}) - \mathbf{A}(\boldsymbol{\mu}))\mathbf{y}_h^m(\boldsymbol{\mu})\|_{\mathbf{X}^{-1}} \Big),$$

which yields the second term in (3.39). The first term is nothing but the usual residual-based a posteriori error estimate [RHP08] for $\mathbf{e}_N(\boldsymbol{\mu})$. Indeed, from (3.33) and (3.32), we have that

$$\mathbf{A}_m(\boldsymbol{\mu})\mathbf{e}_N(\boldsymbol{\mu}) = \mathbf{g}_m(\boldsymbol{\mu}) - \mathbf{A}_m(\boldsymbol{\mu})\mathbf{V}\mathbf{y}_N^m,$$

and therefore

$$\|\mathbf{e}_N(\boldsymbol{\mu})\|_{\mathbf{X}} \leq \|\mathbf{X}^{\frac{1}{2}}\mathbf{A}_m^{-1}(\boldsymbol{\mu})\mathbf{X}^{\frac{1}{2}}\|_2 \, \|\mathbf{A}_m(\boldsymbol{\mu})\mathbf{V}\mathbf{y}_N^m - \mathbf{g}_m(\boldsymbol{\mu})\|_{\mathbf{X}^{-1}},$$

which provides the first term in (3.39). □

Unfortunately, the error bound (3.39) is of little practical use, since it requires the computation of the full-order solution $\mathbf{y}_h^m(\boldsymbol{\mu})$ of (3.33). Nevertheless, we can also prove a similar estimate for the error $\mathbf{e}_N^m(\boldsymbol{\mu})$ which does not involve the full-order solution $\mathbf{y}_h^m(\boldsymbol{\mu})$.

Proposition 3.2. *Under the assumptions of Proposition 3.1, the following estimate holds:*

$$\|\mathbf{y}_h(\boldsymbol{\mu}) - \mathbf{V}\mathbf{y}_N^m(\boldsymbol{\mu})\|_{\mathbf{X}} \leq \Delta_N(\boldsymbol{\mu}) + \Delta_m(\boldsymbol{\mu}), \tag{3.40}$$

where

$$\Delta_N(\boldsymbol{\mu}) = \frac{1}{\beta_h(\boldsymbol{\mu})}\|\mathbf{A}_m(\boldsymbol{\mu})\mathbf{V}\mathbf{y}_N^m(\boldsymbol{\mu}) - \mathbf{g}_m(\boldsymbol{\mu})\|_{\mathbf{X}^{-1}}, \tag{3.41}$$

and

$$\Delta_m(\boldsymbol{\mu}) = \frac{1}{\beta_h(\boldsymbol{\mu})}\Big(\|\mathbf{g}(\boldsymbol{\mu}) - \mathbf{g}_m(\boldsymbol{\mu})\|_{\mathbf{X}^{-1}} + \|\mathbf{A}(\boldsymbol{\mu}) - \mathbf{A}_m(\boldsymbol{\mu})\|_{\mathbf{X},\mathbf{X}^{-1}}\|\mathbf{V}\mathbf{y}_N^m(\boldsymbol{\mu})\|_{\mathbf{X}} \Big). \tag{3.42}$$

Proof. From the problem statement (3.26) we have that

$$\begin{aligned}\mathbf{A}(\boldsymbol{\mu})\mathbf{y}_h - \mathbf{A}(\boldsymbol{\mu})\mathbf{V}\mathbf{y}_N^m &= \mathbf{g}(\boldsymbol{\mu}) - \mathbf{A}(\boldsymbol{\mu})\mathbf{V}\mathbf{y}_N^m \\ &= \mathbf{g}(\boldsymbol{\mu}) - \mathbf{g}_m(\boldsymbol{\mu}) + \mathbf{g}_m(\boldsymbol{\mu}) - \mathbf{A}_m(\boldsymbol{\mu})\mathbf{V}\mathbf{y}_N^m + \mathbf{A}_m(\boldsymbol{\mu})\mathbf{V}\mathbf{y}_N^m - \mathbf{A}(\boldsymbol{\mu})\mathbf{V}\mathbf{y}_N^m,\end{aligned}$$

that is

$$\mathbf{e}_N^m(\boldsymbol{\mu}) = \mathbf{A}^{-1}(\boldsymbol{\mu})\Big(\mathbf{g}(\boldsymbol{\mu}) - \mathbf{g}_m(\boldsymbol{\mu}) + (\mathbf{A}_m(\boldsymbol{\mu}) - \mathbf{A}(\boldsymbol{\mu}))\mathbf{V}\mathbf{y}_N^m + \mathbf{g}_m(\boldsymbol{\mu}) - \mathbf{A}_m(\boldsymbol{\mu})\mathbf{V}\mathbf{y}_N^m \Big).$$

We readily obtain the desired estimate by proceeding as in the proof of Proposition 3.1. □

While the first term in (3.40) – involving the dual norm of the residual – admits an efficient offline-online decomposition (see e.g. [RHP08]), the second term – taking into account the system approximation – still depends on the full-order original and approximated matrices and right-hand sides. However, combining the DEIM error bounds (3.14) and (3.24) we obtain an N_h-independent approximation for $\Delta_m(\boldsymbol{\mu})$,

$$\Delta_m(\boldsymbol{\mu}) \approx \frac{1}{\beta(\boldsymbol{\mu})}\Big(c_1\|(\boldsymbol{\Phi}_{\mathcal{I}}^g)^{-1}\|_2 \, \sigma_{M_g+1}^g + c_2\|(\boldsymbol{\Phi}_{\mathcal{I}}^a)^{-1}\|_2 \, \sigma_{M_a+1}^a \, \|\mathbf{V}\mathbf{y}_N^m(\boldsymbol{\mu})\|_{\mathbf{X}} \Big), \tag{3.43}$$

where $c_1 = \|\mathbf{X}^{-1/2}\|_2$ and $c_2 = \|\mathbf{X}^{-1}\|_2$ are two norm equivalence constants and $\sigma^g_{M_g+1}$ and $\sigma^a_{M_a+1}$ denote the first truncated singular values in the MDEIM approximations of $\mathbf{g}$ and $\mathbf{A}$.

Remark 3.6. The expression for $\Delta_m(\boldsymbol{\mu})$ shows that the error $\mathbf{e}^m_N(\boldsymbol{\mu})$ is controlled by the difference in the action of the operators $\mathbf{A}(\boldsymbol{\mu})$ and $\mathbf{A}_m(\boldsymbol{\mu})$ onto the subspace Range($\mathbf{V}$). This may suggest that, in case of simultaneous system approximation and state-space reduction, the training of MDEIM should not try to match $\mathbf{A}(\boldsymbol{\mu})$ to $\mathbf{A}_m(\boldsymbol{\mu})$, but rather $\mathbf{A}(\boldsymbol{\mu})\mathbf{V}$ to $\mathbf{A}_m(\boldsymbol{\mu})\mathbf{V}$. As a result, instead of approximating a sparse vector function of dimension N_h^2, we would approximate a dense vector function of dimension $N_h N$. Depending on the sparsity and the ratio N_h/N, one of the two approaches may be more efficient. More importantly however, this approach could only be implemented in combination with a simultaneous system approximation and state-space reduction approach. For this reason, we do not further investigate here this alternative option. •

3.5 Application to the shape optimization of an acoustic horn

We now apply the reduction approach developed in the previous section to problem (3.3). The full-order model is given by a $\mathbb{P}_1$ finite element approximation of (3.4) (as described in Sect. 3.2.1), leading to a linear system (3.7) of dimension $N_h = 48\,925$, obtained using a mesh made of 96 537 triangular elements. Unless otherwise stated, the CPU times reported in this section refer to computations performed on a workstation with a Intel Core i5-2400S CPU and 16 GB of RAM.

3.5.1 One parameter case (frequency)

As a first test we keep the geometrical parameters $\boldsymbol{\mu}_g$ fixed to the reference configuration and let the frequency $f = \mu_1$ vary in the range $\mathcal{D} = [10, 1800]$. Since the shape parametrization is not considered here, the problem exhibits a trivial affine decomposition that we expect to recover exactly within our framework. In fact, the interpolation procedure terminates after selecting $M_a = 3$ and $M_g = 1$ bases (out of 20 snapshots in both cases) for the system matrix and right-hand side. In this case, system approximation does not introduce any error in the ROM, so that the two procedures to combine system and state-space reduction coincide. Moreover, the evaluation of the $\theta(\boldsymbol{\mu})$ functions is extremely fast, since only 24 (out of 96 537) reduced elements have been selected (see also Table 3.1).

The next step consists in constructing the reduced basis $\mathbf{V}$. We plug the obtained empirical affine decomposition into the usual (Galerkin) RB framework. We first compute an approximation of the stability factor $\beta_h(\boldsymbol{\mu})$ by means of the adaptive interpolation strategy presented in Sect. 2.5.2, see Fig. 3.4. Then, we run the greedy algorithm using $\Delta_N(\boldsymbol{\mu})$ as an error estimator; by requiring a relative tolerance of 10^{-4}, we end up with a reduced basis of dimension $N = 50$. In Fig. 3.4 we also report the stability factor $\beta^m_N(\boldsymbol{\mu})$ of the reduced problem, defined as

$$\beta^m_N(\boldsymbol{\mu}) = \sigma_{\min}(\mathbf{X}_N^{-\frac{1}{2}} \mathbf{A}^m_N(\boldsymbol{\mu}) \mathbf{X}_N^{-\frac{1}{2}}), \tag{3.44}$$

where $\mathbf{X}_N = \mathbf{V}^T\mathbf{X}\mathbf{V}$. We observe that $\beta_N^m(\boldsymbol{\mu}) \geq \beta_h(\boldsymbol{\mu}) > 0$ over the entire parameter domain, thus numerically showing the stability of the Galerkin projection. Moreover, in Fig. 3.4 we compare the error estimate $\Delta_N(\boldsymbol{\mu}) + \Delta_m(\boldsymbol{\mu})$ with the norm of the error $\mathbf{e}_N^m(\boldsymbol{\mu})$. The estimate is sharp and correctly predicts the error convergence; as expected, the contribute due to system approximation (Δ_m) is negligible with respect to the one due to state-space reduction (Δ_N).

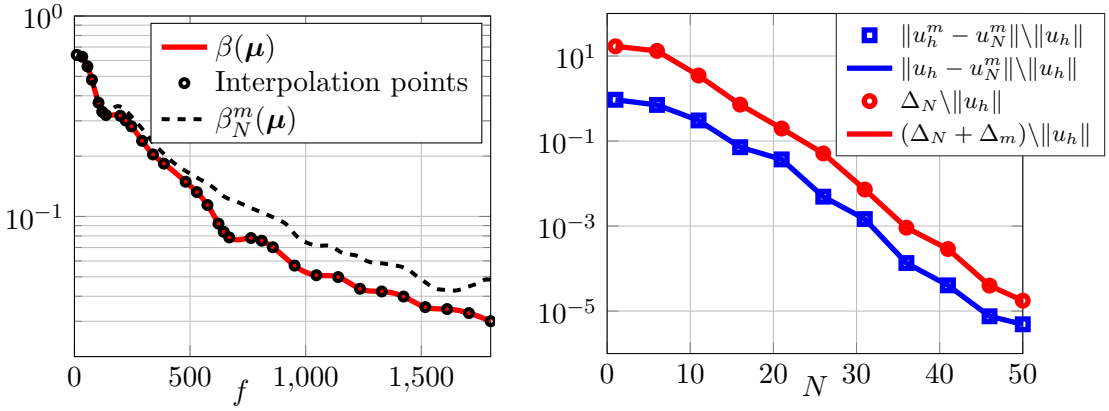


Fig. 3.4 Acoustic horn, one parameter test case. On the left: stability factor $\beta_h(\mu)$ (and its reduced counterpart $\beta_N^m(\boldsymbol{\mu})$). On the right: relative error and estimate with respect to N (average values over a random sample of 200 parameter points).

Table 3.1 Acoustic horn, one parameter test case. Computational details.

Approximation data		Computational performances	
Number of FE dofs N_h	48 925	Number of ROM dofs	50
Number of elements	96 537	Number of reduced elements	24
Number of parameters P	1	Dofs reduction	978:1
Number of matrix snapshots	20	Number of matrix bases M_a	3
Number of rhs snapshots	20	Number of rhs bases M_g	1
FOM solution time	1.5 s	ROM solution time	$5 \cdot 10^{-4}$ s
Tolerance RB greedy ε_{tol}	10^{-4}	ROM online estimation	$2 \cdot 10^{-3}$ s

3.5.2 Two parameters case (frequency plus one RBF control point)

In addition to the frequency, we then also consider a geometrical parameter, namely the vertical displacement of the right-most RBF control point in Fig. 3.1. The parameter domain is then given by $\mathcal{D} = [50, 1000] \times [-0.03, 0.03]$.

We begin by computing a set of 80 matrix and vector snapshots corresponding to 80 parameter samples selected by latin hypercube (LHS) sampling design [Coc07, Loh10] in $\mathcal{D}$. The eigenvalues of the correlation matrices of matrix and vector snapshots are reported in Fig 3.5. Based on the decay of the singular values, we retain the first $M_a = 13$ and $M_g = 3$ POD modes and then perform MDEIM and DEIM, respectively.

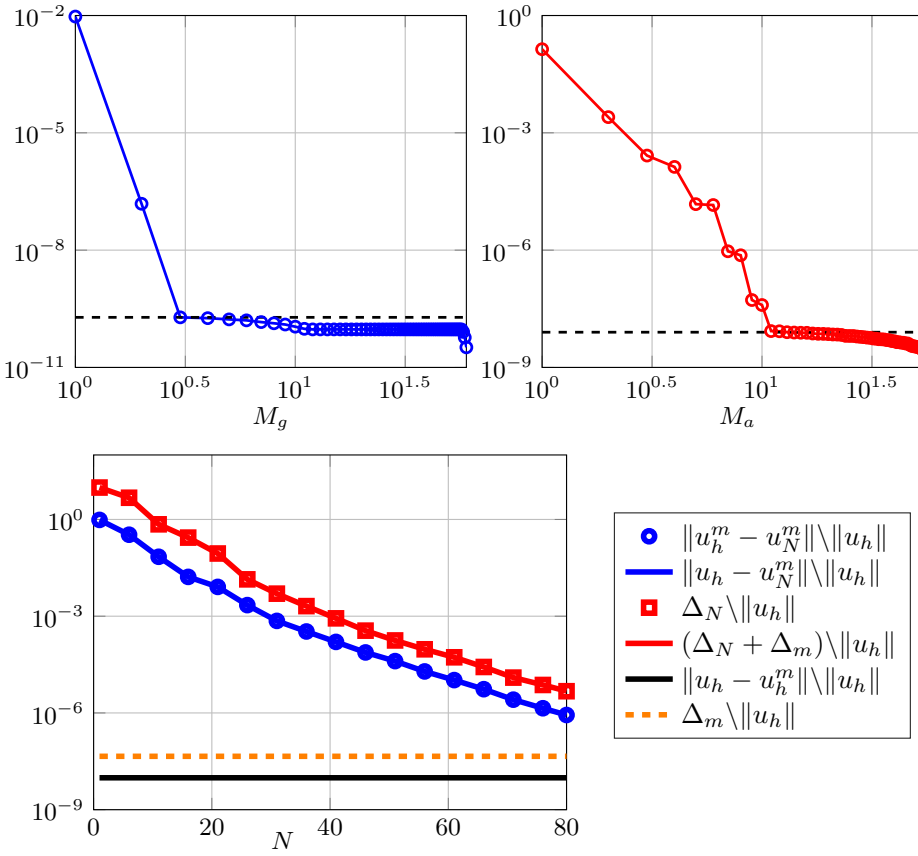


Fig. 3.5 Acoustic horn, two parameters test case. On the left: POD spectrum of vector (blue) and matrix (red) snapshots. On the right: relative error over a testing set of 200 points. $M_a = 13$, $M_f = 3$.

In this case, rather than employing the greedy algorithm, we first solve offline the approximated FOM (3.33) to obtain a set of 150 solution snapshots (corresponding to 150 parameter configurations selected by LHS design) and then we extract $N = 80$ POD basis functions. Finally, we compute all the quantities required to compute the estimate $\Delta_N(\boldsymbol{\mu})$ in the online stage. To verify the ROM accuracy, in Fig. 3.5 we report the average errors and estimates over a testing set of 200 samples; on average, the system approximation error $\|\mathbf{e}_m(\boldsymbol{\mu})\|_{\mathbf{X}}$ is roughly two orders of magnitude less than the reduction error $\|\mathbf{e}_N(\boldsymbol{\mu})\|_{\mathbf{X}}$, so that the effect of system approximation is negligible in the reduced model for $M_a = 13$ and $M_g = 3$. To measure the influence of the system approximation on the reduced model, we compute then the error $\|\mathbf{e}_N^m\|_{\mathbf{X}}$ for different levels of matrix approximation (keeping $M_g = 3$ fixed). The results are reported in Fig. 3.6; we observe that already with $M_a = 9$, we obtain a sufficiently accurate reduced model, the relative error being far below 1%.

3.5.3 Five parameters case

We now let all five parameters $[f\ \boldsymbol{\mu}_g]$ vary: the parameter domain is given by $\mathcal{D} = [50, 1000] \times \mathcal{D}_g$, where $\mathcal{D}_g = [-0.03, 0.03]^4$. As before, we first perform MDEIM and

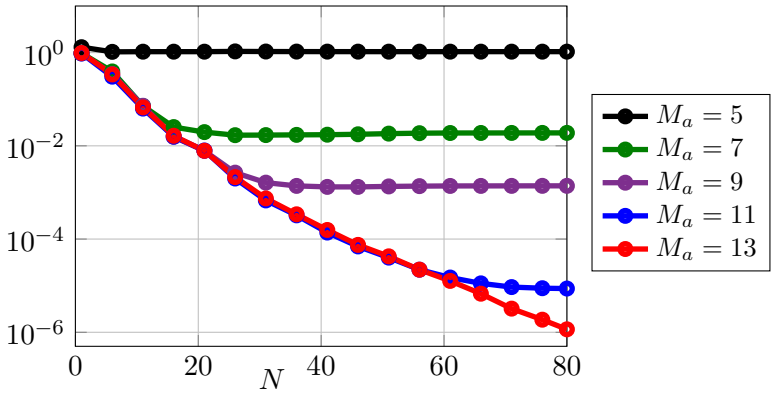


Fig. 3.6 Acoustic horn, two parameters test case. Average (over a testing sample of 200 random points) error $\|\mathbf{e}_N^m(\boldsymbol{\mu})\|_{\mathbf{X}}$ with respect to N for different value of M_a (with $M_g = 3$ fixed).

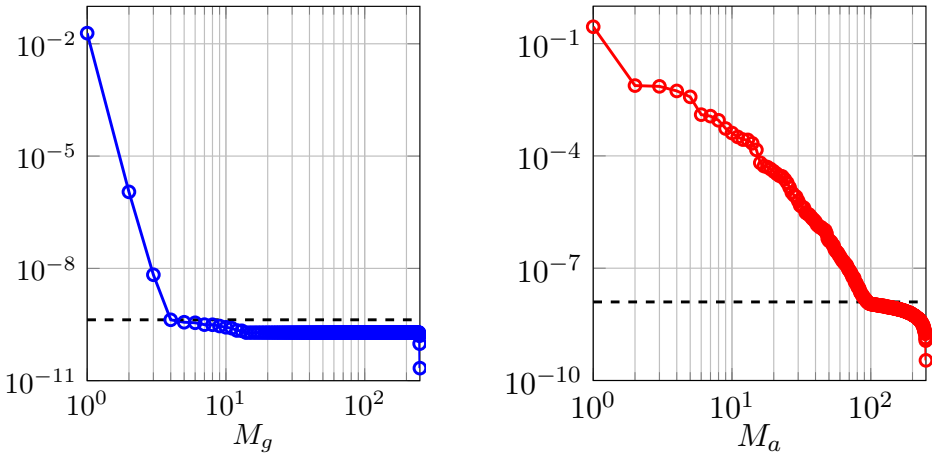


Fig. 3.7 Acoustic horn, five parameters test case. POD spectrum of vector (blue) and matrix (red) snapshots.

DEIM using matrix and vector bases made of $M_a = 95$ and $M_g = 4$ POD modes, extracted from a set of 250 snapshots (the corresponding spectra are reported in Fig. 3.7, while the reduced mesh is shown in Fig 3.8). Then, we employ again POD to build a basis $\mathbf{V}$ of dimension $N = 80$ starting from a random set of 200 solution snapshots. The entire offline time for the construction of the hyper-ROM (including system approximation, reduced space construction, projection and computation of the ingredients for error estimation) is about 11 minutes[1]. In particular, running MDEIM on the matrix snapshots takes only 16 seconds (9 seconds for extracting the POD basis and 7 seconds for selecting the interpolation indices), thus representing a very marginal cost.

In this case the dominating error is the one due to system approximation, as shown in Fig. 3.8, which is however less than 0.5% on average over the parameter space. As a result, we obtain a reliable approximation of the output of interest $J(\boldsymbol{\mu})$ (see Fig. 3.9) whose evaluation is one hundred times faster than that of the original high-fidelity approximation. Further details about the computational performances are provided in

[1]In this case, for offline computations we used 8 cores on a node (equipped with two Intel Xeon E5-2660 processors and 64 GB of RAM) of the SuperB cluster at EPFL.

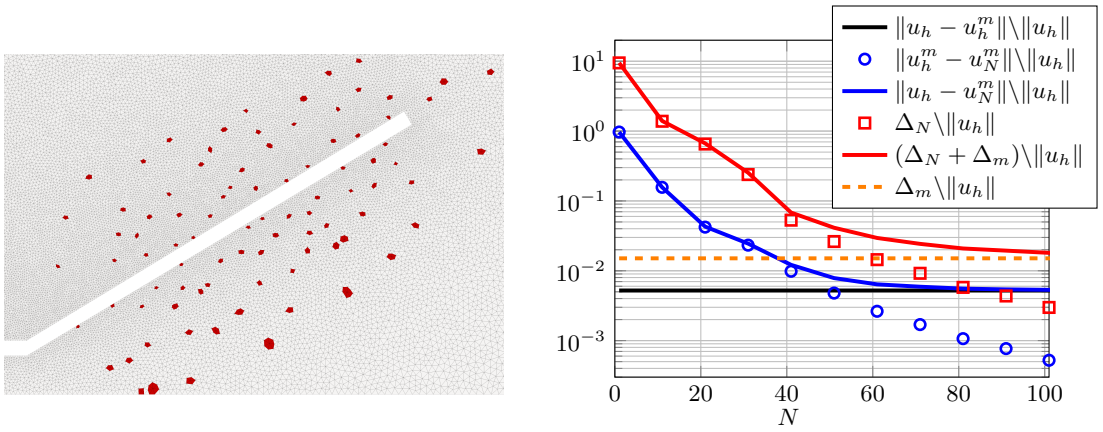


Fig. 3.8 Acoustic horn, five parameters test case. On the left: zoom of the reduced mesh (red elements) around the horn. On the right: relative error and estimates averaged over a testing set of 200 points.

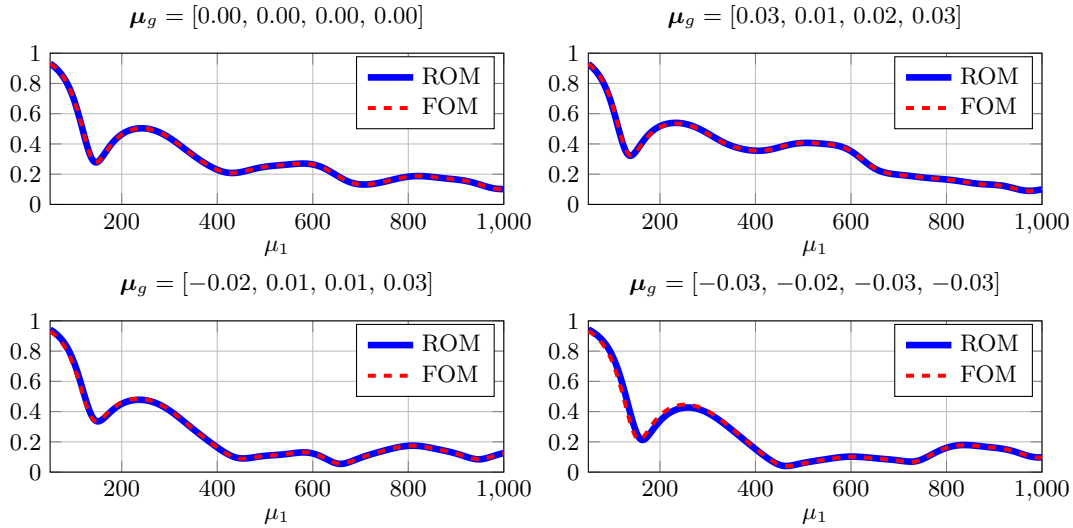


Fig. 3.9 Acoustic horn, five parameters test case: comparison of the reflection spectrum obtained by the FOM and ROM for different shapes.

Table 3.2.

Therefore, we can exploit the ROM to efficiently solve the problem of finding the shape which maximizes the horn efficiency over a certain frequency range. This leads to the following shape optimization problem (see e.g. [BNB03, KWB12, UB08]): find $\boldsymbol{\mu}_g^* \in \mathcal{D}_g$ such that

$$\boldsymbol{\mu}_g^* = \arg\min_{\boldsymbol{\mu}_g \in \mathcal{D}_g} R(f, \boldsymbol{\mu}_g) := \arg\min_{\boldsymbol{\mu}_g \in \mathcal{D}_g} \sum_{f \in \mathcal{F}} \left(J(f, \boldsymbol{\mu}_g)\right)^2, \tag{3.45}$$

where $\mathcal{F}$ is a set of frequencies at which we want to minimize the waves reflection. We solve the PDE-constrained least-squares minimization problem (3.45) by means of a black-box SQP optimization routine (MATLAB `fmincon`) with finite-difference approximation of the gradient and BFGS approximation of the Hessian of the objective

Table 3.2 Acoustic horn, five parameters test case. Computational details.

Approximation data		Computational performances	
Number of FE dofs N_h	48 925	Number of ROM dofs	100
Number of elements	96 537	Number of reduced elements	540
Number of parameters P	5	Dofs reduction	490:1
Number of matrix snapshots	250	Number of matrix bases M_a	95
Number of rhs snapshots	250	Number of rhs bases M_g	4
Solution snapshots	200	ROM solution time	$2 \cdot 10^{-2}$ s

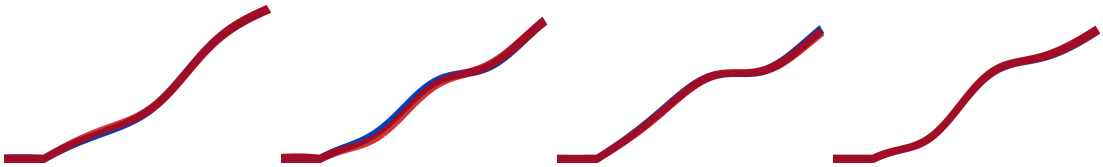


Fig. 3.10 Acoustic horn, five parameters test case. Comparison between the shapes of the horn resulting from different type of optimization using the ROM (red) and the FOM (blue). From left to right: optimization at $f = 300$ Hz, $f = 600$ Hz, $f = 1000$ Hz, and finally optimization over the frequency range $700 - 1000$ Hz.

function $R(\cdot, \cdot)$. As a result, at each optimization iteration the routine requires to solve the PDE at least $5|\mathcal{F}|$ times. We first perform the optimization for a list of given frequencies $\mathcal{F} = 300, 600, 1000$ Hz: finding the optimal $\boldsymbol{\mu}_g$ requires from 25 to 200 ROM evaluations and thus takes no more than 4 seconds (the corresponding geometries and reflection spectra are reported in Figs. 3.10 and 3.11). As a comparison, relying on the FOM, the optimization would require about 3 minutes to achieve convergence. In all cases, the optimal shapes returned by the MDEIM-based procedure matches very well the ones computed using the FOM. The speedup that is achieved is even more evident when we are interested in optimizing the horn efficiency over a certain frequency range, leading to the definition of a robust optimization problem. Considering the solution of (3.45) with 300 frequencies $\mathcal{F} = \{700 + i\}_{i=1}^{300}$ requires about $2.3 \cdot 10^4$ PDE solutions, which takes almost 9 hours using the FOM, but only 8 minutes using the ROM. Again, one can observe in Fig. 3.10 that both optimal shapes match very well. Moreover, even taking into account the offline time for the ROM generation, a speedup of 28 is achieved.

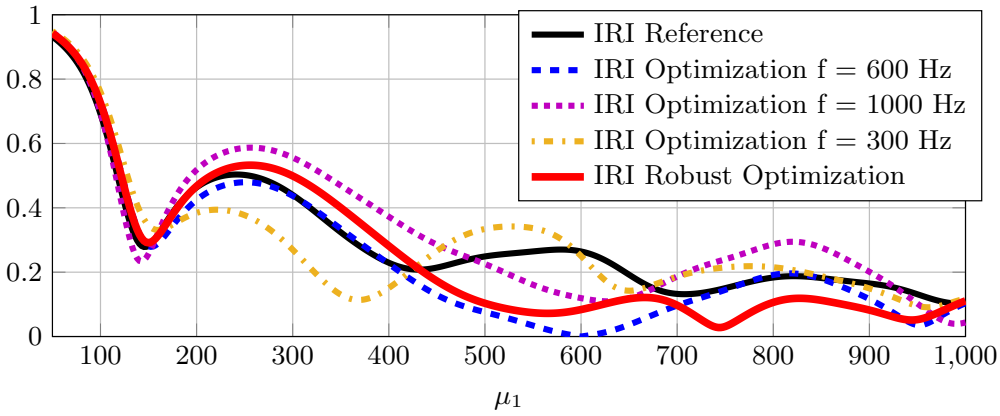


Fig. 3.11 Acoustic horn, five parameters test case. Reflection spectra for the horns in Fig. 3.10 optimized using the ROM.

3.6 Hyper-reduction of parametrized linear parabolic equations

We now consider as full-order model the linear dynamical system (1.11) introduced in Sect. 1.2.2: given $\boldsymbol{\mu} \in \mathcal{D}$, for all $t \in (0, T)$, find $\mathbf{y}_h(t; \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ such that

$$\mathbf{M}(t; \boldsymbol{\mu}) \frac{d\mathbf{y}_h}{dt} + \mathbf{A}(t; \boldsymbol{\mu}) \mathbf{y}_h = \mathbf{g}(t; \boldsymbol{\mu}), \tag{3.46}$$

with $\mathbf{y}_h(0; \boldsymbol{\mu}) = \mathbf{0}$. For the sake of simplicity, we consider here a null initial condition. However, everything still applies in the more general case where $\mathbf{y}_h(0; \boldsymbol{\mu}) = \mathbf{y}_0(\boldsymbol{\mu})$, see Remark 3.7.

Applying MDEIM to approximate $\mathbf{A}(t; \boldsymbol{\mu})$ and $\mathbf{M}(t; \boldsymbol{\mu})$, and DEIM to approximate $\mathbf{g}(t; \boldsymbol{\mu})$ as in Sect. 3.4, leads to the following full-order model with system approximation: find $\mathbf{y}_{h,m}(t; \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ such that

$$\mathbf{M}_m(t; \boldsymbol{\mu}) \frac{d\mathbf{y}_{h,m}}{dt} + \mathbf{A}_m(t; \boldsymbol{\mu}) \mathbf{y}_{h,m} = \mathbf{g}_m(t; \boldsymbol{\mu}). \tag{3.47}$$

Following (3.1), Petrov-Galerkin projection leads to a reduced system of equations in terms of a reduced variable $\mathbf{y}_{N,m}(t; \boldsymbol{\mu}) \in \mathbb{R}^N$

$$\mathbf{M}_N^m(t; \boldsymbol{\mu}) \frac{d\mathbf{y}_{N,m}}{dt} + \mathbf{A}_N^m(t; \boldsymbol{\mu}) \mathbf{y}_{N,m} = \mathbf{g}_N^m(t; \boldsymbol{\mu}), \tag{3.48}$$

where

$$\mathbf{M}_N^m(t; \boldsymbol{\mu}) = \mathbf{W}^T \mathbf{M}_m(t; \boldsymbol{\mu}) \mathbf{V}, \quad \mathbf{A}_N^m(t; \boldsymbol{\mu}) = \mathbf{W}^T \mathbf{A}_m(t; \boldsymbol{\mu}) \mathbf{V}, \quad \mathbf{g}_N^m(t; \boldsymbol{\mu}) = \mathbf{W}^T \mathbf{g}_m(t; \boldsymbol{\mu}).$$

Remark 3.7. If $\mathbf{y}_h(0; \boldsymbol{\mu}) = \mathbf{y}_0(\boldsymbol{\mu})$, then we seek an approximation of $\mathbf{y}_h(t; \boldsymbol{\mu})$ in the affine subspace $\mathbf{y}_0(\boldsymbol{\mu}) + \text{span}(\mathbf{V})$, i.e.

$$\mathbf{y}_h(t; \boldsymbol{\mu}) \approx \mathbf{y}_0(\boldsymbol{\mu}) + \mathbf{V} \mathbf{y}_N(t; \boldsymbol{\mu}).$$

Therefore, the hyper-ROM consists in finding $\mathbf{y}_{N,m} = \mathbf{y}_{N,m}(t;\boldsymbol{\mu}) \in \mathbb{R}^N$ such that

$$\mathbf{M}_N^m(t;\boldsymbol{\mu})\frac{d\mathbf{y}_{N,m}}{dt} + \mathbf{A}_N^m(t;\boldsymbol{\mu})\mathbf{y}_{N,m} = \mathbf{g}_N^m(t;\boldsymbol{\mu}) - \mathbf{V}^T\mathbf{A}_m(t;\boldsymbol{\mu})\mathbf{y}_0(\boldsymbol{\mu}),$$

with $\mathbf{y}_{N,m}(0;\boldsymbol{\mu}) = \mathbf{0}$. •

3.6.1 Generation of the reduced spaces

The two approaches for the construction of the reduced spaces defined in Sect. 3.4.1 can also be applied in the present case. Here, however, we concentrate on a suitable *simultaneous system approximation and state-space reduction* approach where a global reduced basis $\mathbf{V}$ is constructed by means of POD. As the snapshots collection is performed at the fully-discrete level, we start by introducing a time discretization of problem (3.46) by means of the Backward Differentiation Formulas (BDF), a family of implicit linear multistep methods, see e.g. [BCP96, QSS07].

To this end, we first partition the time interval $[0,T]$ into N_t subintervals of equal size $\Delta t = T/N_t$ and denote by $t_n = n\Delta t$, for $n = 0,\dots,N_t$ the discrete time instances. Moreover, we denote by $\mathbf{y}_h^n$ the approximation of $\mathbf{y}_h$ at time t_n. Then, we approximate the time derivative as[2]

$$\frac{\partial \mathbf{y}_h}{\partial t} \approx \frac{\alpha\mathbf{y}_h^{n+1} - \mathbf{y}_h^{n,\sigma}}{\Delta t},$$

where (limiting ourselves to BDF schemes of order $\sigma = 1, 2$)

$$\mathbf{y}_h^{n,\sigma} = \begin{cases} \mathbf{y}_h^n, & \text{if } n \geq 0, \quad \text{for } \sigma = 1 \text{ (BDF1)}, \\ 2\mathbf{y}_h^n - \frac{1}{2}\mathbf{y}_h^{n-1}, & \text{if } n \geq 1, \quad \text{for } \sigma = 2 \text{ (BDF2)}, \end{cases} \tag{3.49}$$

and

$$\alpha = \begin{cases} 1, & \text{for } \sigma = 1 \text{ (BDF1)}, \\ \frac{3}{2}, & \text{for } \sigma = 2 \text{ (BDF2)}. \end{cases} \tag{3.50}$$

The fully-discrete FOM reads: given $\mathbf{y}_h^n(\boldsymbol{\mu}),\dots,\mathbf{y}_h^{n+1-\sigma}(\boldsymbol{\mu})$, for $n \geq \sigma - 1$, find $\mathbf{y}_h^{n+1}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ such that

$$\mathbf{M}(t_{n+1};\boldsymbol{\mu})\frac{\alpha\mathbf{y}_h^{n+1} - \mathbf{y}_h^{n,\sigma}}{\Delta t} + \mathbf{A}(t_{n+1};\boldsymbol{\mu})\mathbf{y}_h^{n+1} = \mathbf{g}(t_{n+1};\boldsymbol{\mu}). \tag{3.51}$$

The construction of the reduced model consists of two main steps, see Algorithm 3.5 for the details. (In view of the application of Section 3.9, we only consider here the case

[2]Given an implicit differential algebraic equation of the form

$$F(t,y,y') = 0,$$

the σ-step BDF method consists of replacing y' by the derivative of the Lagrange polynomial which interpolates the computed solution at times $t_{n+1},\dots,t_{n-\sigma+1}$, evaluated at t_{n+1}. This yields

$$F\left(t_{n+1}, y^{n+1}, \sum_{i=0}^{\sigma}\frac{\gamma_i}{\Delta t}y^{n+1-i}\right) = 0,$$

where γ_i, $i = 0,\dots,\sigma$ are the coefficient of the BDF method.

of Galerkin projection, i.e. we choose $\mathbf{W} = \mathbf{V}$.) First, we introduce a set of K training inputs $\{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^K\}$. For each $\boldsymbol{\mu}^k$, $k = 1, \dots, K$, we solve (3.46), collect snapshots

$$\{\mathbf{y}_h^n(\boldsymbol{\mu}^k)\}_{n=1}^{N_t}, \qquad \{\mathbf{g}(t_n; \boldsymbol{\mu}^k)\}_{n=1}^{N_t}, \qquad \{\mathbf{A}(t_n; \boldsymbol{\mu}^k)\}_{n=1}^{N_t}, \qquad \{\mathbf{M}(t_n; \boldsymbol{\mu}^k)\}_{n=1}^{N_t},$$

compress them independently by POD and progressively build respective global bases $\mathbf{V}$, $\boldsymbol{\Phi}^g$, $\boldsymbol{\Phi}^a$, $\boldsymbol{\Phi}^m$. This progressive construction of POD bases can be efficiently done using the procedure described in [PDTA14]. Then, we perform DEIM on $\boldsymbol{\Phi}^g$ and MDEIM on $\boldsymbol{\Phi}^a$, $\boldsymbol{\Phi}^m$ to obtain affine approximations of $\mathbf{g}(t; \boldsymbol{\mu})$, $\mathbf{A}(t; \boldsymbol{\mu})$ and $\mathbf{M}(t; \boldsymbol{\mu})$, respectively. Finally, the resulting matrices and vectors are projected onto the reduced basis $\mathbf{V}$.

Set $\mathbf{V} = [\,]$, $\boldsymbol{\Phi}^a = [\,]$, $\boldsymbol{\Phi}^m = [\,]$, $\boldsymbol{\Phi}^g = [\,]$
(1) collect and compress solution, matrix and vector snapshots
    for $k = 1 : K$
        (1a) solve (3.51) for $\boldsymbol{\mu} = \boldsymbol{\mu}^k$ to obtain solution and system snapshots
            $\boldsymbol{\Lambda}_u^k = [\mathbf{y}_h^1(\boldsymbol{\mu}^k), \dots, \mathbf{y}_h^{N_t}(\boldsymbol{\mu}^k)]$
            $\boldsymbol{\Lambda}_a^k = [\mathrm{vec}(\mathbf{A}(t_1; \boldsymbol{\mu}^k)), \dots, \mathrm{vec}(\mathbf{A}(t_{N_t}; \boldsymbol{\mu}^k))]$
            $\boldsymbol{\Lambda}_m^k = [\mathrm{vec}(\mathbf{M}(t_1; \boldsymbol{\mu}^k)), \dots, \mathrm{vec}(\mathbf{M}(t_{N_t}; \boldsymbol{\mu}^k))]$
            $\boldsymbol{\Lambda}_g^k = [\mathbf{g}(t_1; \boldsymbol{\mu}^k), \dots, \mathbf{g}(t_{N_t}; \boldsymbol{\mu}^k)]$
        (1b) compress local snapshots matrices and generate global ones
            $\widetilde{\boldsymbol{\Lambda}}_u^k = \mathrm{POD}(\boldsymbol{\Lambda}_u^k, \varepsilon_u^{\mathrm{loc}}), \quad \widetilde{\boldsymbol{\Lambda}}_u = [\mathbf{V} \;\; \widetilde{\boldsymbol{\Lambda}}_u^k]$
            $\widetilde{\boldsymbol{\Lambda}}_a^k = \mathrm{POD}(\boldsymbol{\Lambda}_a^k, \varepsilon_a^{\mathrm{loc}}), \quad \widetilde{\boldsymbol{\Lambda}}_a = [\boldsymbol{\Phi}^a \;\; \widetilde{\boldsymbol{\Lambda}}_a^k]$
            $\widetilde{\boldsymbol{\Lambda}}_m^k = \mathrm{POD}(\boldsymbol{\Lambda}_m^k, \varepsilon_m^{\mathrm{loc}}), \quad \widetilde{\boldsymbol{\Lambda}}_m = [\boldsymbol{\Phi}^m \;\; \widetilde{\boldsymbol{\Lambda}}_m^k]$
            $\widetilde{\boldsymbol{\Lambda}}_g^k = \mathrm{POD}(\boldsymbol{\Lambda}_g^k, \varepsilon_g^{\mathrm{loc}}), \quad \widetilde{\boldsymbol{\Lambda}}_g = [\boldsymbol{\Phi}^g \;\; \widetilde{\boldsymbol{\Lambda}}_g^k]$
        (1c) extract global solution, matrix and vector bases
            $\mathbf{V} \;\; = \mathrm{POD}(\widetilde{\boldsymbol{\Lambda}}_u, \varepsilon_u)$
            $\boldsymbol{\Phi}^a = \mathrm{POD}(\widetilde{\boldsymbol{\Lambda}}_a, \varepsilon_a), \quad \boldsymbol{\Phi}^m = \mathrm{POD}(\widetilde{\boldsymbol{\Lambda}}_m, \varepsilon_m)$
            $\boldsymbol{\Phi}^g = \mathrm{POD}(\widetilde{\boldsymbol{\Lambda}}_g, \varepsilon_g)$
    end for
(2) perform MDEIM on $\boldsymbol{\Phi}^a$, $\boldsymbol{\Phi}^m$ and DEIM on $\boldsymbol{\Phi}^g$; generate a common reduced mesh.
    project resulting matrices and vectors on the reduced basis $\mathbf{V}$

Algorithm 3.5 Simultaneous system approximation and state-space reduction approach for problem (3.46). At each step k of the loop, we compress the local snapshots matrices $\boldsymbol{\Lambda}_u^k$, $\boldsymbol{\Lambda}_a^k$, $\boldsymbol{\Lambda}_m^k$, $\boldsymbol{\Lambda}_g^k$ by POD up to (user-defined) tolerances $\varepsilon_u^{\mathrm{loc}}$, $\varepsilon_a^{\mathrm{loc}}$, $\varepsilon_m^{\mathrm{loc}}$, $\varepsilon_g^{\mathrm{loc}}$, respectively. Alternatively, we could directly specify desired dimensions N^{loc}, M_a^{loc}, M_m^{loc}, M_g^{loc} (the same applies to step (1c)).

In the online phase, we thus solve the following fully-discrete ROM with system approximation (or hyper-ROM): given $\mathbf{y}_{N,m}^n(\boldsymbol{\mu}), \dots, \mathbf{y}_{N,m}^{n+1-\sigma}(\boldsymbol{\mu})$, for $n \geq \sigma - 1$, find $\mathbf{y}_{N,m}^{n+1}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ such that

$$\mathbf{M}_N^m(t_{n+1}; \boldsymbol{\mu}) \frac{\alpha \mathbf{y}_{N,m}^{n+1} - \mathbf{y}_{N,m}^{n,\sigma}}{\Delta t} + \mathbf{A}_N^m(t_{n+1}; \boldsymbol{\mu}) \mathbf{y}_{N,m}^{n+1} = \mathbf{g}_N^m(t_{n+1}; \boldsymbol{\mu}). \tag{3.52}$$

Remark 3.8. In the case of a non-zero initial condition $\mathbf{y}_h(0;\boldsymbol{\mu}) = \mathbf{y}_0(\boldsymbol{\mu})$, as suggested in [CFCA13], the basis $\mathbf{V}$ is generated by performing the POD on the increments

$$\{\mathbf{y}_h^n(\boldsymbol{\mu}^k) - \mathbf{y}_0(\boldsymbol{\mu}^k)\}_{n=1}^{N_t},$$

rather than on the snapshots themselves. •

Remark 3.9. Here, we assume the training inputs $\{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^K\}$ to be selected either a priori guided by physical intuition, or by sampling techniques like random or latin hypercube (LHS) sampling (see, e.g., [Coc07, Loh10]) and sparse grids (see, e.g., [GG98, BG04]). However, the offline construction in Algorithm 3.7 can accommodate more general training techniques such as the greedy [GP05] and POD-greedy [HO08] algorithms, possibly combined with adaptivity [HDO11, HSZ14], heuristic error indicators [PDTA14], and localized bases [WH15]. To this end, however, suitable a posteriori error estimates are needed. •

3.6.2 A posteriori error estimates

The goal of this section is to derive an a posteriori error estimate for the error $\mathbf{e}_N^m(t;\boldsymbol{\mu})$ between the solution $\mathbf{y}_h(t;\boldsymbol{\mu})$ of the semi-discrete FOM and the solution $\mathbf{y}_N^m(t;\boldsymbol{\mu})$ of the semi-discrete ROM with system approximation

$$\mathbf{e}_N^m(t;\boldsymbol{\mu}) = \mathbf{y}_h(t;\boldsymbol{\mu}) - \mathbf{V}\mathbf{y}_{N,m}(t;\boldsymbol{\mu}). \tag{3.53}$$

As in the steady case, this error can be decomposed as $\mathbf{e}_N^m(t;\boldsymbol{\mu}) = \mathbf{e}_m(t;\boldsymbol{\mu}) + \mathbf{e}_N(t;\boldsymbol{\mu})$, where $\mathbf{e}_m(t;\boldsymbol{\mu})$ is the error arising from the system approximation

$$\mathbf{e}_m(t;\boldsymbol{\mu}) = \mathbf{y}_h(t;\boldsymbol{\mu}) - \mathbf{y}_{h,m}(t;\boldsymbol{\mu}), \tag{3.54}$$

and $\mathbf{e}_N(t;\boldsymbol{\mu})$ is the error arising from the state-space reduction

$$\mathbf{e}_N(t;\boldsymbol{\mu}) = \mathbf{y}_{h,m}(t;\boldsymbol{\mu}) - \mathbf{V}\mathbf{y}_{N,m}(t;\boldsymbol{\mu}). \tag{3.55}$$

For the sake of simplicity, we assume here the matrix $\mathbf{M}$ to be $(t,\boldsymbol{\mu})$-independent, and both matrices $\mathbf{M}$ and $\mathbf{A}(t;\boldsymbol{\mu})$ to be symmetric positive definite. Moreover, it is useful to first introduce the following stability factors

$$\beta_h(t;\boldsymbol{\mu}) = \lambda_{\min}(\mathbf{X}^{-\frac{1}{2}}\mathbf{A}(t;\boldsymbol{\mu})\mathbf{X}^{-\frac{1}{2}}), \qquad \beta_h^m(t;\boldsymbol{\mu}) = \lambda_{\min}(\mathbf{X}^{-\frac{1}{2}}\mathbf{A}_m(t;\boldsymbol{\mu})\mathbf{X}^{-\frac{1}{2}}), \tag{3.56}$$

and the residual $\mathbf{r}_m(\mathbf{y}_{N,m};t;\boldsymbol{\mu})$

$$\mathbf{r}_m(\mathbf{y}_{N,m};t;\boldsymbol{\mu}) = \mathbf{g}_m(t;\boldsymbol{\mu}) - \mathbf{A}_m(t;\boldsymbol{\mu})\mathbf{V}\mathbf{y}_{N,m} - \mathbf{M}\mathbf{V}\frac{d\mathbf{y}_{N,m}}{dt}. \tag{3.57}$$

While it is possible to estimate separately the two error components (as we did in Sect. 3.4), we now directly derive an estimate for the $\mathbf{M}$-norm of the error $\mathbf{e}_N^m$.

Proposition 3.3. *The error* $\mathbf{e}_N^m(t;\boldsymbol{\mu})$ *is bounded as*

$$\|\mathbf{e}_N^m(t;\boldsymbol{\mu})\|_{\mathbf{M}}^2 \leq \Delta_m(t;\boldsymbol{\mu}) + \Delta_N(t;\boldsymbol{\mu}), \tag{3.58}$$

where

$$\Delta_N(t;\boldsymbol{\mu}) = \|\mathbf{e}_N(0;\boldsymbol{\mu})\|^2_{\mathbf{M}} + \int_0^t \frac{1}{\beta_h(s;\boldsymbol{\mu})} \|\mathbf{r}_m(\mathbf{y}_{N,m};s;\boldsymbol{\mu})\|^2_{\mathbf{X}^{-1}}\, ds,$$

and

$$\Delta_m(t;\boldsymbol{\mu}) = \int_0^t \frac{1}{\beta_h(s;\boldsymbol{\mu})} \Big(\| \mathbf{g}_m(s;\boldsymbol{\mu}) - \mathbf{g}(s;\boldsymbol{\mu})\|^2_{\mathbf{X}^{-1}} + \|\mathbf{A}_m(s;\boldsymbol{\mu}) - \mathbf{A}(s;\boldsymbol{\mu})\|^2_{\mathbf{X},\mathbf{X}^{-1}} \|\mathbf{V}\mathbf{y}_{N,m}(s;\boldsymbol{\mu})\|^2_{\mathbf{X}} \Big)\, ds.$$

Proof. We start by observing that

$$\begin{aligned}\mathbf{M}\frac{d\mathbf{e}_N^m}{dt} + \mathbf{A}(t;\boldsymbol{\mu})\mathbf{e}_N^m &= \mathbf{g}(t;\boldsymbol{\mu}) - \mathbf{M}\mathbf{V}\frac{d\mathbf{y}_{N,m}}{dt} - \mathbf{A}(t;\boldsymbol{\mu})\mathbf{V}\mathbf{y}_{N,m} \\ &= \mathbf{g}(t;\boldsymbol{\mu}) - \mathbf{g}_m(t;\boldsymbol{\mu}) + (\mathbf{A}_m(t;\boldsymbol{\mu}) - \mathbf{A}(t;\boldsymbol{\mu}))\mathbf{V}\mathbf{y}_{N,m} + \mathbf{r}_m(\mathbf{y}_{N,m};t;\boldsymbol{\mu}).\end{aligned}$$

Pre-multiplying both sides of the equation by $\mathbf{e}_N^{mT}$ leads to

$$\begin{aligned}&\frac{1}{2}\frac{d\|\mathbf{e}_N^m\|^2_{\mathbf{M}}}{dt} + \mathbf{e}_N^{mT}\mathbf{A}(t;\boldsymbol{\mu})\mathbf{e}_N^m \\ &\qquad = \mathbf{e}_N^{mT}\Big(\mathbf{g}(t;\boldsymbol{\mu}) - \mathbf{g}_m(t;\boldsymbol{\mu}) + (\mathbf{A}_m(t;\boldsymbol{\mu}) - \mathbf{A}(t;\boldsymbol{\mu}))\mathbf{V}\mathbf{y}_{N,m} + \mathbf{r}_m(\mathbf{y}_{N,m};t;\boldsymbol{\mu})\Big).\end{aligned}$$

By exploiting the positive definiteness of $\mathbf{A}$ in the left-hand side, Cauchy-Schwarz and Young inequalities in the right-hand side, we obtain

$$\begin{aligned}\frac{1}{2}\frac{d\|\mathbf{e}_N^m\|^2_{\mathbf{M}}}{dt} + \beta_h(t;\boldsymbol{\mu})\|\mathbf{e}_N^m\|^2_{\mathbf{X}} &\le \frac{\beta_h(t;\boldsymbol{\mu})}{2}\|\mathbf{e}_N^m\|^2_{\mathbf{X}} \\ &\quad + \frac{1}{2\beta_h(t;\boldsymbol{\mu})}\Big(\|\mathbf{g}(t;\boldsymbol{\mu}) - \mathbf{g}_m(t;\boldsymbol{\mu})\|^2_{\mathbf{X}^{-1}} \\ &\quad + \|(\mathbf{A}_m(t;\boldsymbol{\mu}) - \mathbf{A}(t;\boldsymbol{\mu}))\mathbf{V}\mathbf{y}_{N,m}\|^2_{\mathbf{X}^{-1}} + \|\mathbf{r}_m(\mathbf{y}_{N,m};t;\boldsymbol{\mu})\|^2_{\mathbf{X}^{-1}}\Big).\end{aligned}$$

Integrating over $(0,t)$, $t \in (0,T]$, we end up with

$$\begin{aligned}\|\mathbf{e}_N^m(t;\boldsymbol{\mu})\|^2_{\mathbf{M}} + \int_0^t \beta_h(s;\boldsymbol{\mu})\|\mathbf{e}_N^m(s;\boldsymbol{\mu})\|^2_{\mathbf{X}}\, ds &\le \|\mathbf{e}_N(0;\boldsymbol{\mu})\|^2_{\mathbf{M}} \\ &\quad + \int_0^t \frac{1}{\beta_h(s;\boldsymbol{\mu})}\Big(\|\mathbf{r}_m(\mathbf{y}_{N,m};s;\boldsymbol{\mu})\|^2_{\mathbf{X}^{-1}} + \|\mathbf{g}_m(s;\boldsymbol{\mu}) - \mathbf{g}(s;\boldsymbol{\mu})\|^2_{\mathbf{X}^{-1}} \\ &\quad + \|(\mathbf{A}_m(s;\boldsymbol{\mu}) - \mathbf{A}(s;\boldsymbol{\mu}))\mathbf{V}\mathbf{y}_{N,m}\|^2_{\mathbf{X}^{-1}}\Big)\, ds,\end{aligned}$$

from which (3.58) easily follows. □

We remark that the dual norm of the residual can be efficiently computed by a proper offline-online decomposition as described in [HO09], while the contributes due to system approximation can be approximated as in (3.43). Moreover, to practically evaluate the above error bound, the integrals over time must be approximated by a quadrature rule.

3.7 Hyper-reduction of parametrized Navier-Stokes equations

In this section, we take advantage of the techniques presented so far to develop a suitable hyper-reduction strategy for the Navier-Stokes equations. This strategy is tailored to the underlying high-fidelity approximation, which employs equal-order SUPG stabilized finite elements for the space discretization, a BDF time discretization and a semi-implicit treatment of the convective term [FD15, GSV06]. The reduced-order model is then generated by a Galerkin projection of the resulting fully-discrete problem onto a POD basis. A hybrid approach for the treatment of the equations nonlinear operators is employed: we apply an exact quadratic expansion to reconstruct the convective term, while MDEIM is used to approximate the nonlinear (with respect to the convective velocity) SUPG terms. The construction is first presented in the parameter-independent case and then extended to the parametrized one.

3.7.1 Weak formulation

Let $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ be an open bounded domain with piecewise smooth boundary $\Gamma = \partial\Omega$. The latter is decomposed into Dirichlet and Neumann components such that $\Gamma = \Gamma_D \cup \Gamma_N$. The Navier-Stokes equations for an incompressible, homogeneous, Newtonian fluid read:

$$\begin{cases} \dfrac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} - \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{v}, p) = \mathbf{0} & \text{in } \Omega \times (0, T) \\ \nabla \cdot \boldsymbol{v} = 0 & \text{in } \Omega \times (0, T) \\ \boldsymbol{v} = \boldsymbol{h} & \text{on } \Gamma_D \times (0, T) \\ \boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n} = \mathbf{0} & \text{on } \Gamma_N \times (0, T) \\ \boldsymbol{v}(0) = \boldsymbol{v}_0 & \text{in } \Omega, \end{cases} \tag{3.59}$$

where $\boldsymbol{v} = \boldsymbol{v}(\mathbf{x}, t)$ is the fluid velocity, $p = p(\mathbf{x}, t)$ the kinematic pressure (i.e. the ratio between the fluid pressure and its density), $\boldsymbol{n}$ the (outward directed) normal unit vector to Γ_N, and $\boldsymbol{\sigma}$ is the stress tensor defined as

$$\boldsymbol{\sigma}(\boldsymbol{v}, p) = -p\mathbf{I} + 2\nu\boldsymbol{\varepsilon}(\boldsymbol{v}). \tag{3.60}$$

Here ν denotes the kinematic viscosity of the fluid (i.e. $\nu = \mu/\rho$, being μ and ρ the dynamic viscosity and density, respectively), while

$$\boldsymbol{\varepsilon}(\boldsymbol{v}) = \frac{1}{2}\Big(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T\Big) \tag{3.61}$$

is the strain tensor. The function $\boldsymbol{h} = \boldsymbol{h}(\mathbf{x}, t)$ indicates the Dirichlet data, while $\boldsymbol{v}_0 = \boldsymbol{v}_0(\mathbf{x})$ is the initial condition.

For this study, problem (3.59) will be parametrized by a vector $\boldsymbol{\mu} \in \mathcal{D}$ of parameters affecting the physical configuration via the boundary conditions, the initial condition, or the kinematic viscosity ν. More involved parametrization affecting the geometry of the domain are not considered. For the time being however, we omit the $\boldsymbol{\mu}$-dependence in the formulation of the problem.

Let us introduce the following functional spaces: $V_D = \{\boldsymbol{w} \in [H^1(\Omega)]^d : \boldsymbol{w}|_{\Gamma_D} = \boldsymbol{h}\}$, $V = \{\boldsymbol{w} \in [H^1(\Omega)]^d : \boldsymbol{w}|_{\Gamma_D} = \boldsymbol{0}\}$ and $M = L^2(\Omega)$. The weak formulation of (3.59) reads: for all $t \in (0,T)$, find $(\boldsymbol{v}(t), p(t)) \in V_D \times M$ such that

$$\left(\frac{\partial \boldsymbol{v}}{\partial t}, \boldsymbol{w}\right) + (\boldsymbol{v} \cdot \nabla \boldsymbol{v}, \boldsymbol{w}) + (\nu(\nabla \boldsymbol{v} + \nabla \boldsymbol{v}^T), \nabla \boldsymbol{w}) - (p, \nabla \cdot \boldsymbol{w}) + (\nabla \cdot \boldsymbol{v}, q) = 0 \quad (3.62)$$

for all $(\boldsymbol{w}, q) \in V \times M$, with $\boldsymbol{v}(0) = \boldsymbol{v}_0$.

3.7.2 Semi-implicit SUPG-stabilized finite element approximation

Let us introduce a FE partition $\mathcal{T}_h$ of the domain Ω from which we construct conforming finite element spaces $V_h \subset V$ and $M_h \subset M$. For the discrete version of problem (3.62) to be well-posed, it is well known that the velocity and pressure spaces V_h and M_h need to fulfill an inf-sup condition

$$\inf_{q_h \in M_h} \sup_{\boldsymbol{w}_h \in V_h} \frac{(q_h, \nabla \cdot \boldsymbol{w}_h)}{\|\boldsymbol{w}_h\|_V \|q_h\|_Q} \geq \bar{\beta} > 0. \quad (3.63)$$

Low-order approximation spaces (such as $\mathbb{P}_1$-$\mathbb{P}_1$ spaces) represent an attractive option, since they mitigate the required computational effort; however, they do not satisfy (3.63). As a remedy, one can resorts to suitable pressure stabilizations, which allow to circumvent the inf-sup condition (3.63), see e.g. [BP84, DB04, BB01, ESW04, BF08]. Nevertheless, pressure stabilizations turn out to be inappropriate when dealing with advection dominated flows, since in this case additional terms are required to enhance the stability with respect to the convective terms and to control the incompressibility constraint.

For these reasons, we resort to the streamline upwind Petrov-Galerkin (SUPG) stabilization – formulated as in the Variational Multiscale framework [HSF04, BCC$^+$07] – which satisfies all these requisites. To this end, let us first introduce the finite element space

$$X_h^r = \{w_h \in C^0(\overline{\Omega}) : w_h|_K \in \mathbb{P}^r \ \forall K \in \mathcal{T}_h\}.$$

Then, we define $V_h = V \cap [X_h^r]^d$, $V_{D,h} = V_D \cap [X_h^r]^d$ and $M_h = M \cap X_h^r$; note that we use equal-order FE spaces for the velocity and pressure variables. We also introduce the strong residuals $\boldsymbol{r}_M(\boldsymbol{v}_h, p_h)$ and $r_C(\boldsymbol{v}_h)$ of the momentum and continuity equations, respectively:

$$\boldsymbol{r}_M(\boldsymbol{v}_h, p_h) = \frac{\partial \boldsymbol{v}_h}{\partial t} + \boldsymbol{v}_h \cdot \nabla \boldsymbol{v}_h + \nabla p - \nu \Delta \boldsymbol{v}_h, \quad (3.64)$$

$$r_C(\boldsymbol{v}_h) = \nabla \cdot \boldsymbol{v}_h. \quad (3.65)$$

The semi-discrete SUPG formulation of the Navier-Stokes equations reads: for all $t \in (0,T)$, find $(\boldsymbol{v}_h, p_h) \in V_{D,h} \times M_h$ such that

$$\left(\frac{\partial \boldsymbol{v}_h}{\partial t}, \boldsymbol{w}_h\right) + (\boldsymbol{v}_h \cdot \nabla \boldsymbol{v}_h, \boldsymbol{w}_h) + (\nu(\nabla \boldsymbol{v}_h + \nabla \boldsymbol{v}_h^T), \nabla \boldsymbol{w}_h) - (p_h, \nabla \cdot \boldsymbol{w}_h) + (\nabla \cdot \boldsymbol{v}_h, q_h)$$
$$+ \sum_{K \in \mathcal{T}_h} (\tau_M \boldsymbol{r}_M(\boldsymbol{v}_h, p_h), \boldsymbol{v}_h \cdot \nabla \boldsymbol{w}_h + \nabla q_h)_K + \sum_{K \in \mathcal{T}_h} (\tau_C r_C(\boldsymbol{v}_h), \nabla \cdot \boldsymbol{w}_h)_K = 0 \quad (3.66)$$

for all $(\boldsymbol{w}_h, q_h) \in V_h \times M_h$, with $\boldsymbol{v}_h(0) = \boldsymbol{v}_0$. The stabilization parameters $\tau_M = \tau_M(\boldsymbol{v}_h)$ and $\tau_C = \tau_C(\boldsymbol{v}_h)$ are defined (element-wise) as in [BCC+07]:

$$\tau_M = \left(\frac{\sigma^2}{(\Delta t)^2} + \boldsymbol{v}_h \cdot \mathbf{G}_K \boldsymbol{v}_h + C_I \nu^2 \mathbf{G}_K : \mathbf{G}_K \right)^{-1/2}, \qquad \tau_C = \Big(\tau_M \mathbf{g}_K \cdot \mathbf{g}_K \Big)^{-1}, \quad (3.67)$$

where $C_I = 60 \cdot 2^{r-2}$, σ is a constant equal to the order of the time discretization and Δt is the time step that will be chosen for the time discretization. Moreover, $\mathbf{G}_K$ and $\mathbf{g}_K$ are metric tensors of the computational domain, which can be derived from the inverse Jacobian of the mapping between the reference and physical elements as

$$(\mathbf{g}_K)_i = \sum_{j=1}^{d} \frac{\partial \xi_j}{\partial x_i}, \qquad (\mathbf{G}_K)_{ij} = \sum_{l=1}^{d} \frac{\partial \xi_l}{\partial x_i} \frac{\partial \xi_l}{\partial x_j}, \qquad i, j = 1, \dots, d,$$

being ξ_i and x_i the reference and physical coordinates, respectively.

For the time discretization of (3.66) we consider the BDF scheme with semi-implicit treatment of the convective terms proposed in [FD15] (see also [GSV06]). This approach allows to mitigate the computational burden associated to the use of a fully implicit BDF scheme by linearizing the nonlinear convective terms. The linearization is done by extrapolating the convective velocity via an extrapolation formula of the same order of the BDF used.

To begin with, we partition the time interval $[0, T]$ into N_t subintervals of equal size $\Delta t = T/N_t$ and denote by $t_n = n\Delta t$, for $n = 0, \dots, N_t$ the discrete time instances. Moreover, we denote by $\boldsymbol{v}_h^n$ and p_h^n the approximations of $\boldsymbol{v}_h$ and p_h at time t_n, respectively. Following Sect. 3.6.1, we approximate the time derivative of the velocity as

$$\frac{\partial \boldsymbol{v}_h}{\partial t} \approx \frac{\alpha \boldsymbol{v}_h^{n+1} - \boldsymbol{v}_h^{n,\sigma}}{\Delta t}.$$

Then, we approximate the convective velocity at time t^{n+1} with the Lagrange polynomial interpolating $\boldsymbol{v}_h^n, \dots, \boldsymbol{v}_h^{n-\sigma+1}$ evaluated at time t_{n+1}. We thus obtain the following expression for the extrapolated velocity at time t^{n+1}

$$\boldsymbol{v}_h^{n,*} = \begin{cases} \boldsymbol{v}_h^n, & \text{if } n \geq 0, \quad \text{for } \sigma = 1 \text{ (BDF1)}, \\ 2\boldsymbol{v}_h^n - \boldsymbol{v}_h^{n-1}, & \text{if } n \geq 1, \quad \text{for } \sigma = 2 \text{ (BDF2)}. \end{cases} \quad (3.68)$$

The (fully discrete) semi-implicit BDF-SUPG approximation of the Navier-Stokes equations reads: given $\boldsymbol{v}_h^n, \dots, \boldsymbol{v}_h^{n+1-\sigma}$, for $n \geq \sigma - 1$ find $(\boldsymbol{v}_h^{n+1}, p_h^{n+1}) \in V_h \times M_h$ such that

$$\begin{aligned} &\left(\frac{\alpha \boldsymbol{v}_h^{n+1} - \boldsymbol{v}_h^{n,\sigma}}{\Delta t}, \boldsymbol{w}_h \right) + \left(\boldsymbol{v}_h^{n,*} \cdot \nabla \boldsymbol{v}_h^{n+1}, \boldsymbol{w}_h \right) + \left(\nu (\nabla \boldsymbol{v}_h^{n+1} + (\nabla \boldsymbol{v}_h^{n+1})^T), \nabla \boldsymbol{w}_h \right) \\ &- \left(p_h^{n+1}, \nabla \cdot \boldsymbol{w}_h \right) + \left(\nabla \cdot \boldsymbol{v}_h^{n+1}, q_h \right) + \sum_K (\tau_M^* \boldsymbol{r}_M^*(\boldsymbol{v}_h^{n+1}, p_h^{n+1}), \boldsymbol{v}_h^{n,*} \cdot \nabla \boldsymbol{w}_h + \nabla q_h)_K \\ &+ \sum_K (\tau_C^* r_C(\boldsymbol{v}_h^{n+1}), \nabla \cdot \boldsymbol{w}_h)_K = g((\boldsymbol{w}_h, q_h); t_{n+1}) \qquad \forall (\boldsymbol{w}_h, q_h) \in V_h \times M_h, \quad (3.69) \end{aligned}$$

where

$$\boldsymbol{r}_M^*(\boldsymbol{v}_h^{n+1}, p_h^{n+1}) = \frac{\alpha \boldsymbol{v}_h^{n+1} - \boldsymbol{v}_h^{n,\sigma}}{\Delta t} + \boldsymbol{v}_h^{n,*} \cdot \nabla \boldsymbol{v}_h^{n+1} + \nabla p^{n+1} - \nu \Delta \boldsymbol{v}_h^{n+1} \tag{3.70}$$

is the residual of the momentum equation, and

$$\tau_M^* = \left(\frac{\sigma^2}{(\Delta t)^2} + \boldsymbol{v}_h^{n,*} \cdot \mathbf{G}_K \boldsymbol{v}_h^{n,*} + C_I \nu^2 \mathbf{G}_K : \mathbf{G}_K \right)^{-1/2}, \qquad \tau_C^* = \left(\tau_M^* \mathbf{g}_K \cdot \mathbf{g}_K \right)^{-1} \tag{3.71}$$

are the stabilization parameters. Here, the functional $g(\cdot; t) : V_h \times M_h \to \mathbb{R}$ encodes the action of the nonhomogeneous Dirichlet condition $\boldsymbol{v}_h|_{\Gamma_D} = \boldsymbol{h}(t)$.

Remark 3.10. Problem (3.69) can be similarly obtained by first introducing the semi-discrete (in time) approximation (see [GSV06])

$$\begin{cases} \dfrac{\alpha \boldsymbol{v}^{n+1} - \boldsymbol{v}^{n,\sigma}}{\Delta t} + \boldsymbol{v}^{n,*} \cdot \nabla \boldsymbol{v}^{n+1} - \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{v}^{n+1}, p^{n+1}) = \mathbf{0} & \text{in } \Omega \\ \nabla \cdot \boldsymbol{v}^{n+1} = 0 & \text{in } \Omega \\ \boldsymbol{v}^{n+1} = \boldsymbol{h}(t^{n+1}) & \text{on } \Gamma_D \\ \boldsymbol{\sigma}(\boldsymbol{v}^{n+1}, p^{n+1}) \boldsymbol{n} = \mathbf{0} & \text{on } \Gamma_N \end{cases} \tag{3.72}$$

and then discretizing the resulting Oseen equations by finite elements with SUPG stabilization. •

3.7.3 Algebraic formulation

We denote by $\{\boldsymbol{\varphi}_i\}_{i=1}^{N_{h,v}}$ and $\{\eta_k\}_{k=1}^{N_{h,p}}$ Lagrangian FE bases for V_h and M_h respectively. We also denote by $\mathbf{v}_h^n \in \mathbb{R}^{N_{h,v}}$ and $\mathbf{p}_h^n \in \mathbb{R}^{N_{h,p}}$ the vectors of coefficients in the expansions of $\boldsymbol{v}_h^n$ and p_h^n with respect to the FE bases. Finally, we set $\mathbf{U}_h^n = (\mathbf{v}_h^n, \mathbf{p}_h^n) \in \mathbb{R}^{N_h}$. The algebraic formulation of (3.69) reads: given $\mathbf{U}_h^n, \ldots, \mathbf{U}_h^{n+1-\sigma}$, for $n \geq \sigma - 1$ find $\mathbf{U}_h^{n+1} \in \mathbb{R}^{N_h}$ such that

$$\left(\frac{\alpha}{\Delta t} \mathbf{M}(\mathbf{U}_h^{n,*}) + \mathbf{A} + \mathbf{C}(\mathbf{U}_h^*) + \mathbf{S}(\mathbf{U}_h^{n,*}) \right) \mathbf{U}_h^{n+1} = \frac{1}{\Delta t} \mathbf{M} \mathbf{U}_h^{n,\sigma} + \mathbf{g}(t_{n+1}) + \mathbf{f}^S(\mathbf{U}_h^{n,*}, \mathbf{U}_h^{n,\sigma}). \tag{3.73}$$

The $N_h \times N_h$ block-partitioned matrices $\mathbf{M}$, $\mathbf{A}$ and $\mathbf{C}(\mathbf{U}_h^{n,*})$ are defined as

$$\mathbf{M} = \begin{pmatrix} \mathbf{M}_v & 0 \\ 0 & 0 \end{pmatrix}, \qquad \mathbf{A} = \begin{pmatrix} \nu \mathbf{K} & \mathbf{B}^T \\ -\mathbf{B} & 0 \end{pmatrix}, \qquad \mathbf{C}(\mathbf{U}_h^{n,*}) = \begin{pmatrix} \widetilde{\mathbf{C}}(\mathbf{v}_h^{n,*}) & 0 \\ 0 & 0 \end{pmatrix} \tag{3.74}$$

where $\mathbf{M}_v \in \mathbb{R}^{N_{h,v} \times N_{h,v}}$ is the velocity mass matrix, $\mathbf{K} \in \mathbb{R}^{N_{h,v} \times N_{h,v}}$ is the velocity stiffness matrix, $\mathbf{B} \in \mathbb{R}^{N_{h,v} \times N_{h,v}}$ is the divergence matrix and $\widetilde{\mathbf{C}}(\mathbf{v}_h^{n,*})$ is the convective matrix defined as

$$(\widetilde{\mathbf{C}}(\mathbf{v}_h^{n,*}))_{ij} = \left(\mathbf{v}_h^{n,*} \cdot \nabla \boldsymbol{\varphi}_j, \boldsymbol{\varphi}_i \right) \qquad i, j = 1, \ldots, N_{h,v}. \tag{3.75}$$

Similarly, the matrix $\mathbf{S}(\mathbf{U}_h^{n,*}) \in \mathbb{R}^{N_h \times N_h}$ encoding the SUPG operator is defined as

$$\mathbf{S}(\mathbf{U}_h^{n,*}) = \begin{pmatrix} \mathbf{S}_{vv}(\mathbf{v}_h^{n,*}) & \mathbf{S}_{vp}(\mathbf{v}_h^{n,*}) \\ \mathbf{S}_{pv}(\mathbf{v}_h^{n,*}) & \mathbf{S}_{pp}(\mathbf{v}_h^{n,*}) \end{pmatrix},$$

with

$$\left(\mathbf{S}_{vv}(\mathbf{v}_h^{n,*})\right)_{ij} = \left(\tau_M^* \left(\tfrac{\alpha}{\Delta t}\boldsymbol{\varphi}_j + \mathbf{v}_h^{n,*} \cdot \nabla \boldsymbol{\varphi}_j - \nu \Delta \boldsymbol{\varphi}_j\right), \mathbf{v}_h^{n,*} \cdot \nabla \boldsymbol{\varphi}_i\right) + \left(\tau_C^* \nabla \cdot \boldsymbol{\varphi}_j, \nabla \cdot \boldsymbol{\varphi}_i\right),$$

$$\left(\mathbf{S}_{pv}(\mathbf{v}_h^{n,*})\right)_{kj} = \left(\tau_M^* \left(\tfrac{\alpha}{\Delta t}\boldsymbol{\varphi}_j + \mathbf{v}_h^{n,*} \cdot \nabla \boldsymbol{\varphi}_j - \nu \Delta \boldsymbol{\varphi}_j\right), \nabla \eta_k\right),$$

$$\left(\mathbf{S}_{vp}(\mathbf{v}_h^{n,*})\right)_{ik} = \left(\tau_M^* \nabla \eta_k, \mathbf{v}_h^{n,*} \cdot \nabla \boldsymbol{\varphi}_i\right), \qquad \left(\mathbf{S}_{pp}(\mathbf{v}_h^{n,*})\right)_{kl} = \left(\tau_M^* \nabla \eta_l, \nabla \eta_k\right),$$

for $i, j = 1, \ldots, N_{h,v}$ and $k, l = 1, \ldots, N_{h,p}$. The SUPG contribute to the right-hand side is instead given by $\mathbf{f}^S(\mathbf{U}_h^{n,*}, \mathbf{U}_h^{n,\sigma}) = (\mathbf{f}_v^S, \mathbf{f}_p^S)^T$, where

$$(\mathbf{f}_v^S)_i = \left(\tau_M^* \tfrac{1}{\Delta t}\mathbf{v}_h^{n,\sigma}, \mathbf{v}_h^{n,*} \cdot \nabla \boldsymbol{\varphi}_i\right), \qquad (\mathbf{f}_v^S)_k = \left(\tau_M^* \tfrac{1}{\Delta t}\mathbf{v}_h^{n,\sigma}, \nabla \eta_k\right).$$

Thanks to the semi-implicit treatment of the nonlinear terms, the fully-discrete system (3.69) yields a linear problem – rather than a nonlinear one – to be solved at each time t_n. Moreover, the matrices $\mathbf{M}$ and $\mathbf{A}$ are constant in time and independent of $\mathbf{U}_h^{n,*}$, so that they can be assembled once and for all at $t = t_0$. On the other hand, the matrices $\mathbf{C}$ and $\mathbf{S}$ depend on $\mathbf{U}_h^{n,*}$ and need to be assembled at each time step. We also remark that while $\mathbf{C}$ is linear with respect to $\mathbf{U}_h^{n,*}$, $\mathbf{S}$ depends nonlinearly on $\mathbf{U}_h^{n,*}$. Therefore, a suitable system approximation strategy will be developed in Sect. 3.7.5 to generate an affine approximation of $\mathbf{S}$. To this end, however, it is convenient to express problem (3.73) in the following equivalent form: given $\mathbf{U}_h^n, \ldots, \mathbf{U}_h^{n+1-\sigma}$, for $n \geq \sigma - 1$ find $\mathbf{U}_h^{n+1} \in \mathbb{R}^{N_h}$ such that

$$\left(\frac{\alpha}{\Delta t}\widetilde{\mathbf{M}}(\mathbf{U}_h^{n,*}) + \mathbf{A} + \mathbf{C}(\mathbf{U}_h^{n,*}) + \widetilde{\mathbf{S}}(\mathbf{U}_h^{n,*})\right)\mathbf{U}_h^{n+1} = \frac{1}{\Delta t}\widetilde{\mathbf{M}}(\mathbf{U}_h^{n,*})\mathbf{U}_h^{n,\sigma} + \mathbf{g}(t_{n+1}), \quad (3.76)$$

where the block-partitioned matrices $\widetilde{\mathbf{M}}$ and $\widetilde{\mathbf{S}}$ are defined as

$$\widetilde{\mathbf{M}}(\mathbf{U}_h^{n,*}) = \begin{pmatrix} \mathbf{M}_v + \widetilde{\mathbf{M}}_{vv}(\mathbf{v}_h^{n,*}) & 0 \\ \widetilde{\mathbf{M}}_{pv}(\mathbf{v}_h^{n,*}) & 0 \end{pmatrix}, \qquad \widetilde{\mathbf{S}}(\mathbf{U}_h^{n,*}) = \begin{pmatrix} \widetilde{\mathbf{S}}_{vv}(\mathbf{v}_h^{n,*}) & \mathbf{S}_{vp}(\mathbf{v}_h^{n,*}) \\ \widetilde{\mathbf{S}}_{pv}(\mathbf{v}_h^{n,*}) & \mathbf{S}_{pp}(\mathbf{v}_h^{n,*}) \end{pmatrix}, \quad (3.77)$$

with

$$\left(\widetilde{\mathbf{S}}_{vv}(\mathbf{v}_h^{n,*})\right)_{ij} = \left(\tau_M^* \left(\mathbf{v}_h^{n,*} \cdot \nabla \boldsymbol{\varphi}_j - \nu \Delta \boldsymbol{\varphi}_j\right), \mathbf{v}_h^{n,*} \cdot \nabla \boldsymbol{\varphi}_i\right) + \left(\tau_C^* \nabla \cdot \boldsymbol{\varphi}_j, \nabla \cdot \boldsymbol{\varphi}_i\right),$$

$$\left(\widetilde{\mathbf{S}}_{pv}(\mathbf{v}_h^{n,*})\right)_{kj} = \left(\tau_M^* \left(\mathbf{v}_h^{n,*} \cdot \nabla \boldsymbol{\varphi}_j - \nu \Delta \boldsymbol{\varphi}_j\right), \nabla \eta_k\right),$$

and

$$\left(\widetilde{\mathbf{M}}_{vv}(\mathbf{v}_h^{n,*})\right)_{ij} = \left(\tau_M^* \boldsymbol{\varphi}_j, \mathbf{v}_h^{n,*} \cdot \nabla \boldsymbol{\varphi}_i\right), \qquad \left(\widetilde{\mathbf{M}}_{pv}(\mathbf{v}_h^{n,*})\right)_{kj} = \left(\tau_M^* \boldsymbol{\varphi}_j, \nabla \eta_k\right),$$

for $i, j = 1, \ldots, N_{h,v}$ and $k, l = 1, \ldots, N_{h,p}$. In the following, (3.76) shall represent our high-fidelity problem; for the sake of readability, we omit the symbol $\widetilde{\ }$ on the matrices $\mathbf{M}$ and $\mathbf{S}$ as no ambiguity occurs.

3.7.4 State space reduction

To reduce the dimension of (3.76), we seek an approximate solution

$$\mathbf{U}_h^{n+1} \approx \mathbf{V}\mathbf{U}_N^{n+1} \tag{3.78}$$

belonging to the subspace generated by the columns of a reduced-order basis $\mathbf{V} \in \mathbb{R}^{N_h \times N}$. The latter is constructed starting from suitable velocity and pressure POD bases $\mathbf{V}_v \in \mathbb{R}^{N_{h,v} \times N_v}$ and $\mathbf{V}_p \in \mathbb{R}^{N_{h,v} \times N_p}$ as

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_v & 0 \\ 0 & \mathbf{V}_p \end{pmatrix}.$$

Inserting (3.78) into (3.76) and operating a Galerkin projection, we obtain the following reduced-order model: given $\mathbf{U}_N^n, \dots, \mathbf{U}_N^{n+1-\sigma}$, for $n \geq \sigma - 1$ find $\mathbf{U}_N^{n+1} \in \mathbb{R}^N$ such that

$$\begin{aligned} \left(\frac{\alpha}{\Delta t} \mathbf{M}_N(\mathbf{U}_N^{n,*}) + \mathbf{A}_N + \mathbf{C}_N(\mathbf{U}_N^{n,*}) + \mathbf{S}_N(\mathbf{U}_N^{n,*}) \right) \mathbf{U}_N^{n+1} \\ = \frac{1}{\Delta t} \mathbf{M}_N(\mathbf{U}_N^{n,*}) \mathbf{U}_N^{n,\sigma} + \mathbf{g}_N(t_{n+1}), \end{aligned} \tag{3.79}$$

where the reduced matrices $\mathbf{M}_N$, $\mathbf{A}_N$, $\mathbf{C}_N$ and $\mathbf{S}_N$ are obtained by left and right multiplying their finite element counterparts by $\mathbf{V}$. Similarly for the right-hand side contribute.

As in many previous works [BGL06, WABI12, WLBI10, BBI09, IW14, XFB+14, Col14] on model order reduction for Navier-Stokes equations, a Galerkin projection onto a POD basis is employed to generate the ROM. Depending on the underlying high-fidelity approximation, this kind of projection may lead to a singular system if both velocity and pressure are sought online. Indeed, if the equations are approximated using a stable pair of velocity and pressure spaces (such as Taylor-Hood finite elements), then the POD basis for the velocity is (discretely) divergence free since all velocity snapshots satisfy the continuity equation. As a result, the inf-sup condition (3.63) is violated at the reduced level and the matrix $\mathbf{A}_N$ is singular. Different strategies have been proposed to overcome this issue. For instance, enrichment of the velocity space by so-called supremizer functions was proposed in [RV07] (see [Col14, Bal15] for recent applications), while a pressure reconstruction based on the Poisson equation was introduced in [NPM05, ANR09] (see also [CIJS14, TALL15] and references therein). A different strategy relies instead on a least-squares – rather than Galerkin – projection onto the POD basis, see e.g. [CFCA13, WH15, BCI15].

In our approach, the POD basis for the velocity is not divergence free, but fulfills instead a stabilized continuity equation. Since the additional stabilization terms in the variational formulation prevent a possible violation of the inf-sup condition, no further treatment for the pressure terms in the ROM is required. We note that a similar approach, yet combined with an explicit formulation of the ROM, was proposed in [BCI13].

Let us now discuss the computational costs associated with the evaluation of the ROM (3.79). The time-invariant matrix $\mathbf{A}_N$ can be formed once and for all at $t = t_0$. Moreover, thanks to its linearity with respect to $\mathbf{U}_N^{n,*}$, the convective matrix $\mathbf{C}_N(\mathbf{U}_N^{n,*})$ admits the following affine decomposition

$$\mathbf{C}_N(\mathbf{U}_N^{n,*}) = \mathbf{V}^T \mathbf{C}(\mathbf{V}\mathbf{U}_N^{n,*})\mathbf{V} = \sum_{j=1}^{N_v} (\mathbf{U}_N^{n,*})_j \; \mathbf{V}^T \mathbf{C}(\mathbf{V}_j)\mathbf{V},$$

where $\mathbf{V}_j$ denotes the j-th column of $\mathbf{V}$ and $(\mathbf{U}_N^*)_j$ the j-th component of $\mathbf{U}_N^*$. On the other hand, the matrices $\mathbf{S}_N(\mathbf{U}_N^{n,*})$, $\mathbf{M}_N(\mathbf{U}_N^{n,*})$ resulting form the SUPG stabilization do not admit a similar decomposition. Their assembly thus scales with the dimension N_h of the full-order model (3.76), preventing the efficient resolution of the ROM (3.79). To address this computational bottleneck, we approximate them by MDEIM.

3.7.5 System approximation

The MDEIM approximation of the matrix $\mathbf{S}(\mathbf{V}\mathbf{U}_N^{n,*})$ resulting form the SUPG stabilization is given by

$$\mathbf{S}(\mathbf{V}\mathbf{U}_N^{n,*}) \;\approx\; \mathbf{S}_m(\mathbf{V}_{\mathcal{R}}\mathbf{U}_N^{n,*}) = \mathbf{\Phi}^s\boldsymbol{\theta}_q^s(\mathbf{V}_{\mathcal{R}}\mathbf{U}_N^{n,*}), \tag{3.80}$$

where

1. $\mathbf{\Phi}^s = [\mathrm{vec}(\mathbf{S}_1) \;\cdots\; \mathrm{vec}(\mathbf{S}_{M_s})] \in \mathbb{R}^{N_h^2 \times M_s}$ is a POD basis for a suitable subspace of
$$\mathcal{M}_{\mathbf{S}} = \left\{\mathrm{vec}(\mathbf{S}(\mathbf{V}\mathbf{U}_N^{n,*})) : n = 1, \dots, N_t\right\} \subset \mathbb{R}^{N_h^2};$$
2. the interpolation weights $\boldsymbol{\theta}^s(\mathbf{V}_{\mathcal{R}}\mathbf{U}_N^{n,*}) = \left(\theta_1^s(\mathbf{V}_{\mathcal{R}}\mathbf{U}_N^{n,*}), \dots, \theta_{M_s}^s(\mathbf{V}_{\mathcal{R}}\mathbf{U}_N^{n,*})\right) \in \mathbb{R}^{M_s}$ are given by
$$\boldsymbol{\theta}^s(\mathbf{V}_{\mathcal{R}}\mathbf{U}_N^{n,*}) = (\mathbf{\Phi}_{\mathcal{I}}^s)^{-1}\left(\mathbf{s}(\mathbf{V}_{\mathcal{R}}\mathbf{U}_N^{n,*})\right)_{\mathcal{I}},$$
with $\mathbf{s}(\cdot) = \mathrm{vec}(\mathbf{S}(\cdot))$;
3. the interpolation indices $\mathcal{I}$ are selected by DEIM as detailed in Section 3.3;
4. $\mathcal{R} \subset \{1, \dots, N_h\}$ denotes the set of degrees of freedom associated to the reduced elements and $\mathbf{V}_{\mathcal{R}}$ the restriction of $\mathbf{V}$ to the rows $\mathcal{R}$.

Then, we approximate $\mathbf{S}_N(\mathbf{U}_N^{n,*})$ by

$$\mathbf{S}_N^m(\mathbf{U}_N^{n,*}) = \mathbf{V}^T\mathbf{S}_m(\mathbf{V}_{\mathcal{R}}\mathbf{U}_N^{n,*})\mathbf{V} = \sum_{q=1}^{M_s} \theta_q^s(\mathbf{V}_{\mathcal{R}}\mathbf{U}_N^{n,*})\; \mathbf{V}^T\mathbf{S}_q\mathbf{V}, \tag{3.81}$$

where $\mathbf{S}_N^q = \mathbf{V}^T\mathbf{S}_q\mathbf{V} \in \mathbb{R}^{N\times N}$, $q = 1, \dots, M_s$, are precomputable matrices of small dimension. Similarly, we employ MDEIM to obtain an approximate affine decomposition for the matrix $\mathbf{M}$,

$$\mathbf{M}(\mathbf{V}\mathbf{U}_N^{n,*}) \;\approx\; \mathbf{M}_m(\mathbf{V}_{\mathcal{R}}\mathbf{U}_N^{n,*}) = \mathbf{\Phi}^m\boldsymbol{\theta}_q^m(\mathbf{V}_{\mathcal{R}}\mathbf{U}_N^{n,*}),$$

yielding

$$\mathbf{M}_N^m(\mathbf{U}_N^{n,*}) = \mathbf{V}^T\mathbf{M}_m(\mathbf{V}_{\mathcal{R}}\mathbf{U}_N^{n,*})\mathbf{V} = \sum_{q=1}^{M_m} \theta_q^m(\mathbf{V}_{\mathcal{R}}\mathbf{U}_N^{n,*})\; \mathbf{V}^T\mathbf{M}_q\mathbf{V}. \tag{3.82}$$

The ROM with system approximation (or hyper-ROM) reads: given $\mathbf{U}_{N,m}^n, \dots, \mathbf{U}_{N,m}^{n+1-\sigma}$, for $n \geq \sigma - 1$ find $\mathbf{U}_{N,m}^{n+1} \in \mathbb{R}^N$ such that

$$\left(\frac{\alpha}{\Delta t}\mathbf{M}_N^m(\mathbf{U}_{N,m}^{n,*}) + \mathbf{A}_N + \mathbf{C}_N(\mathbf{U}_{N,m}^{n,*}) + \mathbf{S}_N^m(\mathbf{U}_{N,m}^{n,*})\right)\mathbf{U}_{N,m}^{n+1} = \frac{1}{\Delta t}\mathbf{M}_N^m(\mathbf{U}_{N,m}^{n,*})\mathbf{U}_{N,m}^{n,\sigma} + \mathbf{g}_N(t_{n+1}). \tag{3.83}$$

The offline construction of the reduced model consists of three main steps, as detailed in Algorithm 3.6. First, we solve (3.76), collect snapshots $\{\mathbf{U}_h^n\}_{n=1}^{N_t}$ and compress them by POD to generate a basis $\mathbf{V}$. Here, POD is computed with respect to the discrete $[H^1(\Omega)]^d$ norm for the velocity and $L^2(\Omega)$ norm for the pressure. Then, we solve (3.79), collect snapshots $\{\mathbf{S}(\mathbf{V}\mathbf{U}_N^{n,*})\}_{n=1}^{N_t}$, $\{\mathbf{M}(\mathbf{V}\mathbf{U}_N^{n,*})\}_{n=1}^{N_t}$ and compress them by POD to generate the bases $\mathbf{\Phi}^s$ and $\mathbf{\Phi}^m$, respectively. Finally, we run MDEIM on $\mathbf{\Phi}^s$, $\mathbf{\Phi}^m$ to obtain affine approximations of $\mathbf{S}$ and $\mathbf{M}$, respectively. The resulting matrices are then projected onto $\mathbf{V}$ to generate the reduced ones.

Remark 3.11. As in the GNAT framework [CBMF11, CFCA13] (see also [WH15]), the snapshots generating the reduced bases $\mathbf{\Phi}^s$ and $\mathbf{\Phi}^m$ are collected by solving the reduced model (3.79), rather than the high-fidelity one. •

1: Set $\mathbf{V} = [\,]$, $\mathbf{\Phi}^s = [\,]$, $\mathbf{\Phi}^m = [\,]$
2: (1) collect and compress solution snapshots
3: (1a) solve (3.73) to obtain solution snapshots
4: $\mathbf{\Lambda}_u = [\mathbf{U}_h^1, \dots, \mathbf{U}_h^{N_t}]$
5: (1b) generate basis $\mathbf{V}$
6: $\mathbf{V} = \mathrm{POD}(\mathbf{\Lambda}_u, \varepsilon_u)$
7: (2) collect and compress system snapshots
8: (2a) solve (3.79) to obtain system snapshots
9: $\mathbf{\Lambda}_s = [\mathrm{vec}(\mathbf{S}(\mathbf{V}\mathbf{U}_N^{1,*})), \dots, \mathrm{vec}(\mathbf{S}(\mathbf{V}\mathbf{U}_N^{N_t,*}))]$
10: $\mathbf{\Lambda}_m = [\mathrm{vec}(\mathbf{M}(\mathbf{V}\mathbf{U}_N^{1,*})), \dots, \mathrm{vec}(\mathbf{M}(\mathbf{V}\mathbf{U}_N^{N_t,*}))]$
11: (2b) generate system bases
12: $\mathbf{\Phi}^s = \mathrm{POD}(\mathbf{\Lambda}_s, \varepsilon_s)$
13: $\mathbf{\Phi}^m = \mathrm{POD}(\mathbf{\Lambda}_m, \varepsilon_m)$
14: (3) perform MDEIM on $\mathbf{\Phi}^s$ and $\mathbf{\Phi}^m$
15: generate a common reduced mesh
16: project resulting matrices and vectors on the reduced basis $\mathbf{V}$

Algorithm 3.6 Offline procedure for the construction of the hyper-ROM (3.83).

3.7.6 The parametrized case

We now consider the case where some parameters $\boldsymbol{\mu} \in \mathcal{D}$ affect the physical configuration of the system via the boundary conditions or the kinematic viscosity ν. Then, the high-fidelity problem (3.76) becomes: given $\mathbf{U}_h^n, \dots, \mathbf{U}_h^{n+1-\sigma}$, for $n \geq \sigma - 1$ find $\mathbf{U}_h^{n+1} \in \mathbb{R}^{N_h}$ such that

$$\left(\frac{\alpha}{\Delta t}\mathbf{M}(\mathbf{U}_h^{n,*};\boldsymbol{\mu}) + \mathbf{A}(\boldsymbol{\mu}) + \mathbf{C}(\mathbf{U}_h^{n,*}) + \mathbf{S}(\mathbf{U}_h^{n,*};\boldsymbol{\mu})\right)\mathbf{U}_h^{n+1} = \frac{1}{\Delta t}\mathbf{M}(\mathbf{U}_h^{n,*};\boldsymbol{\mu})\mathbf{U}_h^{n,\sigma} + \mathbf{g}(t_{n+1};\boldsymbol{\mu}). \quad (3.84)$$

In this context, we build a *global* ROM – that is, a ROM trained at multiple points $\{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^K\}$ in the parameter space – by combining the strategy presented in Sect. 3.6.1

with Algorithm 3.6. First, for each training input $\boldsymbol{\mu}^k$, $k = 1, \dots, K$, we solve (3.84), collect snapshots $\{\mathbf{U}_h^n(\boldsymbol{\mu}^k)\}_{n=1}^{N_t}$ and compress them by POD to generate a global basis $\mathbf{V}$. Then, for each $\boldsymbol{\mu}^k$, we solve the parametrized counterpart of (3.79), collect system snapshots

$$\{\mathbf{S}(\mathbf{V}\mathbf{U}_N^{n,*}; \boldsymbol{\mu}^k)\}_{n=1}^{N_t}, \qquad \{\mathbf{M}(\mathbf{V}\mathbf{U}_N^{n,*}; \boldsymbol{\mu}^k)\}_{n=1}^{N_t}$$

and compress them by POD to build global bases $\boldsymbol{\Phi}^s$ and $\boldsymbol{\Phi}^m$. Finally, we run MDEIM on $\boldsymbol{\Phi}^s$, $\boldsymbol{\Phi}^m$ to obtain affine approximations of $\mathbf{S}$ and $\mathbf{M}$, respectively. The details of the procedure are reported in Algorithm 3.7.

Set $\mathbf{V} = [\,]$, $\boldsymbol{\Phi}^s = [\,]$, $\boldsymbol{\Phi}^m = [\,]$
(1) collect and compress solution snapshots
    for $k = 1 : K$
        (1a) solve (3.76) for $\boldsymbol{\mu} = \boldsymbol{\mu}^k$ to obtain solution snapshots
            $\boldsymbol{\Lambda}_u^k = [\mathbf{U}_h^1(\boldsymbol{\mu}^k), \dots, \mathbf{U}_h^{N_t}(\boldsymbol{\mu}^k)]$
        (1b) compress local snapshot matrix and generate global one
            $\widetilde{\boldsymbol{\Lambda}}_u^k = \mathrm{POD}(\boldsymbol{\Lambda}_u^k, \varepsilon_u^{\mathrm{loc}}), \quad \widetilde{\boldsymbol{\Lambda}}_u = [\mathbf{V} \;\; \widetilde{\boldsymbol{\Lambda}}_u^k]$
        (1c) extract global solution basis
            $\mathbf{V} \;\; = \mathrm{POD}(\widetilde{\boldsymbol{\Lambda}}_u, \varepsilon_u)$
    end for
(2) collect and compress system snapshots
    for $k = 1 : K$
        (2a) solve (3.79) for $\boldsymbol{\mu} = \boldsymbol{\mu}^k$ to obtain system snapshots
            $\boldsymbol{\Lambda}_s^k = [\mathrm{vec}(\mathbf{S}(\mathbf{V}\mathbf{U}_N^{1,*}; \boldsymbol{\mu}^k)), \dots, \mathrm{vec}(\mathbf{S}(\mathbf{V}\mathbf{U}_N^{N_t,*}; \boldsymbol{\mu}^k))]$
            $\boldsymbol{\Lambda}_m^k = [\mathrm{vec}(\mathbf{M}(\mathbf{V}\mathbf{U}_N^{1,*}; \boldsymbol{\mu}^k)), \dots, \mathrm{vec}(\mathbf{M}(\mathbf{V}\mathbf{U}_N^{N_t,*}; \boldsymbol{\mu}^k))]$
        (2b) compress local snapshot matrices and generate global ones
            $\widetilde{\boldsymbol{\Lambda}}_s^k = \mathrm{POD}(\boldsymbol{\Lambda}_s^k, \varepsilon_s^{\mathrm{loc}}), \quad \widetilde{\boldsymbol{\Lambda}}_s = [\boldsymbol{\Phi}^s \;\; \widetilde{\boldsymbol{\Lambda}}_S^k]$
            $\widetilde{\boldsymbol{\Lambda}}_m^k = \mathrm{POD}(\boldsymbol{\Lambda}_f^k, \varepsilon_f^{\mathrm{loc}}), \quad \widetilde{\boldsymbol{\Lambda}}_m = [\boldsymbol{\Phi}^m \;\; \widetilde{\boldsymbol{\Lambda}}_m^k]$
        (2c) extract global system bases
            $\boldsymbol{\Phi}^s = \mathrm{POD}(\widetilde{\boldsymbol{\Lambda}}_s, \varepsilon_s),$
            $\boldsymbol{\Phi}^m = \mathrm{POD}(\widetilde{\boldsymbol{\Lambda}}_m, \varepsilon_m)$
    end for
(3) perform MDEIM on $\boldsymbol{\Phi}^s$ and $\boldsymbol{\Phi}^m$
    generate a common reduced mesh
    project resulting matrices and vectors on the reduced basis $\mathbf{V}$

Algorithm 3.7 Offline procedure for the construction of the ROM for the parametrized problem (3.84).

3.8 Flow past a cylinder: numerical results

We now apply the methodology developed in Sect. 3.7 to the model problem described in Sect. 3.2.2. Here we concentrate on the fluid problem, while Sect. 3.9 is devoted to

the reduction of the temperature equation. The setup follows [STD+96]: with respect to equations (3.59) and Fig. 3.2, we impose the following boundary conditions

$$\begin{aligned} \boldsymbol{v} &= \boldsymbol{h} && \text{on } \Gamma_d \times (0,T) \\ \boldsymbol{v} &= \boldsymbol{0} && \text{on } \Gamma_w \cup \Gamma_c \times (0,T) \\ -p\boldsymbol{n} + \nu(\nabla \boldsymbol{v})\boldsymbol{n} &= \boldsymbol{0} && \text{in } \Gamma_n \times (0,T), \end{aligned} \tag{3.85}$$

where the inflow velocity profile prescribed on Γ_d is given by

$$\boldsymbol{h}(\mathbf{x}) = \left(\frac{4U x_2 (H - x_2)}{H^2}, 0 \right)^T,$$

with $U = 2$. The initial condition $\boldsymbol{v}_0$ is given by the solution of the corresponding steady Stokes equations. All the numerical results are obtained using $\mathbb{P}_1$-$\mathbb{P}_1$ finite elements on an unstructured mesh made of 38 968 triangles and 19 903 vertices, leading to a full-order model of dimension $N_h = 58\,113$. Using a sparse direct solver to solve the linear system (3.76), each time step costs on average about 2.2 seconds[3]. While in a first test case we fix $1/\nu = 3750$ yielding a Reynolds number

$$\mathrm{Re} = \frac{\frac{2}{3} U \, 2r}{\nu} = 500,$$

we then consider $\mu = 1/\nu \in [150, 1500]$ as a parameter, yielding a Reynolds number $\mathrm{Re} \in [20, 200]$.

3.8.1 Efficient evaluation of drag and lift coefficients

We assess the accuracy of the ROM by monitoring the drag and lift coefficients on the cylinder, defined as

$$C_D(\boldsymbol{v}, p) = -\frac{1}{\frac{1}{2} V_\infty^2 \, 2r} \int_{\Gamma_c} (\boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}) \cdot \boldsymbol{v}_\infty \, d\Gamma, \tag{3.86}$$

$$C_L(\boldsymbol{v}, p) = \frac{1}{\frac{1}{2} V_\infty^2 \, 2r} \int_{\Gamma_c} (\boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n}) \cdot \boldsymbol{n}_\infty \, d\Gamma, \tag{3.87}$$

where r is radius of the cylinder, $\boldsymbol{v}_\infty$ is a unit vector directed as the incoming flow, $V_\infty = 2/3\,U$ is the average inflow velocity, while $\boldsymbol{n}_\infty$ is a unit vector orthogonal to $\boldsymbol{v}_\infty$. We denote by $C_{DI}^n = C_D(\boldsymbol{v}_h^n, p_h^n)$, $C_{LI}^n = C_L(\boldsymbol{v}_h^n, p_h^n)$ the drag and lift coefficients computed using the high-fidelity model, by C_{DII}^n, C_{LII}^n the ones predicted by the ROM without system approximation (3.79), and by C_{DIII}^n, C_{LIII}^n the ones predicted by the hyper-ROM (3.83).

In practice, to avoid integration over the boundary Γ_c, we compute C_{DI}^n and C_{LI}^n by evaluating the residual of (3.69) using as test functions suitable extensions

$$\hat{\boldsymbol{v}}_\infty \in V_h, \qquad \hat{\boldsymbol{v}}_\infty|_{\Gamma_c} = \boldsymbol{v}_\infty, \qquad \hat{\boldsymbol{v}}_\infty|_{\Gamma \setminus \Gamma_c} = \boldsymbol{0},$$

$$\hat{\boldsymbol{n}}_\infty \in V_h, \qquad \hat{\boldsymbol{n}}_\infty|_{\Gamma_c} = \boldsymbol{n}_\infty, \qquad \hat{\boldsymbol{n}}_\infty|_{\Gamma \setminus \Gamma_c} = \boldsymbol{0}$$

[3] All the CPU times reported in this section refer to computations performed on a workstation with a Intel Core i5-2400S CPU and 16 GB of RAM.

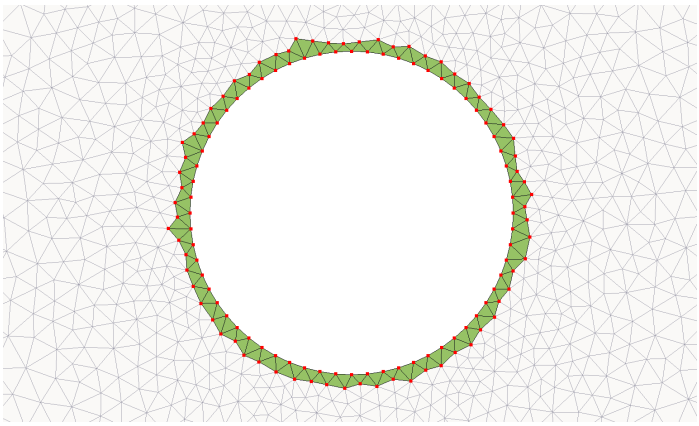


Fig. 3.12 Zoom of the post-processing mesh around the cylinder: red points represent the degrees of freedom $\mathcal{C}$, while green triangles are the corresponding elements. The figure refers to a coarser mesh than the one used for computations.

to the fluid domain of $\boldsymbol{v}_\infty$ and $\boldsymbol{n}_\infty$, respectively. See, e.g., [GLLS97, Ded07]. Specifically, for any vertex $\mathbf{N}_i$ of the computational mesh, we define the nodal values of $\hat{\boldsymbol{v}}_\infty$ and $\hat{\boldsymbol{n}}_\infty$ as

$$\hat{\boldsymbol{v}}_\infty(\mathbf{N}_i) = \begin{cases} \boldsymbol{v}_\infty(\mathbf{N}_i), & \text{if } \mathbf{N}_i \in \Gamma_c \\ \mathbf{0}, & \text{otherwise,} \end{cases} \qquad \hat{\boldsymbol{n}}_\infty(\mathbf{N}_i) = \begin{cases} \boldsymbol{n}_\infty(\mathbf{N}_i), & \text{if } \mathbf{N}_i \in \Gamma_c \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{3.88}$$

Then, thanks to the local support of the FE basis functions, given $(\boldsymbol{v}_h, p_h)$ computing the drag and lift coefficients only requires to assemble the residual of (3.69) on the mesh elements adjacent to Γ_c. This is crucial in order to efficiently compute online the drag and lift coefficients predicted by the reduced models. Indeed, relying on the *post-processing mesh* concept (see Fig. 3.12) introduced in [Car11, CFCA13], computing $C^n_{D,III}$ and $C^n_{L,III}$ only requires to:

1. offline (once and for all): define the set $\mathcal{C} \subset \{1, \dots, N_h\}$ of degrees of freedom associated to the elements adjacent to Γ_c and denote by $\mathbf{V}_\mathcal{C} \in \mathbb{R}^{|\mathcal{C}| \times N}$ the restriction of $\mathbf{V}$ to the rows $\mathcal{C}$;
2. online: given $\mathbf{U}_N = (\mathbf{v}_N, \mathbf{p}_N)^T \in \mathbb{R}^N$, compute its expansion $\mathbf{V}_\mathcal{C}\mathbf{U}_N$ and assemble the residual of (3.69) on the post-processing mesh.

As a result, the online operation count is $O(N|\mathcal{C}|)$, independent of N_h.

3.8.2 Reduced-order model for Re = 500

As a first test case, we assess the performances of the hyper-ROM in a purely reproductive setting, with Re = 500 fixed. In Fig. 3.13 we report the drag and lift coefficients obtained by solving the high-fidelity model with a BDF2 scheme, $T = 5$ and $\Delta t = 2.5 \cdot 10^{-3}$.

Periodic regime

After a short initial transient phase, the solution exhibits a periodic behavior, where a vortex shedding can be observed behind the obstacle. For the time being, we aim

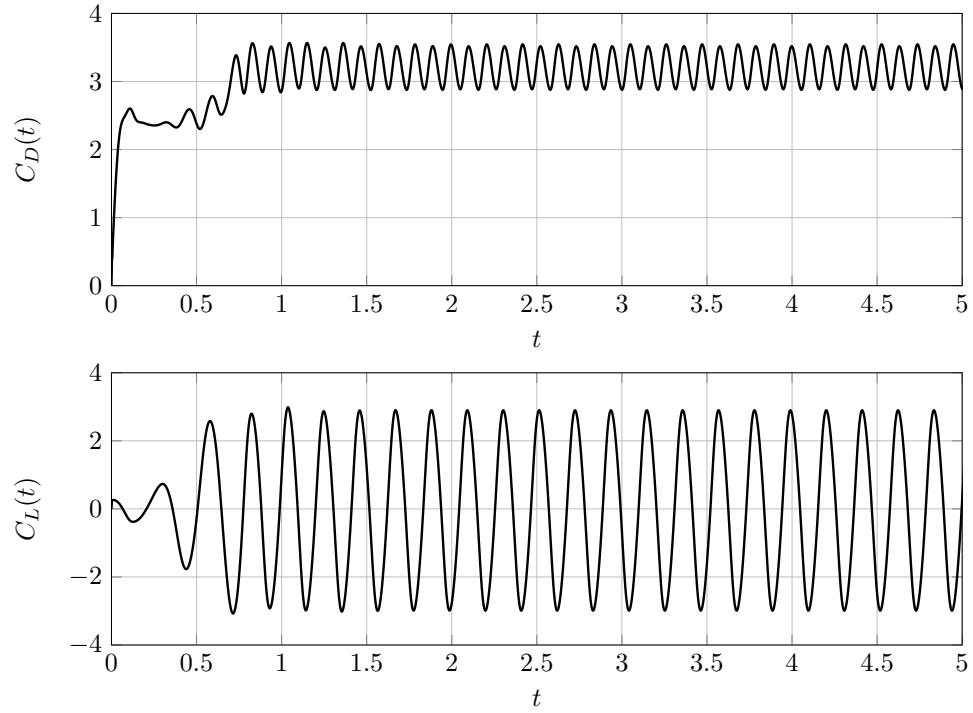


Fig. 3.13 Drag and lift coefficients obtained by solving the FOM for Re = 500.

to construct a ROM for the fully developed periodic regime. To this end, we collect solution snapshots in the time window $t \in (4, 4.5)$ and build the ROM without system approximation (3.79). Using a tolerance $\varepsilon_u = 5 \cdot 10^{-4}$, POD retains the first $N_v = 37$ velocity and $N_p = 24$ pressure modes, see Fig. 3.14. In step (2) of Algorithm 3.6, the ROM (3.79) is then solved in the time interval $t \in (4, 4.5]$ using as initial condition the solution of the high-fidelity problem at $t = 4$; during the resolution, we collect system snapshots every two time steps.

We then run MDEIM on the system snapshots: using a tolerance $\varepsilon_s = \varepsilon_m = 10^{-5}$, we obtain affine approximations of dimension $M_s = 59$ and $M_m = 64$. The decay of the singular values of the snapshot matrices $\mathbf{\Lambda}_s$ and $\mathbf{\Lambda}_m$ is reported in Fig. 3.14. As regards the computational costs, generating the POD modes from these snapshots matrices and selecting the MDEIM interpolation indices requires overall less than 40 seconds. The resulting reduced mesh contains about 2.1% of the original elements, see Fig. 3.15. Most of them concentrate in the wake region behind the body, which is consistent with the fact that the flow is separated in this region and is characterized by a strong vorticity.

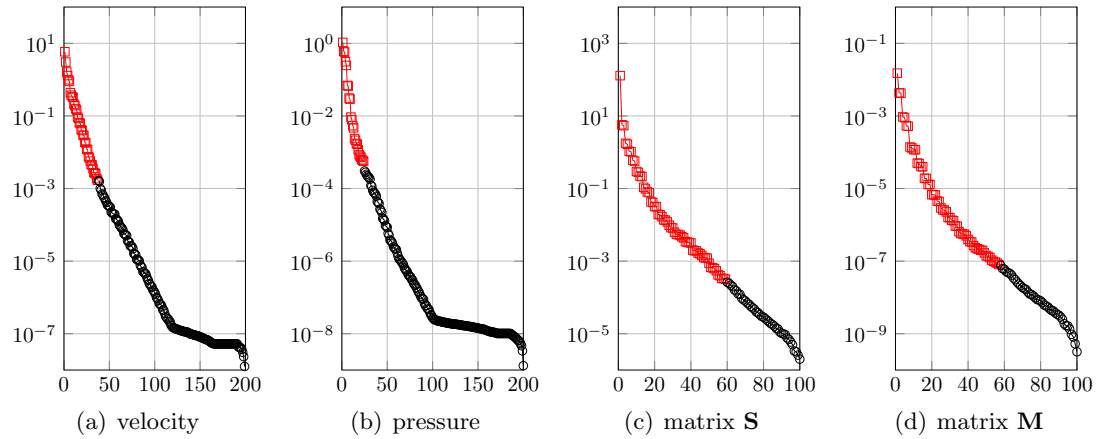
Figures 3.16 and 3.17 report the obtained time-histories (over the interval $t \in [4, 7]$) of the drag and lift coefficient for the high-fidelity model and the ROMs with and without system approximation (we often indicate the latter by HROM in the figures legend). A comparison of the lift-drag phase diagram for high-fidelity model and the hyper-ROM is also reported in Fig. 3.18.

To better investigate the effect of the chosen tolerances on the accuracy and efficiency of the resulting ROMs, we consider three different settings as reported in Table 3.3 (see also Fig. 3.19). We quantify the accuracy of the ROMs by computing the following

relative errors [CFCA13] in the drag and lift coefficients over the interval $t \in [4, 5]$

$$E_D^{I,III} = \frac{\frac{1}{N_t}\sum_{n=1}^{N_t} |C_{DI}^n - C_{DIII}^n|}{\max_n C_{DI}^n - \min_n C_{DI}^n}, \qquad E_L^{I,III} = \frac{\frac{1}{N_t}\sum_{n=1}^{N_t} |C_{LI}^n - C_{LIII}^n|}{\max_n C_{LI}^n - \min_n C_{LI}^n}.$$

The latter denote the errors between the high-fidelity solution and the one obtained by solving the hyper-ROM. Similarly, we denote by $E_D^{I,II}$ and $E_D^{I,II}$ the errors between the high-fidelity approximation and the ROM (3.79). Table 3.4 compares the error, CPU time and resulting speedup in these three different settings. Even using a very small number of solution and system basis functions, the reduced model delivers solutions with relative errors of roughly 1%.



(a) velocity (b) pressure (c) matrix **S** (d) matrix **M**

Fig. 3.14 Decay of the singular values of $\mathbf{\Lambda}_u$ (with velocity and pressure components), $\mathbf{\Lambda}_s$ and $\mathbf{\Lambda}_m$. Red squares correspond to the retained modes, while black circles correspond to the discarded ones.

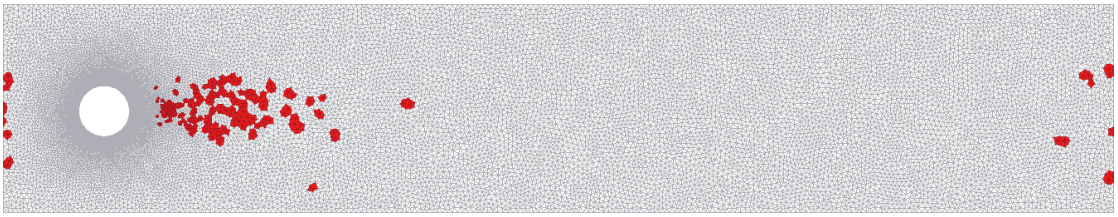


Fig. 3.15 Reduced mesh for Re = 500.

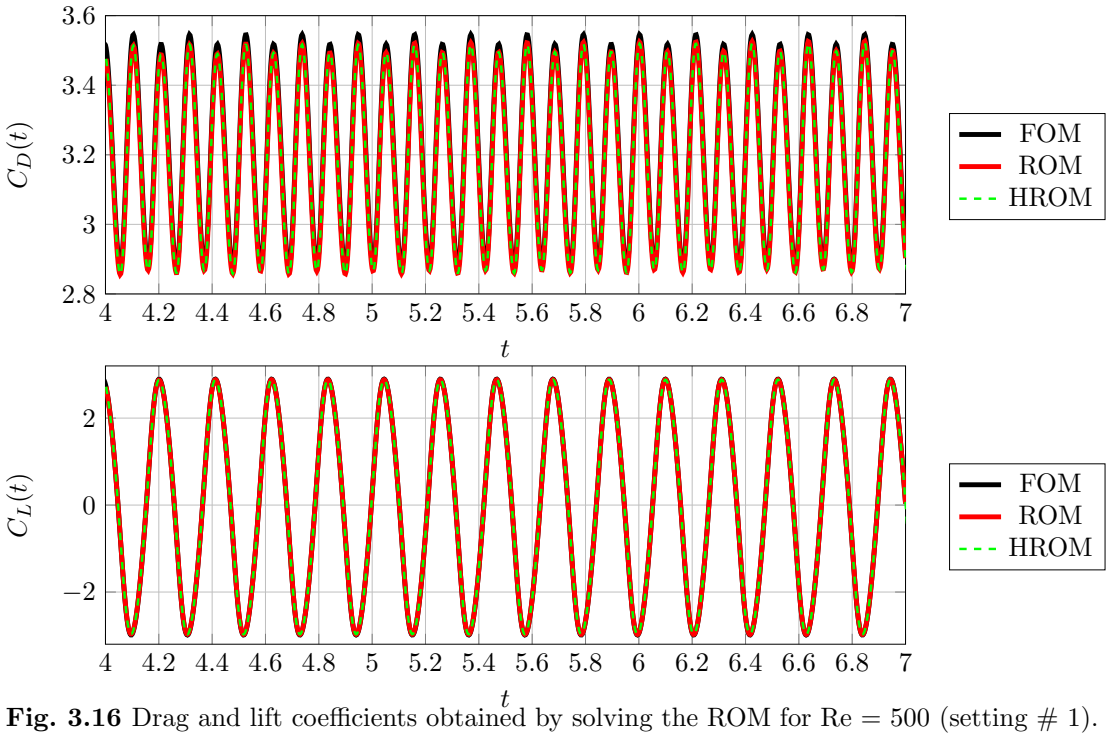


Fig. 3.16 Drag and lift coefficients obtained by solving the ROM for Re = 500 (setting # 1).

Table 3.3 Settings description: we report the tolerances used for POD and the resulting dimension of the state and system bases.

Setting	ε_u	ε_s, ε_m	$N = N_v + N_p$	M_s	M_m	Fraction of elements (%)
# 1	5e-4	1e-5	61	59	64	2.1
# 2	5e-3	1e-4	38	30	30	1.22
# 3	1e-2	1e-3	30	16	18	0.8

Table 3.4 Errors in the drag and lift coefficients, CPU time per time step and speedup for the three different settings described in Table 3.3.

Setting	$E_D^{I,II}(\%)$	$E_D^{I,III}(\%)$	$E_L^{I,II}(\%)$	$E_L^{I,III}(\%)$	CPU time	Speedup
# 1	3.62	3.62	0.18	0.18	35 ms	62
# 2	3.66	3.74	0.18	0.19	23 ms	96
# 3	3.78	4.31	0.20	0.29	17 ms	130

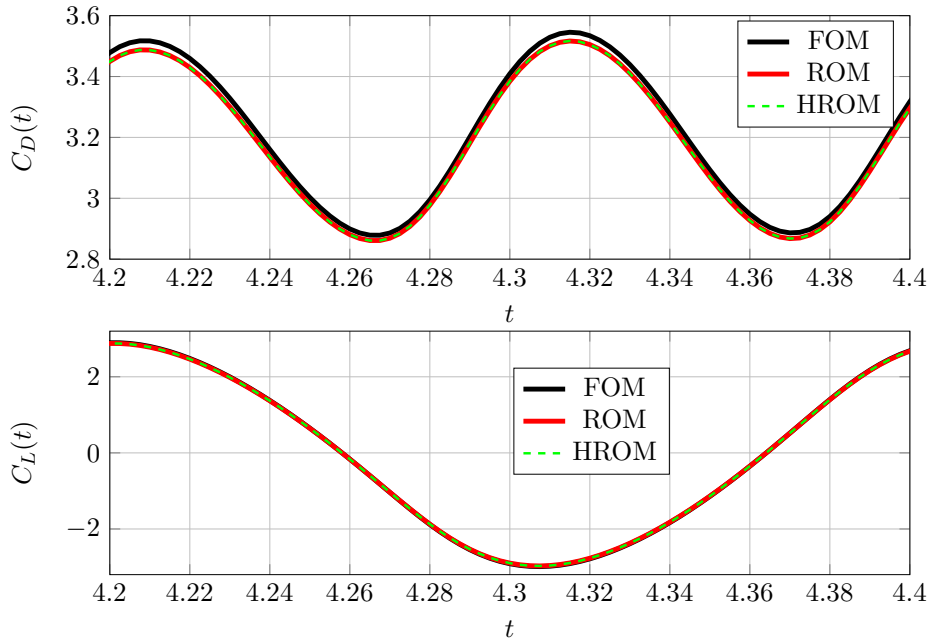


Fig. 3.17 Zoom of the drag and lift coefficients over $t \in [4.2, 4.4]$ obtained by solving the ROM for Re = 500.

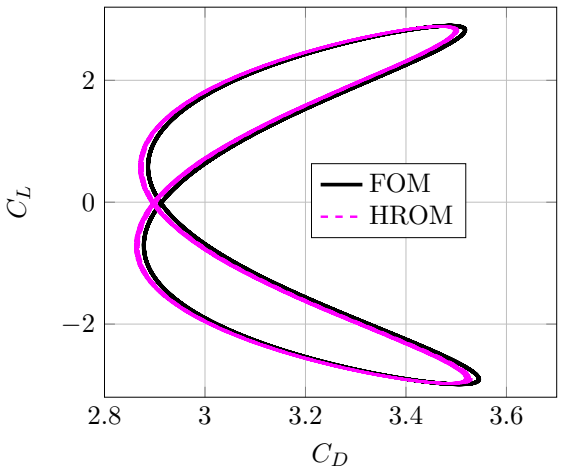
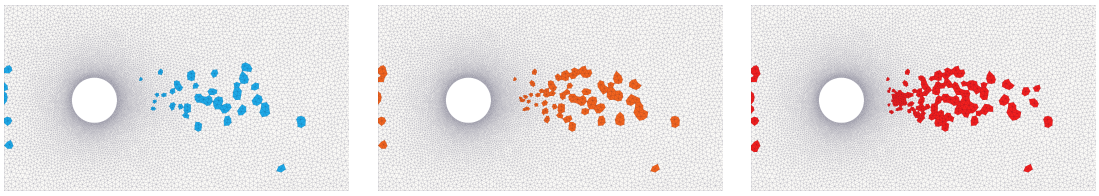


Fig. 3.18 Phase diagram of the drag and lift coefficients obtained by solving the FOM and the hyper-ROM for Re = 500.



(a) setting # 1 (b) setting # 2 (c) setting # 3

Fig. 3.19 Comparison between the reduced meshes obtained with the three settings described in Table 3.3. Zoom in the wake of the cylinder.

Transient and periodic phases

We now aim at generating a ROM able to reproduce not only the periodic regime, but also the initial transient phase. To this end, we train the ROM by collecting solution (every 4 time steps) and system (every 5 time steps) snapshots in the time interval $(0, 2.5]$. To determine the dimension of the POD basis for the state, a tolerance $\varepsilon_u = 10^{-2}$ employed. This gives $N_v = 151$, $N_p = 84$, resulting in a ROM of dimension equal to 0.4% of that of the original model. Similarly, the number of terms in the MDEIM approximation of the system matrices is $M_s = 180$ and $M_m = 184$, corresponding to $\varepsilon_s = \varepsilon_m = 5 \cdot 10^{-5}$. The resulting reduced mesh (see Fig. 3.20) contains the 6.4% of elements of the original one. Figure 3.21 reports the obtained time-histories of the drag and lift coefficient for the high-fidelity model and the ROMs with and without system approximation. The associated errors are given by

$$E_D^{I,II} = 0.79\% \qquad E_D^{I,III} = 0.86\%, \qquad E_L^{I,II} = 0.69\%, \qquad E_L^{I,III} = 0.69\%,$$

We further investigate the effects of system approximation by computing the errors for different values of $M_s = M_m$. The results reported in Figs. 3.22 and 3.23 indicate the following:

- the error $E_D^{I,III}$ (resp. $E_D^{I,III}$) correctly converge to $E_D^{I,II}$ ($E_L^{I,II}$) as M_s, M_m increase;
- more than about 50 terms in the MDEIM approximation are required to have a relative error below 10%;
- even when using a very poor system approximation ($M_s = M_m = 40$), the ROM remains stable (also for longer integration times).

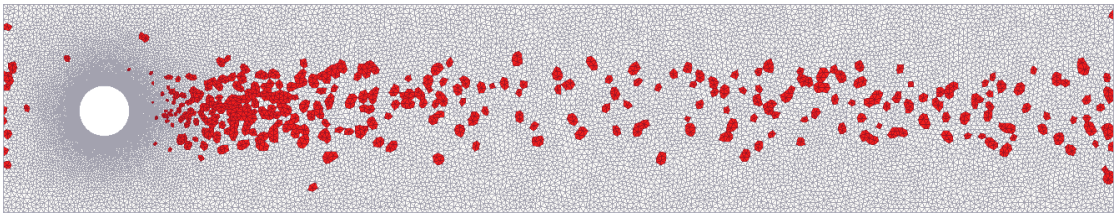


Fig. 3.20 Reduced mesh for Re = 500 (transitory regime).

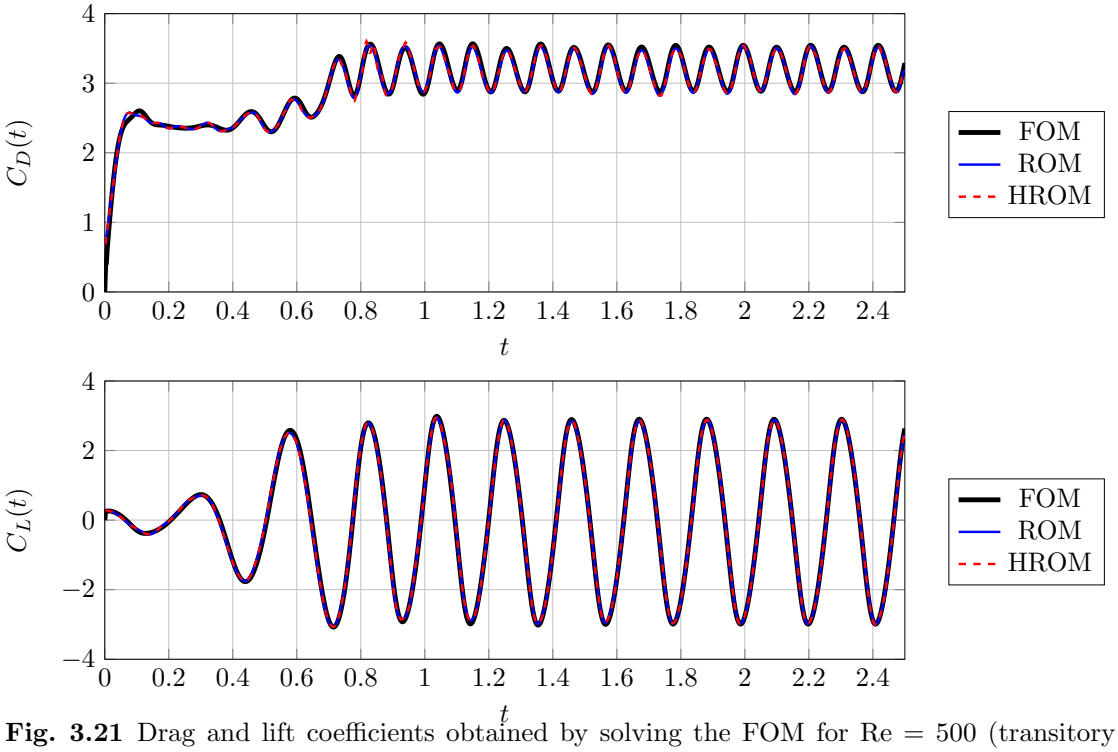


Fig. 3.21 Drag and lift coefficients obtained by solving the FOM for Re = 500 (transitory regime).

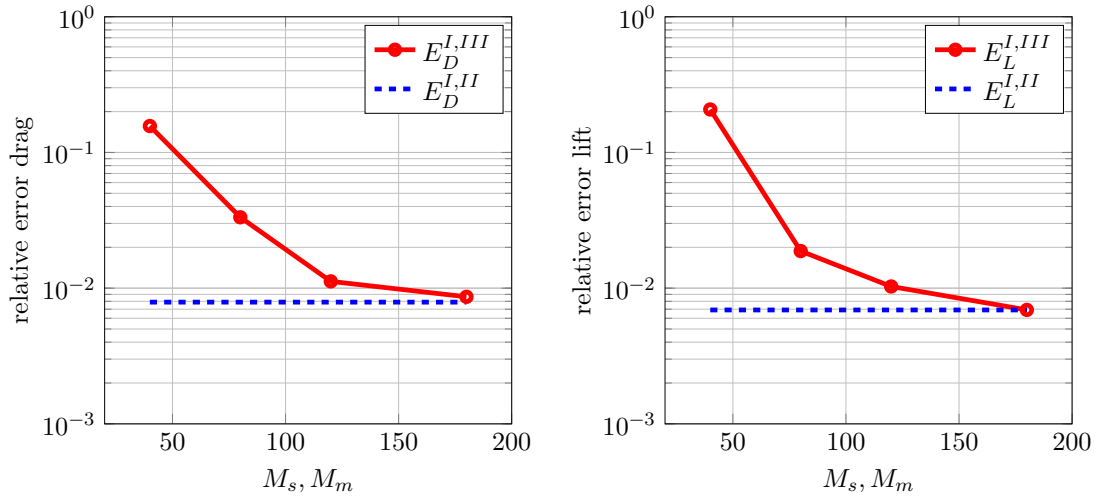


Fig. 3.22 Root mean square error on the drag and lift coefficients for $M_s = M_m = \{40, 80, 20, 180\}$ (with fixed state approximation).

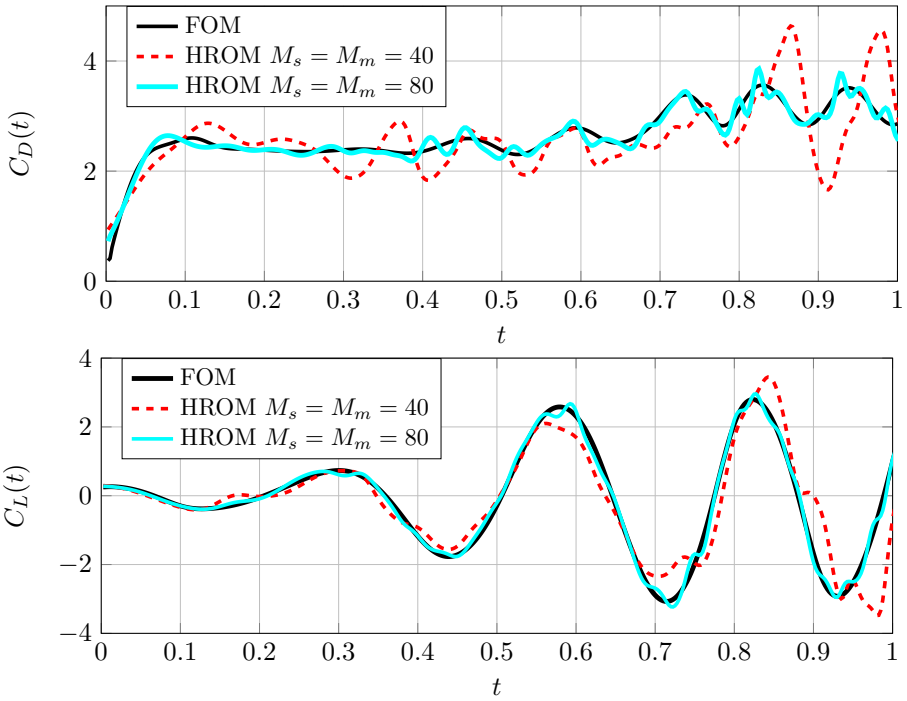


Fig. 3.23 Time-history of the drag (top) and lift (bottom) coefficients in the interval $t \in [0, 1]$ for the FOM and the ROM with $M_s = M_m = \{40, 80\}$.

3.8.3 The parametrized case

We now consider a parametrized scenario with $\mu = 1/\nu \in [150, 1500]$, yielding a Reynolds number $\mathrm{Re} \in [20, 200]$. The time horizon of interest is $t \in [0, 5.5]$ and the time step used for computations is $\Delta t = 10^{-2}$. The initial condition is given by the solution of the corresponding steady Stokes problem.

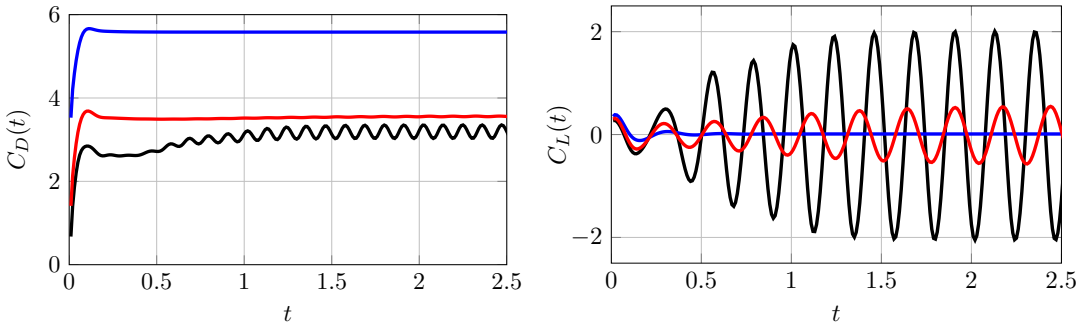
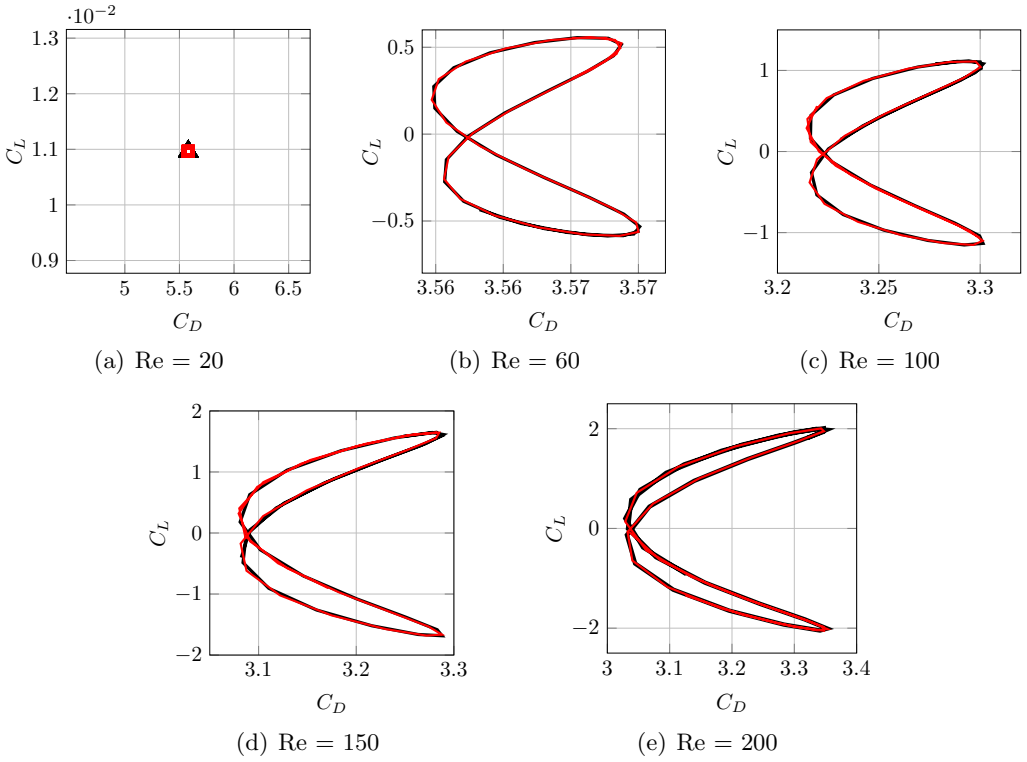


Fig. 3.24 Time-history of the drag (left) and lift (right) coefficients in the interval $t \in [0, 2.5]$ obtained by the FOM at some training parameters.

The reduced model is generated using the procedure detailed in Algorithm 3.7. In particular, we consider $K = 5$ training points corresponding to $\mathrm{Re} = \{20, 60, 100, 150, 200\}$. Figure 3.24 reports the drag and lift time-history for some training points. Note that the responses are significantly different from one another. Our goal is to build a ROM able to accurately predict the behavior of the flow in the fully developed regime for a

varying Reynolds number. To this end, in step (1) of Algorithm 3.7 solution snapshots are collected every time step in the time interval $[5, 5.5]$. A POD basis of dimension $N_v = N_p = 150$ is then built for both velocity and pressure snapshots. A comparison of the lift-drag phase diagram at the training parameters for the high-fidelity and reduced models (prior to system approximation) is reported in Fig. 3.25.

In step (2) of the algorithm we collect system snapshots every 6 times steps in the time interval $[0, 4]$. Using a tolerance of 10^{-4} for POD, we end up with MDEIM approximations of size $M_s = 170$ and $M_m = 176$. The resulting reduced mesh is made of 2 013 elements, corresponding to 5.2% of the original ones. We first asses the accuracy of the resulting hyper-ROM at the training inputs. Table 3.5 reports the relative errors on the drag ($E_D^{II,III}$) and lift ($E_L^{II,III}$) coefficients between the ROM with and without system approximation. Overall, the error is always below 2.2%.



(a) Re = 20 (b) Re = 60 (c) Re = 100

(d) Re = 150 (e) Re = 200

Fig. 3.25 Phase diagram (for $t \in [5, 5.5]$) of the drag and lift coefficients at the training parameters for the FOM (black) and the ROM (red) without system approximation.

Table 3.5 Relative errors at the training parameters in the drag and lift coefficients predicted by the ROM with and without system approximation.

	Re = 20	Re = 60	Re = 100	Re = 150	Re = 200
$E_D^{II,III}$ (%)	0.01	0.42	1.64	1.06	1.14
$E_L^{II,III}$ (%)	0.04	0.29	2.09	2.15	1.84

Then, we consider 5 test parameters different from the training ones, corresponding

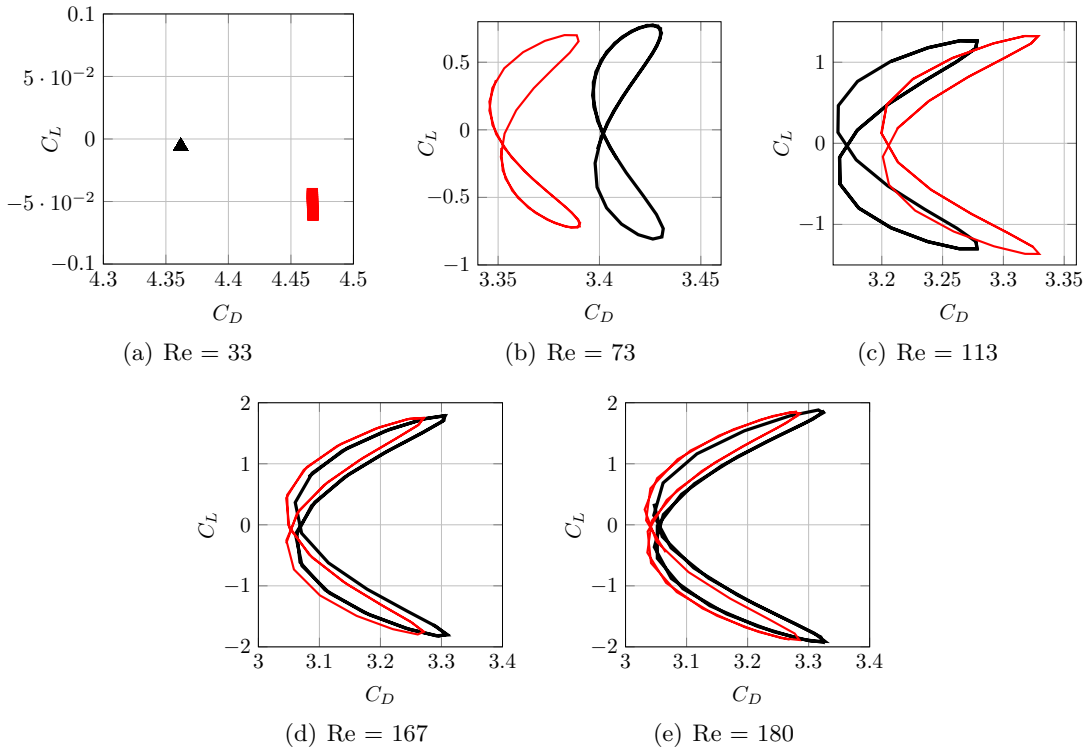
to $\mathrm{Re} = \{33, 73, 113, 167, 180\}$. We first investigate the accuracy of the state reduction by comparing the lift-drag phase diagram (for $t \in [5, 5.5]$) at the test parameters for the high-fidelity and reduced models (prior to system approximation), see Fig. 3.26. To better quantify the discrepancy between the two models, we compute the following relative errors (see Table 3.6):

$$\mathrm{Max}_D^{I,II} = \frac{|\max_n C_{DI}^n - \max_n C_{DII}^n|}{|\max_n C_{DI}^n|}, \qquad \mathrm{Min}_D^{I,II} = \frac{|\min_n C_{DI}^n - \min_n C_{DII}^n|}{|\min_n C_{DI}^n|},$$

$$\mathrm{Max}_L^{I,II} = \frac{|\max_n C_{LI}^n - \max_n C_{LII}^n|}{|\max_n C_{LI}^n|}, \qquad \mathrm{Min}_L^{I,II} = \frac{|\min_n C_{LI}^n - \min_n C_{LII}^n|}{|\min_n C_{LI}^n|}.$$

The errors in the drag coefficient never exceed 2.5%, while the ones in the lift reach 10% for $\mathrm{Re} = 73$. For $\mathrm{Re} = 33$ the errors in the lift are not reported, because the small value of C_D ($\approx 10^{-2}$) makes $\mathrm{Max}_L^{I,II}$ and $\mathrm{Min}_L^{I,II}$ very poor error indicators.

The accuracy of system approximation is monitored by the relative errors on the drag ($E_D^{II,III}$) and lift ($E_L^{II,III}$) coefficients predicted by the ROMs with and without system approximation. The results are reported in Table 3.7. In all cases, the hyper-ROM reproduces the time history of the drag coefficient (computed using the ROM without system approximation) with less than 4% discrepancy. Moreover, the hyper-ROM generates a speedup greater than 20 over the high-dimensional model, as it takes only 0.1 s per time step.



(a) Re = 33 (b) Re = 73 (c) Re = 113

(d) Re = 167 (e) Re = 180

Fig. 3.26 Phase diagram (for $t \in [5, 5.5]$) of the drag and lift coefficients at the test parameters for the FOM (black) and the ROM (red) without system approximation.

Table 3.6 Relative errors at the test parameters in the drag and lift coefficients predicted by the FOM and the ROM without system approximation.

	Re = 33	Re = 73	Re = 113	Re = 167	Re = 180
$\text{Max}_L^{I,II}$ (%)		9.73	4.66	1.97	1.43
$\text{Min}_L^{I,II}$ (%)		10.38	4.85	1.49	1.93
$\text{Max}_D^{I,II}$ (%)	2.43	1.20	1.56	1.14	1.24
$\text{Min}_D^{I,II}$ (%)	2.41	1.50	1.13	0.48	0.49

Table 3.7 Relative errors at the test parameters in the drag and lift coefficients predicted by the ROMs with and without system approximation.

	Re = 33	Re = 73	Re = 113	Re = 167	Re = 180
$E_D^{II,III}$ (%)	0.42	1.17	1.48	1.16	1.21
$E_L^{II,III}$ (%)	3.90	1.18	1.88	2.17	2.09

3.9 Heat transfer past a cylinder: numerical results

We now apply the methodology developed in Sect. 3.6 to problem (3.8). We consider as full-order model for problem (3.8) the finite element discretization (3.9) with $\boldsymbol{v}_h(t;\boldsymbol{\mu})$ replaced by $\boldsymbol{v}_{N,m}(t;\boldsymbol{\mu})$. The latter denotes the velocity field solution of the hyper-ROM for the Navier-Stokes equations. Moreover, we employ the BDF1 (i.e. backward Euler) method with a fixed time step $\Delta t = 10^{-2}$ to advance in time. As anticipated in Sect. 3.2.2, we consider as parameters the inverse of the fluid viscosity $\mu_1 = 1/\nu \in [150, 1500]$, the temperature $\mu_2 \in [3, 7]$ imposed on the cylinder and the thermal diffusivity $\mu_3 \in [10^{-2}, 10^{-3}]$.

We run Algorithm 3.5 to build the ROM using a training set of $K = 10$ parameter values selected by LHS sampling. At each step, we first solve the hyper-ROM for the

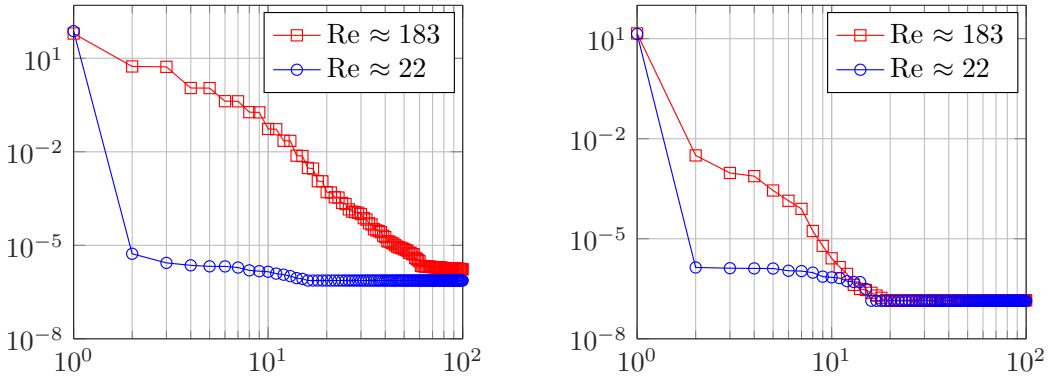


Fig. 3.27 First 100 singular values of the local snapshots matrices $\boldsymbol{\Lambda}_a^k$ (on the left) and $\boldsymbol{\Lambda}_g^k$ (on the right) for $k = 3, 10$, which correspond to Re ≈ 183 and Re ≈ 22 respectively.

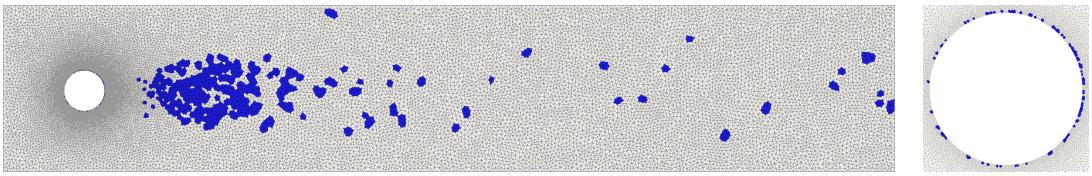


Fig. 3.28 Reduced mesh for the heat transfer problem (a zoom on the cylinder is shown on the right)

Navier-Stokes equations and the FOM for the temperature equation; then, we compress the local snapshots matrices by POD with $\varepsilon_u^{\text{loc}} = 10^{-3}$, $\varepsilon_a^{\text{loc}} = \varepsilon_m^{\text{loc}} = \varepsilon_g^{\text{loc}} = 10^{-4}$. The number of retained POD modes varies significantly depending of the value of the Reynolds number. We report in Fig. 3.27 the singular values of $\mathbf{\Lambda}_a^k$ and $\mathbf{\Lambda}_g^k$ for $\text{Re} \approx 22$ (corresponding to a stationary flow) and $\text{Re} \approx 183$ (corresponding to a vortex-shedding flow); while in the former case the first mode is sufficient to describe the entire dynamics, in the latter case the singular values decay more slowly and at least 10 modes have to be retained. After the snapshots collection we perform step (2) of Algorithm 3.5 obtaining a reduced model with $N = 150$, $M_a = 140$, $M_m = 140$ and $M_g = 30$. The resulting reduced mesh (see Fig. 3.28) is made of 1 563 elements, corresponding to about the 4% of the original ones; note that they concentrates in the wake and on the boundary of the cylinder, i.e. in the regions where the dynamics of the physical phenomena is more relevant.

In the online phase, given a parameter value, to obtain the temperature field we first compute $\mathbf{v}_{N,m}(t;\boldsymbol{\mu})$ and then solve the reduced order model for the temperature equation. Note that the online assembly of $\mathbf{A}(t;\boldsymbol{\mu})$, $\mathbf{M}(t;\boldsymbol{\mu})$ and $\mathbf{g}(t;\boldsymbol{\mu})$ on the reduced mesh would require the full-order expansion of the reduced transport field $\mathbf{v}_{N,m}(t;\boldsymbol{\mu})$, i.e. $\mathbf{V}_v\mathbf{v}_{N,m}(t;\boldsymbol{\mu})$; however, thanks to the local support of the FE basis functions, only its restriction to the nodes belonging to the reduced elements is actually needed.

We assess the accuracy of the hyper-ROM by monitoring the temperature at a point $\overline{\mathbf{x}} = (0.51, 0.26)$ in the wake of the cylinder. The comparison between this latter and the one predicted by the FE model for various parameter values (different from the training ones) is shown in Fig. 3.29. In Fig. 3.30, we also report the time-history of the relative $H^1(\Omega)$ error between the full and reduced solutions at the same testing parameters. Moreover, some of the corresponding temperature fields are shown in Fig 3.31. The hyper-ROM correctly reproduces the dynamics of the system in both the stationary and periodic regimes. Moreover, its solution takes about 0.03 seconds per time step, while each FOM time step takes about 0.9 seconds, thus delivering a speedup of about 30.

A schematic summary of the entire offline-online computational strategy is offered in Fig. 3.32. In the next chapter, we apply this reduction strategy to the simulation of blood flow in cerebral aneurysms and the simulation of solute dynamics in blood flow and arterial walls.

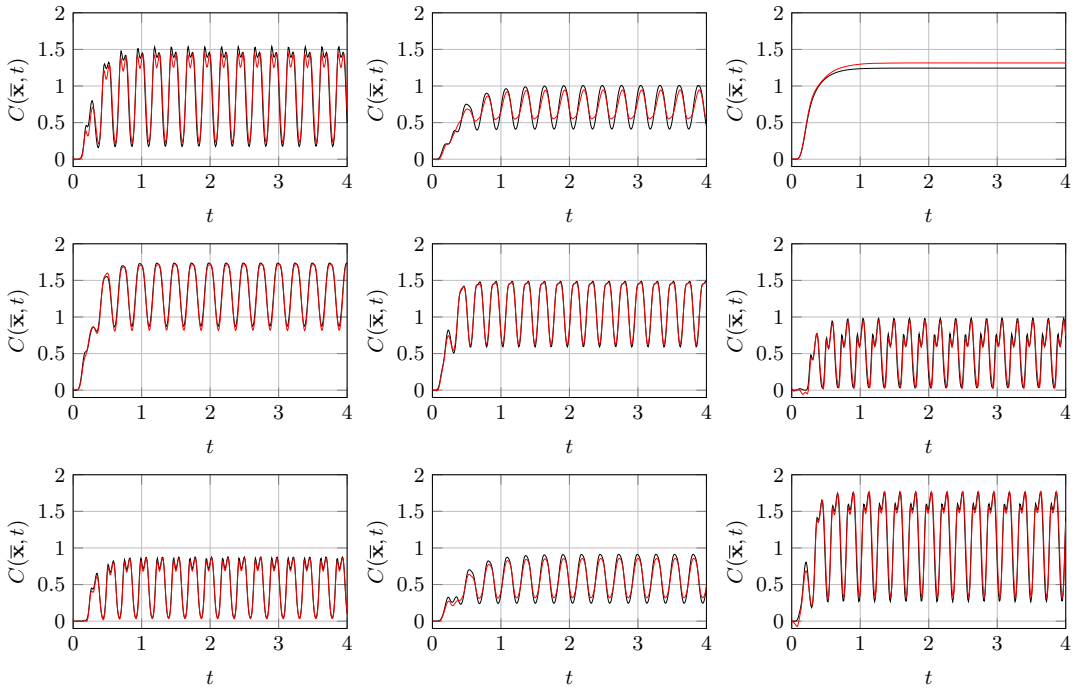


Fig. 3.29 Comparison between the temperature at point $\overline{\mathbf{x}} = (0.51, 0.26)$ obtained solving the high-fidelity (red line) and reduced (blue line) models for different testing parameter values.

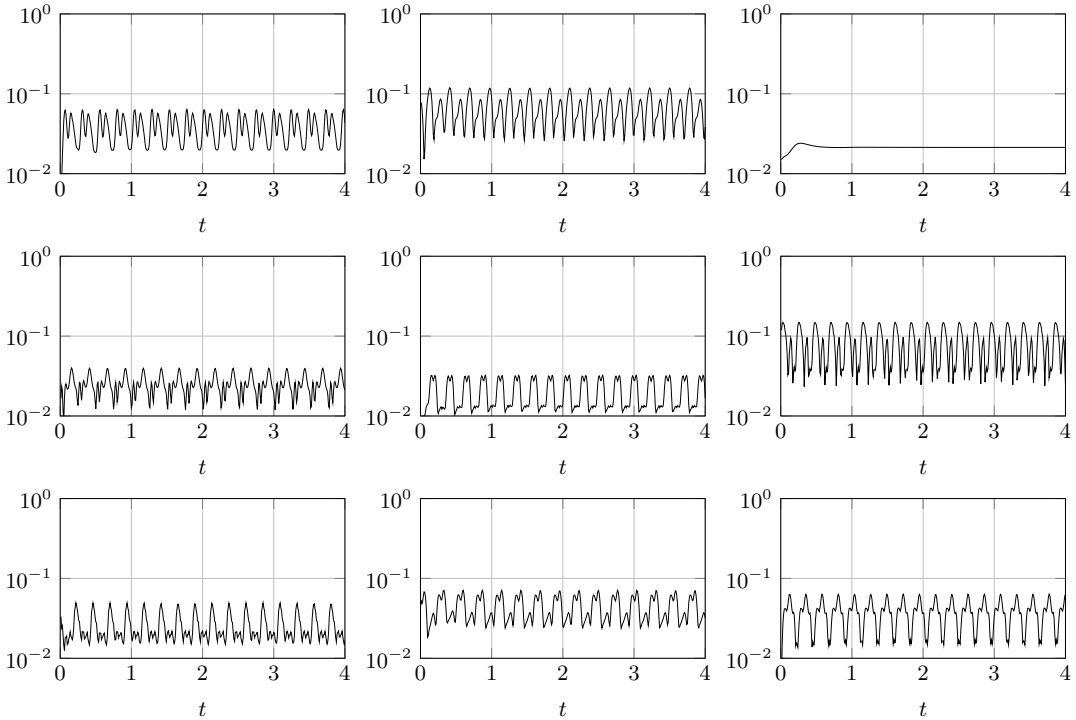


Fig. 3.30 Time-history of the relative error $\|C_h^n(\boldsymbol{\mu}) - C_{N,m}^n(\boldsymbol{\mu})\|_{H^1(\Omega)} / \|C_h^n(\boldsymbol{\mu})\|_{H^1(\Omega)}$ for the same testing parameters of Fig. 3.29.

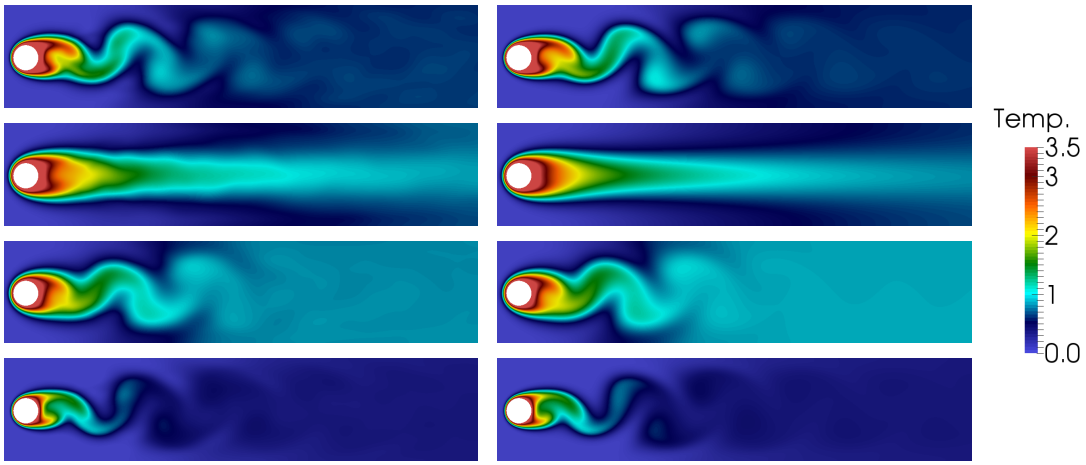


Fig. 3.31 Comparison between the temperature field at final time obtained by solving the ROM (on the left) and FOM (on the right) for some of the parameter configurations reported in Fig 3.29.

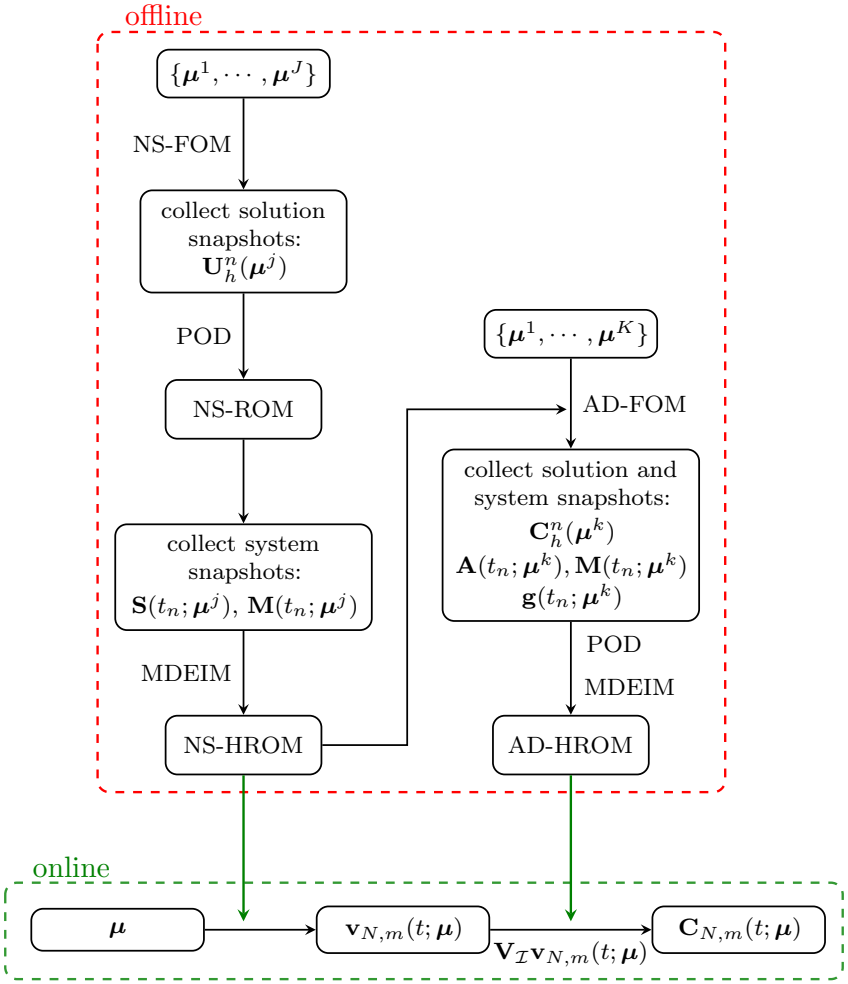


Fig. 3.32 Scheme of the offline and online phases for the coupled flow and heat transfer problem. Here NS-ROM and NS-HROM refer to (3.79) and (3.83), respectively. AD-HROM denotes instead the ROM (3.52) for the advection-diffusion equation.

4 Reduced-order models for blood flow and mass transport problems

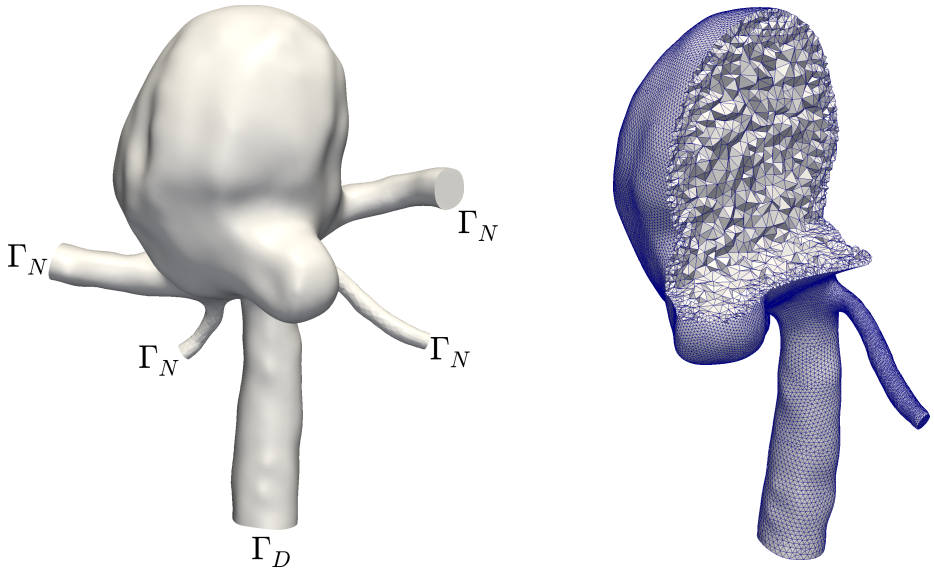
The reduction strategies developed in Chap. 3 are applied in this chapter to simulate first blood flow in a cerebral aneurysm and then oxygen transfer in a femoropopliteal bypass. In the former case, Navier-Stokes equations modeling the blood velocity and pressure fields are considered. In the latter case, the same equations are also coupled with a fluid-wall advection-diffusion model describing the dynamics of the oxygen concentration.

4.1 Blood flow in a cerebral aneurysm

A cerebral (also called intracranial) aneurysm is a bulge or ballooning resulting from the pathologic dilation of a blood vessel in the brain. The most common location for brain aneurysms is in the network of blood vessels at the base of the brain called the circle of Willis. Rupture of a cerebral aneurysm causes subarachnoid hemorrhage with potentially severe brain damages. Cerebral aneurysms are classified by size (from small to super-giant) and shape (saccular and non-saccular). Typically, saccular aneurysms (the most common ones) arise at a bifurcation or along a curve of the parent vessel. Classic treatments of saccular aneurysms are surgical clipping and endovascular coiling [Sch97, BSN06].

Hemodynamic factors, such as blood velocity, wall shear stress (WSS), pressure, particle residence time, and flow impingement, play an important role in the growth and rupture of cerebral aneurysms. In particular, high WSS is regarded as a major factor in the initiation and development of cerebral aneurysms, while aneurysm rupture is related to a low level of WSS and flow stagnation [TMV$^+$03, SOT$^+$04]. For these reasons, computational studies of blood flow dynamics inside models of saccular aneurysms are important to obtain quantitative criteria for the treatment of aneurysms, see e.g. [VMR$^+$08, RBAB$^+$08, BHZ$^+$10] and references therein.

Here, we aim at applying the reduction strategy developed in Sect. 3.7 to enable fast simulations of blood flow in a basal artery aneurysm for different flow conditions. To this end, we introduce two crucial assumptions: (i) blood is a Newtonian fluid, i.e. we assume that blood is a fluid with constant viscosity; (ii) the arterial wall is



(a) geometry and boundary conditions (b) computational mesh

Fig. 4.1 Geometry and (a cut of the) mesh of the basilar artery aneurysm. Average aneurysm radius is 0.4 cm, while the radius of the basilar artery at the inlet section is about 0.14 cm; see [AT12] for a detailed description of other geometrical features.

rigid. As such, blood flow dynamics is described by the Navier-Stokes equations. More complex models taking into account the effect of arterial compliance and/or the influence of non-Newtonian properties of the blood have been compared and discussed, e.g., in [VGFA06, FR09, BHZ+10, Tri14].

4.1.1 Physical model and finite element approximation

We consider the basal artery aneurysm shown in Fig. 4.1(a). The geometry has been supplied by the Aneurisk project [PVS+09, AT12] (case ID C0096), while the computational mesh have been generated using the Vascular Modeling Toolkit [APB+08] for centerlines extraction and Gmsh [GR09]. The resulting mesh (shown in Fig. 4.1(b)) is made of 89 048 vertices and 406 248 tetrahedral elements.

Blood dynamic viscosity and density are set to $\mu = 0.035$ P and $\rho = 1\,\mathrm{g\,cm^{-3}}$, respectively, yielding a kinematic viscosity $\nu = 0.035$ $\mathrm{cm^2 s^{-1}}$. The arterial wall Γ_w is considered to be a rigid body so that a no-slip condition on the fluid velocity at the wall is imposed, flow resistance at the outlet boundaries Γ_N is neglected, while a parabolic profile $\boldsymbol{v}_{\text{in}}$ is specified at the lumen inlet, yielding

$$\begin{aligned} \boldsymbol{\sigma}(\boldsymbol{v}, p)\boldsymbol{n} &= \boldsymbol{0} && \text{on } \Gamma_N \\ \boldsymbol{v} &= \boldsymbol{0} && \text{on } \Gamma_w \\ \boldsymbol{v} &= k\boldsymbol{v}_{\text{in}}\, Q(t, \boldsymbol{\mu}) && \text{on } \Gamma_D. \end{aligned} \tag{4.1}$$

The parametrization of the inlet flow rate profile $Q(t, \boldsymbol{\mu})$ has been obtained by interpolating with radial basis functions a base profile $Q(t, \boldsymbol{0})$ taken from [BWP+15], and then treating some of the interpolated values as parameters. In particular, we considered a set

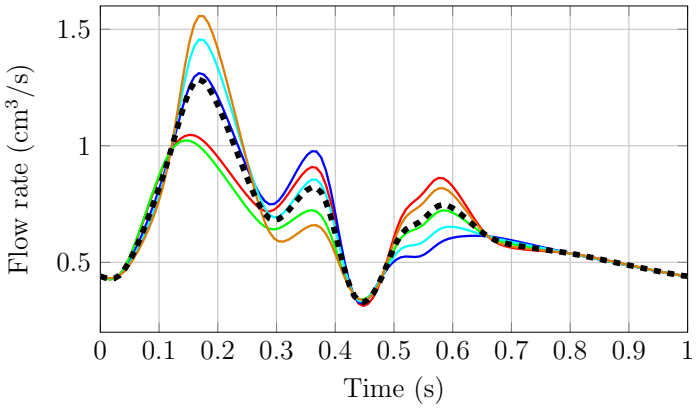


Fig. 4.2 Inlet flow rate $Q(t, \boldsymbol{\mu})$ during the heart cycle for different parameter values; the black dashed curve corresponds to the base profile $Q(t, \mathbf{0})$.

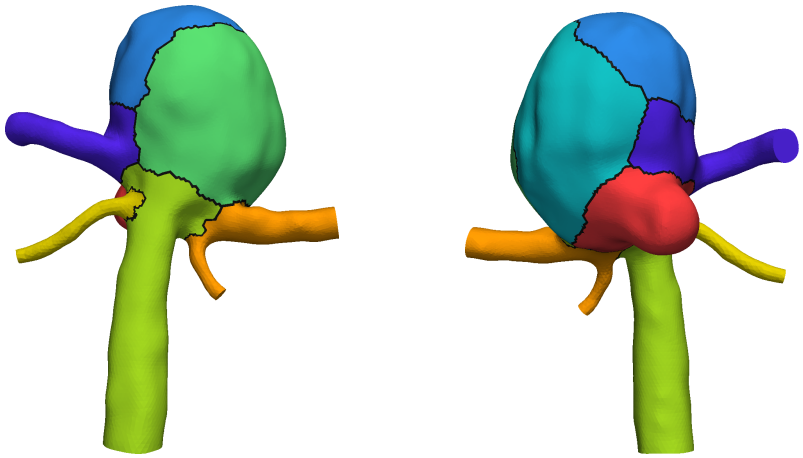


Fig. 4.3 Domain decomposition in 8 subdomains (overlap elements in black).

of three parameters $\boldsymbol{\mu} \in \mathcal{D} = [-25, 25] \times [-25, 25] \times [-25, 25]$ such that the flow rate at $t_1 = 0.16$, $t_2 = 0.38$ and $t_3 = 0.55$ admits variations up to 25% of the reference value. A comparison between some flow rate profiles corresponding to different parameter values is shown in Fig. 4.2. The scaling factor k in (4.1) is such that

$$\int_{\Gamma_D} k\, \boldsymbol{v}_{\text{in}} \cdot \boldsymbol{n}\, d\Gamma = 1.$$

As regards the high-fidelity discretization, we employ the SUPG-BDF semi-implicit scheme described in Sect. 3.7 with linear finite elements for both velocity and pressure variables, $\sigma = 2$ and $\Delta t = 0.008$ s. The dimension of the high-fidelity model is thus $N_h = 356\,192$. We simulate the blood flow for about two heartbeats ($T = 2$ s) starting from an initial condition obtained by solving the steady Stokes problem. We use the GMRES method with a two-level Additive Schwarz (AS) preconditioner to solve the linear system (3.76) arising at each time step. To build the AS preconditioner for the system matrix $\mathbf{K} = \frac{\alpha}{\Delta t}\mathbf{M} + \mathbf{A} + \mathbf{C}(\mathbf{U}_h^{n,*}) + \mathbf{S}(\mathbf{U}_h^{n,*})$, we first partition the domain Ω into overlapping subdomains Ω_j^δ, $j = 1 \ldots, J$, featuring an overlap of size $\delta = h$ (see Fig. 4.3). To this end, we use the Metis library [KK09] through the C-Mex interface

provided by the Meshpart toolbox [GT02]. We then build suitable restriction matrices $\mathrm{R}_j \in \mathbb{R}^{N_{h,j} \times N_h}$ so that the local matrices $\mathbf{K}_j = \mathrm{R}_j \mathbf{K} \mathrm{R}_j^T \in \mathbb{R}^{N_{h,j} \times N_{h,j}}$ correspond to the restriction of $\mathbf{K}$ to the subdomains Ω_j^δ. Finally we build a suitable coarse correction matrix $\mathbf{K}_0 = \mathrm{R}_0 \mathbf{K} \mathrm{R}_0^T \in \mathbb{R}^{4J \times 4J}$, whose restriction matrix $\mathrm{R}_0 \in \mathbb{R}^{4J \times N_h}$ is obtained by aggregation [Sal04]. The AS preconditioner is then defined as

$$\mathrm{P}_{AS}^{-1} = \sum_{j=0}^{J} \mathrm{R}_j^T \mathbf{K}_j^{-1} \mathrm{R}_j, \tag{4.2}$$

$\mathbf{K}_j^{-1}$ being the inverse of $\mathbf{K}_j$, here computed by means of an exact LU factorization; each local preconditioner is then applied in parallel. Using 8 subdomains, the iterative solver converges on average in 30 iterations up to a tolerance of 10^{-8} on the relative norm of the residual. Overall, each time step takes about 80 seconds[1].

As already mentioned, an important quantity of interest is the wall shear stress distribution $\boldsymbol{\tau}_w$ on Γ_w, which is defined as

$$\boldsymbol{\tau}_w(\mathbf{x}) = (2\mu \boldsymbol{\varepsilon}(\boldsymbol{v})\boldsymbol{n}) \cdot \boldsymbol{t} = 2\mu \left(\boldsymbol{\varepsilon}(\boldsymbol{v})\boldsymbol{n} - (\boldsymbol{\varepsilon}(\boldsymbol{v})\boldsymbol{n} \cdot \boldsymbol{n})\boldsymbol{n}\right) \qquad \forall \mathbf{x} \in \Gamma_w, \tag{4.3}$$

where $\boldsymbol{n}$ and $\boldsymbol{t}$ are the (outer) normal and tangential unit vectors at the wall, respectively, while $\boldsymbol{\varepsilon}(\boldsymbol{v})$ is the strain tensor introduced in Sect. 3.7. For linear finite elements, we compute a nodal reconstruction of the WSS by a recovery technique based on patch averaging [Ver13]. Precisely, for any vertex $\mathbf{N}_i \in \Gamma_w$, we define the nodal values of $\boldsymbol{\tau}_w$ as

$$\boldsymbol{\tau}_w(\mathbf{N}_i) = \sum_{K \in P_{\mathbf{N}_i}} \frac{m(K)}{m(P_{\mathbf{N}_i})} \boldsymbol{\tau}_w|_K, \tag{4.4}$$

where $P_{\mathbf{N}_i}$ denotes the patch of faces K sharing the node $\mathbf{N}_i$, $m(A)$ denotes the surface area of any $A \subset \Gamma_w$, and

$$\boldsymbol{\tau}_w|_K = \mu \left((\nabla \boldsymbol{v}_h|_K + \nabla \boldsymbol{v}_h^T|_K)\boldsymbol{n}_K - ((\nabla \boldsymbol{v}_h|_K + \nabla \boldsymbol{v}_h^T|_K)\boldsymbol{n}_K \cdot \boldsymbol{n}_K)\boldsymbol{n}_K \right).$$

is obtained by an exact calculation on the computed FE solution. Since the same procedure is adopted to compute the WSS distribution predicted by the ROMs, in this case the post-processing mesh (see Sect. 3.8.1) consists of a layer of elements adjacent to the arterial wall Γ_w. In the following, we shall denote by $\tau_w = \|\boldsymbol{\tau}_w\|_2$ the WSS magnitude.

4.1.2 Reduced-order model

We generate a reduced-order model for the problem at hand using the procedure detailed in Algorithm 3.7. In particular, we consider $K = 6$ training input parameters selected by latin hypercube sampling. Our goal is to build a ROM able to accurately predict the behavior of the flow in the fully developed periodic regime. To this end, in step (1) of Algorithm 3.7 (FE velocity and pressure) solution snapshots are collected every two time steps in the time interval $[1, 2]$ s. Using a tolerance $\varepsilon_u = 10^{-3}$, POD retains the

[1] In this section, high-fidelity computations are performed on a node of the SuperB cluster at EPFL with two Intel Xeon X5675E@3.07GHz processors and 192 GB of RAM. Online computations are performed on a workstation with a Intel Core i5-2400S processor and 16 GB of RAM.

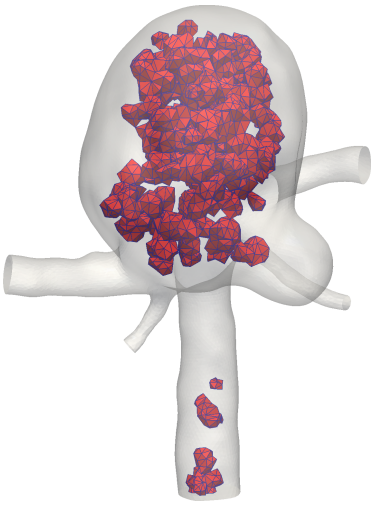


Fig. 4.4 Reduced mesh for the aneurysm problem.

first $N_v = 139$ velocity $N_p = 27$ pressure modes. The resulting ROM without system approximation takes on average 23 seconds per time step to be solved, the finite element arrays assembly being the most expensive operation.

In step (2) of the algorithm we collect system snapshots every 5 times steps in the time interval $[1, 2]$ s. Using a tolerance of 10^{-4} for POD, we end up with MDEIM approximations of size $M_s = 106$ and $M_m = 93$. The reduced mesh shown in Fig. 4.4 is made of 7 700 elements, corresponding to 1.9% of the original ones.

We measure the accuracy of the resulting hyper-ROM at 4 test inputs different from the training ones. Figures 4.5 and 4.6 reports the time-history of the $H^1(\Omega)$-error in the velocity fields predicted by the high-fidelity model and the hyper-ROM. Overall, the relative error is always below 6%. Figure 4.7 compares the time-history of the WSS magnitude predicted by the high-fidelity and reduced models at a probe located on the top surface of the aneurysm; a very good agreement is observed. Moreover, solving the hyper-ROM takes about 0.5 seconds per time step (using one core for the assembly on the reduced mesh), thus delivering a speedup of about 160 with respect to the high-fidelity model.

Figures 4.8, 4.9 show the blood flow velocity magnitude predicted by the hyper-ROM at different times for three parameter configurations. The corresponding WSS distribution are also shown in Figures 4.10, 4.11 and 4.12.

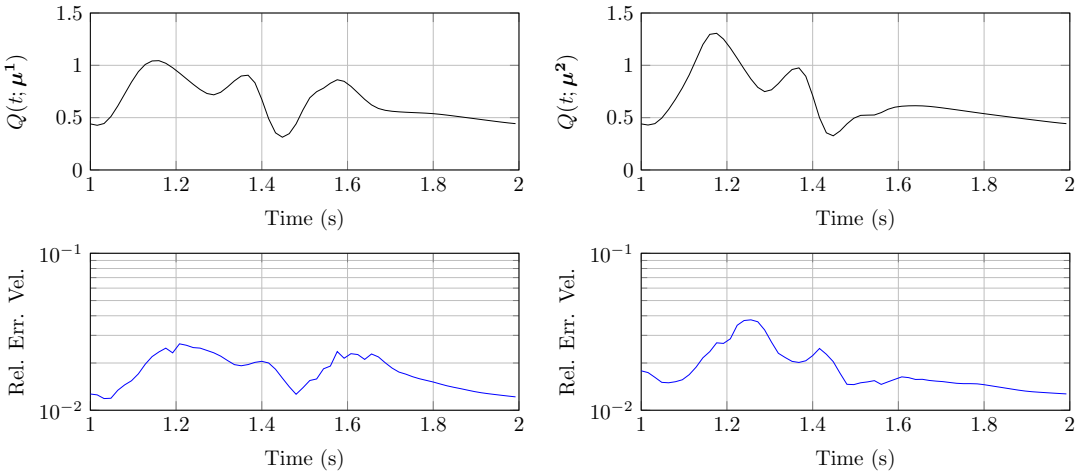


Fig. 4.5 Time-history of the relative $H^1(\Omega)$-error in the velocity fields predicted by the high-fidelity model and the hyper-ROM at two (out of four) test parameters (different from the training ones).

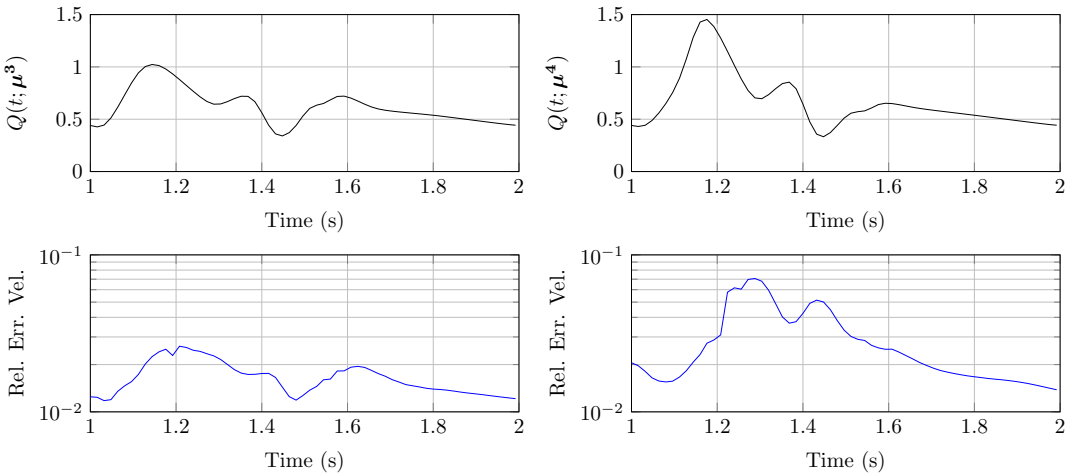
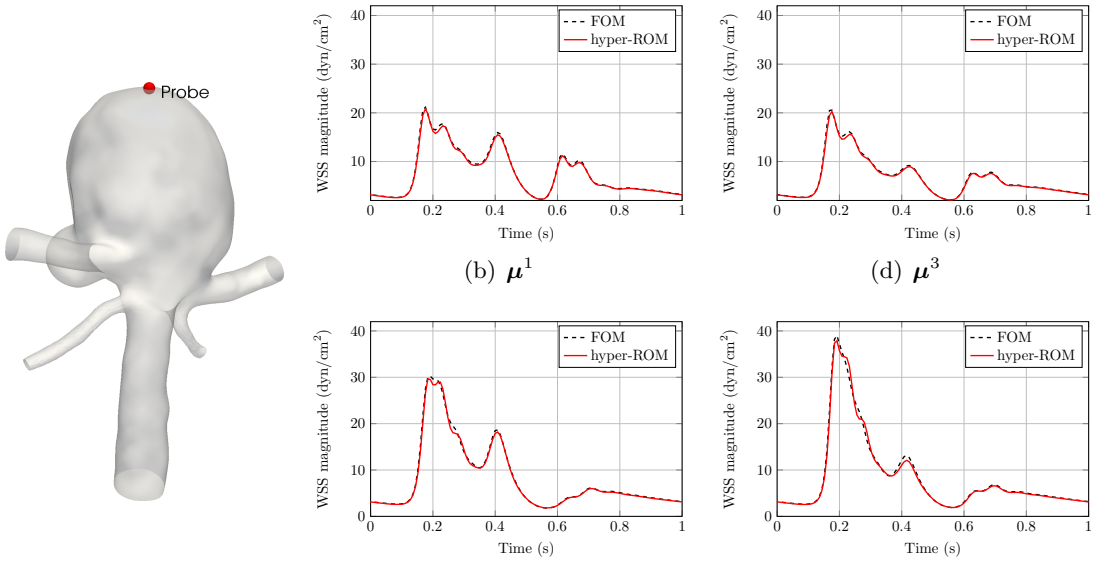
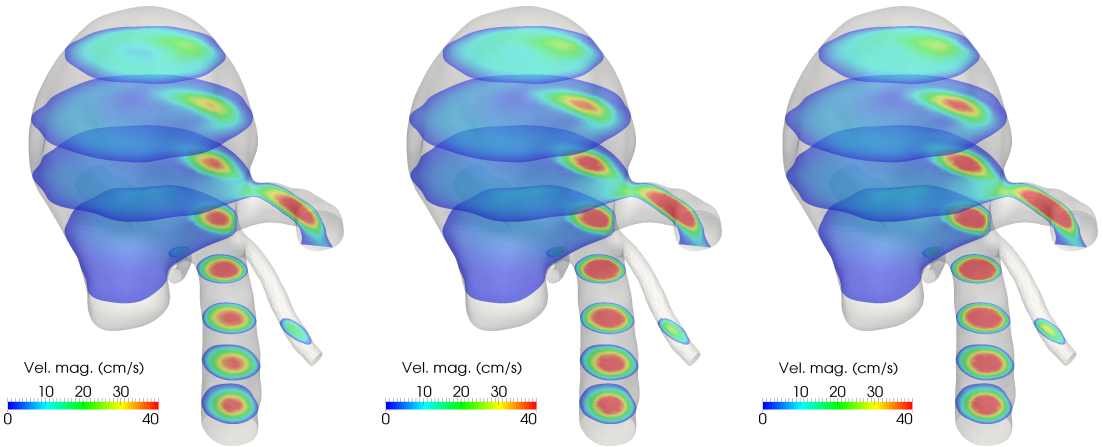


Fig. 4.6 Time-history of the relative $H^1(\Omega)$-error in the velocity fields predicted by the high-fidelity model and the hyper-ROM at two (out of four) test parameters (different from the training ones).



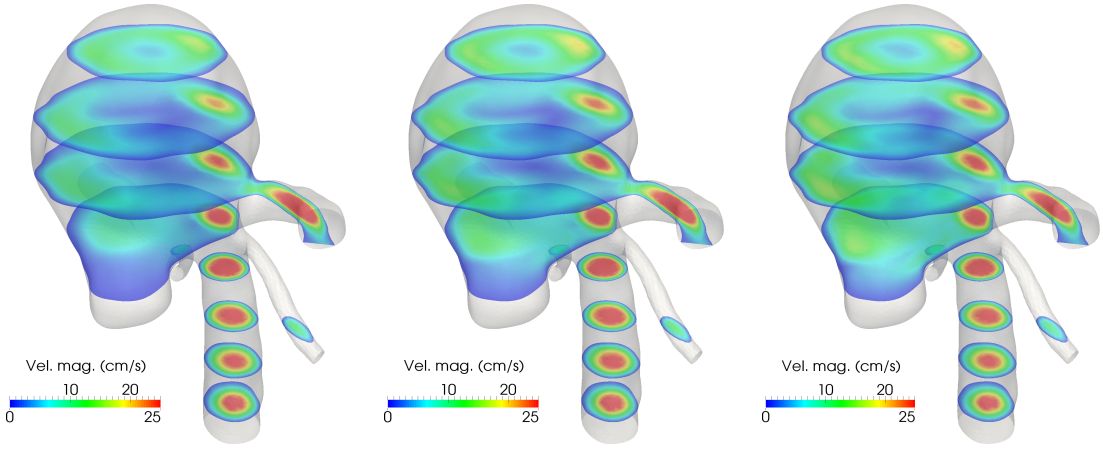
(a) Probe position (b) $\boldsymbol{\mu}^1$ (c) $\boldsymbol{\mu}^2$ (d) $\boldsymbol{\mu}^3$ (e) $\boldsymbol{\mu}^4$

Fig. 4.7 Time-history of the WSS magnitude at a probe predicted by the high-fidelity and reduced-order models. Subfigures (b)-(e) refer to the parameter configurations reported in Figs. 4.5 and 4.6.



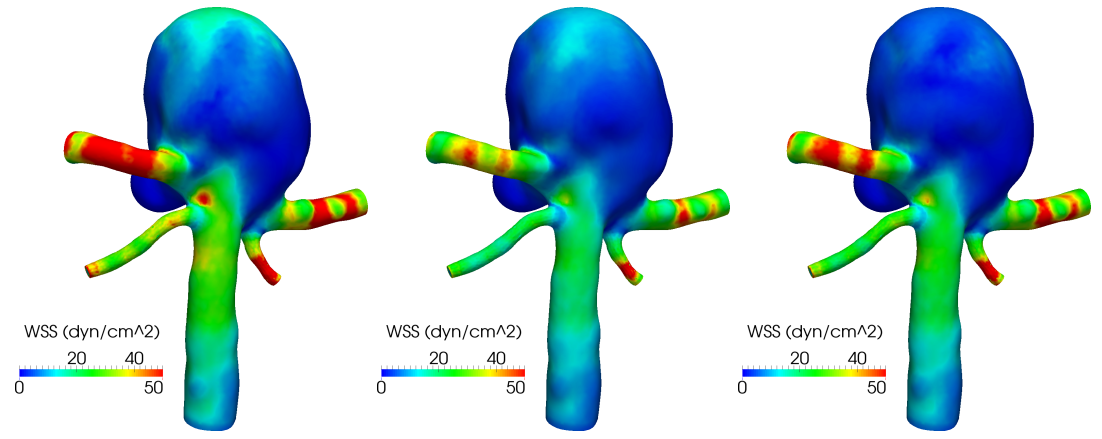
(a) $\boldsymbol{\mu}^1$ (b) $\boldsymbol{\mu}^2$ (c) $\boldsymbol{\mu}^4$

Fig. 4.8 Blood flow velocity on interior cuts at $t = 0.168$ s for different parameter configurations.



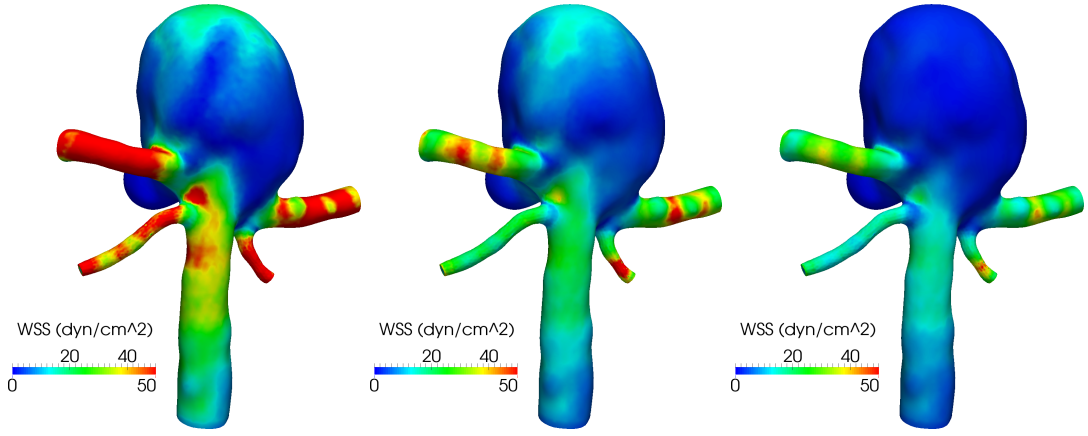
(a) $\boldsymbol{\mu}^1$ (b) $\boldsymbol{\mu}^2$ (c) $\boldsymbol{\mu}^4$

Fig. 4.9 Blood flow velocity on interior cuts at $t = 0.304$ s for different parameter configurations.



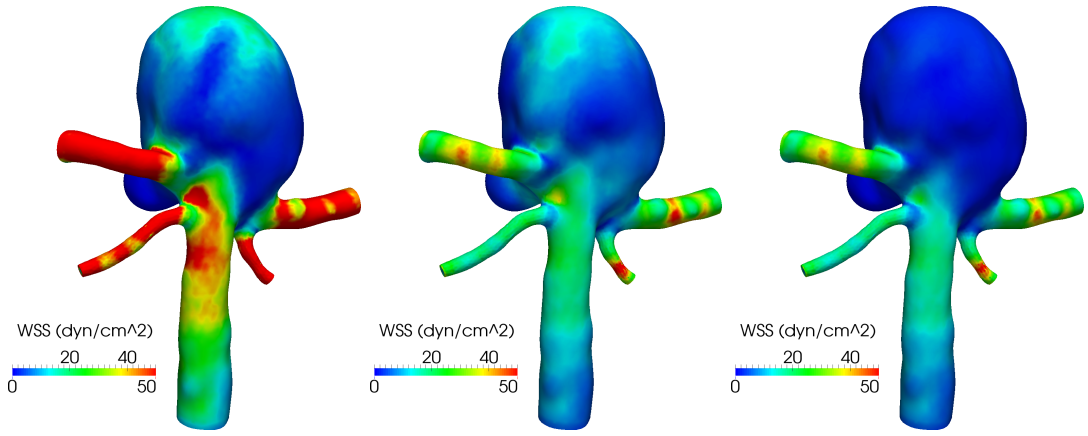
(a) $t = 0.168$ s (b) $t = 0.304$ s (c) $t = 0.584$ s

Fig. 4.10 WSS magnitude distribution predicted by the hyper-ROM for configuration $\boldsymbol{\mu}^1$.



(a) $t = 0.168$ s (b) $t = 0.304$ s (c) $t = 0.584$ s

Fig. 4.11 WSS magnitude distribution predicted by the hyper-ROM for configuration $\boldsymbol{\mu}^2$.



(a) $t = 0.168$ s (b) $t = 0.304$ s (c) $t = 0.584$ s

Fig. 4.12 WSS magnitude distribution predicted by the hyper-ROM for configuration $\boldsymbol{\mu}^4$.

4.2 Blood flow in a femoropopliteal bypass

Bypass grafting is a surgical procedure to create an alternate channel for blood flow, bypassing an obstructed or damaged portion of a vessel. In particular, peripheral vascular bypass is the surgical rerouting of blood flow around an obstructed artery that supplies blood to the legs and feet. This surgery is performed to restore blood flow in the arteries of people who have peripheral arterial disease. The latter consists in a widespread hardening and narrowing of the arteries (atherosclerosis) caused by the gradual build-up of plaques (fatty deposits). In critical cases, plaque accumulations obstruct arteries blocking the flow of blood, oxygen, and nutrients to the lower extremities.

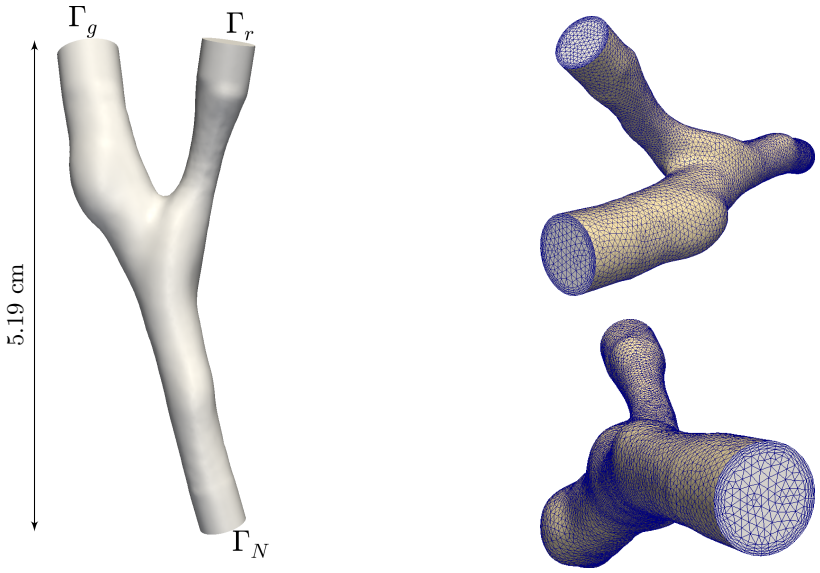
Depending on the severity of the stenosis, different medical treatments can be used. For instance, chemical drugs injection and/or stents implantation can be used to reduce the deposit of fat inside the arterial wall. However, in case of advanced peripheral arterial disease the stenosis is treated with surgical intervention that bypasses the obstruction using a graft. This latter reroutes blood from above the obstructed portion of an artery to another vessel below the obstruction. The graft can be either synthetic or a healthy segment of the patient's own saphenous vein (autogenous graft). Despite the success of bypass surgery, it is well known (see, e.g., [ODM12]) that arterial bypass grafts tend to fail after some years due to restenosis formation. In particular, the formation of intimal hyperplasia – a process in which the thickness of the inner wall of a vessel increases – is a major cause of bypass grafts failure. This phenomenon preferentially occurs at the distal end of the graft forming an end-to-side anastomosis with the host artery. It is often correlated with abnormally high or low values of shear stress, high values of its gradient, recirculation regions and graft deformation. Indeed, it has been observed that irregular hemodynamics patterns result in disturbed mass distributions, which may eventually contribute to the development of intimal thickening. For instance, local hypoxia is recognized as a source of disease initiation and acceleration [Lev95, CC09, STL00].

Here, we consider a patient-specific femoropopliteal bypass connecting the femoral artery to the popliteal one. Specifically, our domain of interest is the end-to-side anastomosis shown in Fig. 4.13(a) and our goal is to generate a reduced-order model for the simulation of oxygen transport in the femoropopliteal bypass. To this end, here we concentrate on the reduction of the fluid dynamics problem, while Sect. 4.3 is devoted to the reduction of the equations modeling the oxygen transport.

4.2.1 Physical model and finite element approximation

The geometry of the bypass was reconstructed through MRI scanner and then meshed as described in [MCG+12]. The resulting mesh (shown in Fig. 4.13(b)) is made of 65 893 vertices and 382 301 tetrahedral elements. The surface of the artery has been decomposed in several regions to impose boundary conditions. The boundary Γ_g denotes the inlet of the graft, while Γ_r is the inlet of the host artery. The latter is assumed to be partially occluded, with a residual flow still entering from Γ_r. Finally, Γ_N denotes the outflow boundary, while $\Gamma_w = \partial\Omega \setminus (\Gamma_g \cup \Gamma_r \cup \Gamma_N)$ is the arterial wall.

Blood dynamic viscosity and density are set to $\mu = 0.035$ P and $\rho = 1\,\mathrm{g\,cm^{-3}}$, respectively, yielding a kinematic viscosity $\nu = 0.035$ $\mathrm{cm^2 s^{-1}}$. The arterial wall is considered to be a rigid body so that a no-slip condition on the fluid velocity at Γ_w is imposed, flow resistance at the outlet boundaries Γ_N is neglected, while parabolic profiles



(a) geometry and boundary conditions (b) computational mesh

Fig. 4.13 Geometry and mesh (with flow extensions at the inlet and outlet boundaries) of the femoropolipteal bypass. The radius of the graft at the inlet section is about 0.34 cm.

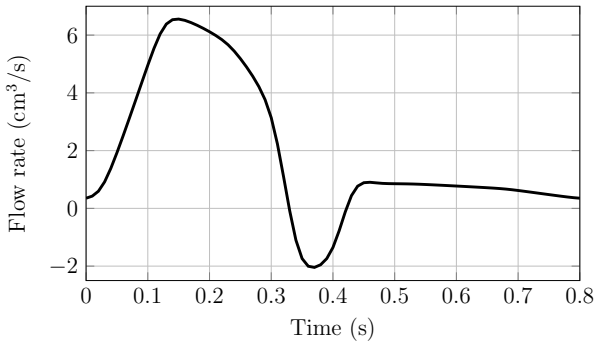


Fig. 4.14 Inlet flow rate $Q(t)$ during the heart cycle.

$\boldsymbol{v}_g$, $\boldsymbol{v}_r$ are specified at the graft and host artery inlets, respectively,

$$\begin{aligned} \boldsymbol{\sigma}(\boldsymbol{v},p)\boldsymbol{n} &= \boldsymbol{0} && \text{on } \Gamma_N \\ \boldsymbol{v} &= \boldsymbol{0} && \text{on } \Gamma_w \\ \boldsymbol{v} &= \mu_1\, \boldsymbol{v}_r\, Q(t) && \text{on } \Gamma_r \\ \boldsymbol{v} &= (1-\mu_1)\, \boldsymbol{v}_g\, Q(t) && \text{on } \Gamma_g. \end{aligned} \tag{4.5}$$

The inlet flow rate profile $Q(t)$ is reported in Fig. 4.14. Following [Col14], we consider as parameter $\mu_1 \in [0, 0.4]$ the percentage of residual flow entering from the host artery.

As regards the high-fidelity discretization, we employ the setup of Sect. 4.1.1 with $T = 1.6$ s (two heartbeats of 0.8 s each) and $\Delta t = 0.005$ s. The dimension of the high-fidelity model is $N_h = 263\,572$. In this case, each time step takes about 45 seconds using 8 CPU cores[2].

[2]Offline computations are performed on a node of the SuperB cluster at EPFL with two Intel Xeon

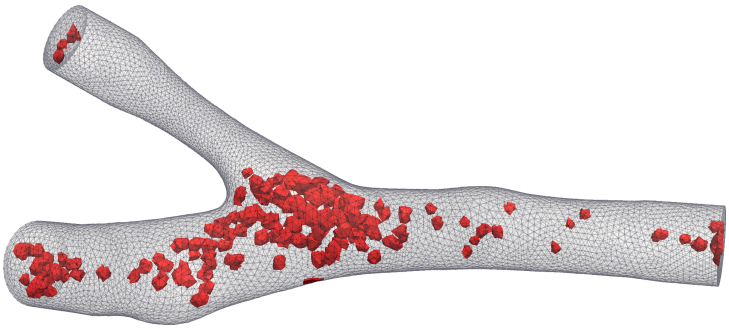


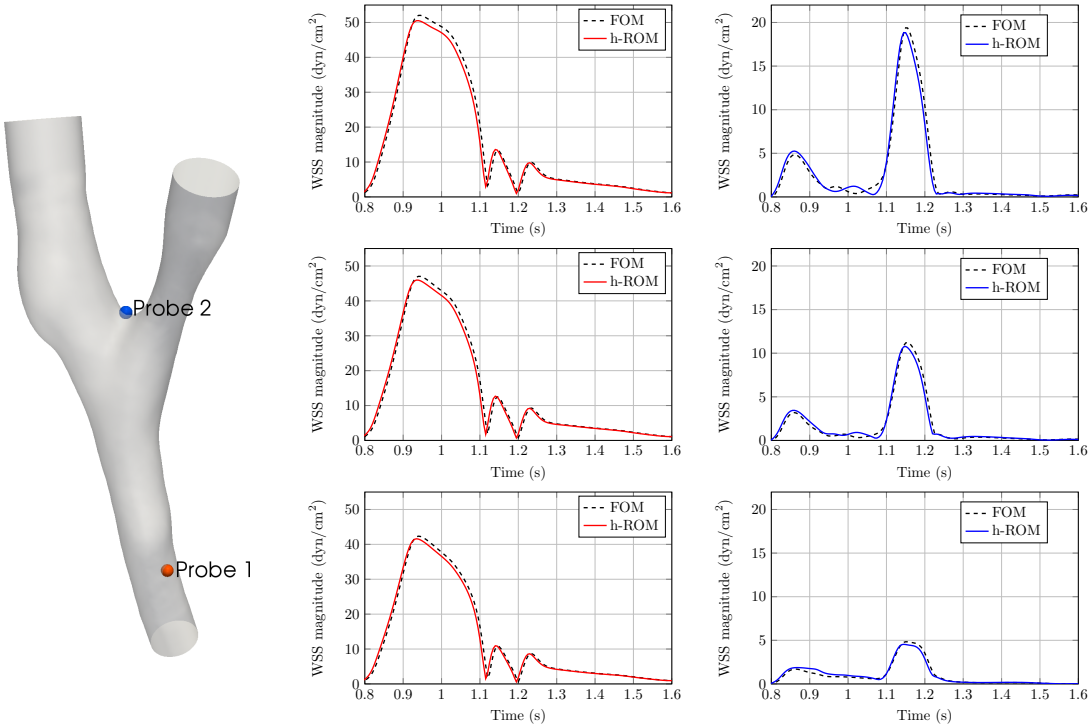
Fig. 4.15 Reduced mesh for the femoropolipteal bypass problem.

4.2.2 Reduction

We generate a reduced-order model for the problem at hand using the procedure detailed in Algorithm 3.7. In particular, we consider $K = 3$ training input parameters corresponding to $\mu_1 = \{0, 0.2, 0.4\}$. As in the case of the cerebral aneurysm, our goal is to build a ROM able to accurately predict the behavior of the flow in the fully developed periodic regime. To this end, in step (1) of Algorithm 3.7 solution snapshots are collected every two time steps in the time interval $[0.8, 1.6]$ s. Using a tolerance $\varepsilon_u = 10^{-3}$, POD retains the first $N_v = 85$ velocity $N_p = 48$ pressure modes. In step (2) of the algorithm we collect system snapshots every four times steps in the time interval $[0.8, 1.6]$ s. Using a tolerance of 10^{-4} for POD, we end up with MDEIM approximations of size $M_s = 68$ and $M_m = 87$. The reduced mesh shown in Fig. 4.15 is made of 5 935 elements, corresponding to 1.55% of the original ones.

We measure the accuracy of the resulting hyper-ROM at the training parameters by comparing the WSS. Specifically, Fig. 4.16 compares the time-history of the WSS magnitude predicted by the high-fidelity and reduced models at two probes. Figures 4.17 and 4.18 show the blood flow velocity and the WSS distribution predicted by the hyper-ROM at different times for $\mu_1 = 0$ and $\mu_1 = 0.4$. Solving the hyper-ROM takes about 0.4 seconds per time step (using one core for the assembly on the reduced mesh), thus delivering a speedup of about 110 with respect to the high-fidelity model.

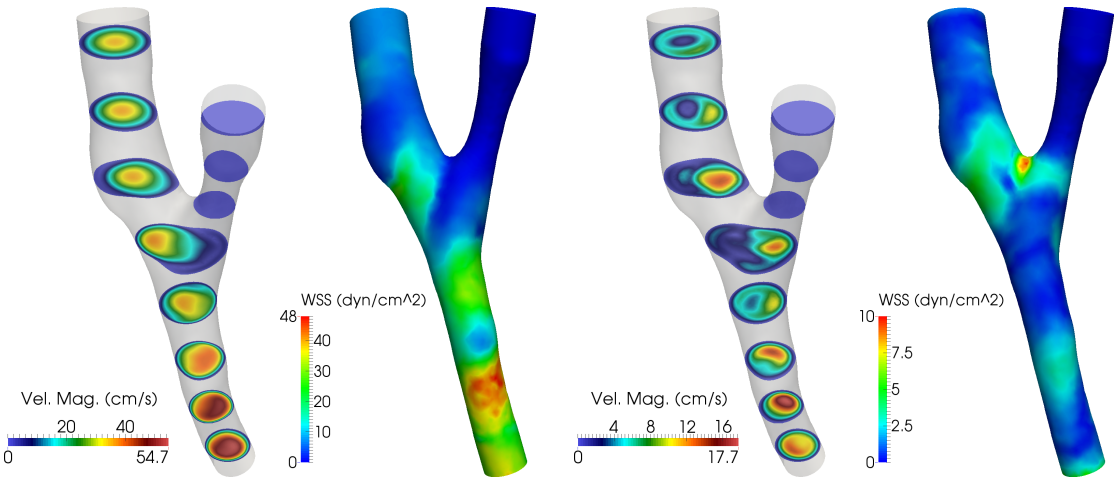
E5-2660 processors and 64 GB of RAM.



(a) Probes position

(b) Probe 1 on the left, probe 2 on the right; $\mu_1 = \{0, 0.2, 0.4\}$ from top to bottom

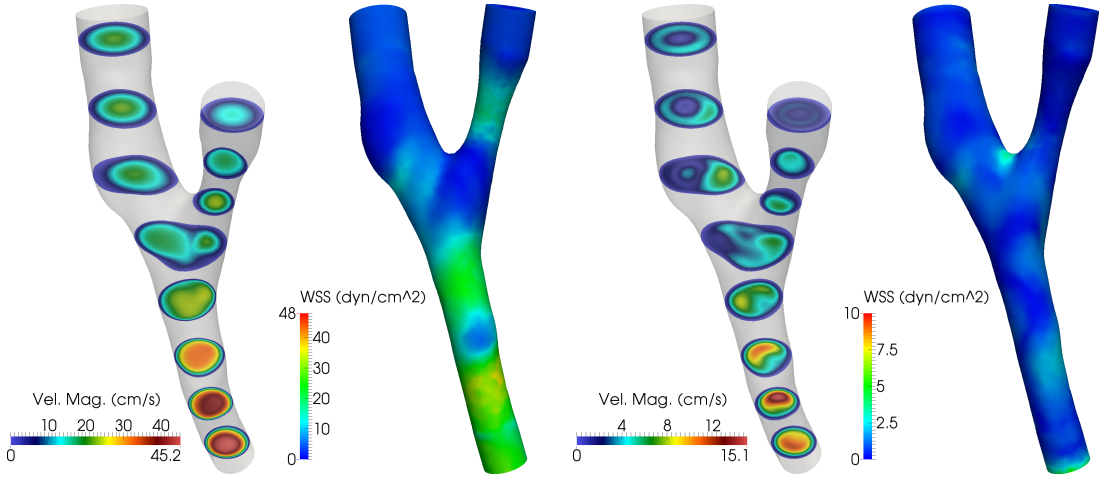
Fig. 4.16 Time-history of the WSS magnitude at two probes for the high-fidelity and reduced-order models.



(a) $t = 0.21$ s

(b) $t = 0.40$ s

Fig. 4.17 Blood flow velocity on interior cuts and WSS magnitude distribution predicted by the hyper-ROM for $\mu_1 = 0$.



(a) $t = 0.21$ s (b) $t = 0.40$ s

Fig. 4.18 Blood flow velocity on interior cuts and WSS magnitude distribution predicted by the hyper-ROM for $\mu_1 = 0.4$.

4.3 Hyper-reduction of a fluid-wall mass transport model

In the cardiovascular context, mass transfer refers to exchange of substances between blood and the arterial wall. Substances of interest include oxygen [ME97, PLP+02, PP03, PZPQ05], low-density lipoprotein (LDL) [SE02, OKP08, Olg12], as well as potential therapeutic agents designed for local delivery by intravascular infusion [CBBH08, HHD14, HHB+12]. In all these cases, the solutes are passive scalars essentially convected by the blood along the vessels, while the absorption processes through the arterial wall are related to the stress induced by the blood on the vascular tissue. The solute distribution and availability inside the vessels and into the vascular walls is thus strongly related to flow dynamics of blood. In particular, it has been observed that irregular flow patterns (such as flow separation, flow recirculation, low and oscillating wall shear stresses) result in disturbed mass distributions. In the case of LDL and oxygen transport, this may eventually lead to the development of atherosclerotic diseases. The numerical simulation of solutes dynamics inside the vessels as well as into the vascular walls could therefore be useful for revealing the relationships between irregular flow patterns, mass transfer, and possible pathogenesis.

Here, we consider the *fluid-wall model* for the dynamics of blood solutes which was introduced in [KP99, QVZ01]; see also [RP96a, PP03, CCC+13] and references therein. This model is based on an advection-diffusion equation describing the solute dynamics in the arterial lumen, the convective field being provided by the blood velocity. This equation is coupled with a pure diffusive one accounting for the solute dynamics inside the arterial wall, where convection is negligible. The coupling conditions matching the two subproblems follows from a suitable modeling of the endothelium, an active membrane which physically separate the two subdomains.

We state the problem in a rather general form which is suitable to model oxygen, macromolecules, as well as drug transport. Its well-posedness and numerical approx-

imation, coupled with the Navier-Stokes equations for the description of the blood velocity and pressure fields, have been analyzed in [QVZ01, QVZ02]. More complex models taking into account also the artery wall compliance have been proposed, e.g., in [CBBH08, YCC$^+$14].

In Sect. 4.4 we shall specify this model to the case of oxygen transport in the femoropopliteal bypass introduced in Sect. 4.2.

4.3.1 Model description

We denote by $\Omega \in \mathbb{R}^d$ $(d = 2, 3)$ a vascular district composed by a fluid subdomain Ω_f and a solid subdomain Ω_w. The former coincides with the artery lumen, while the latter consists of the artery wall; in particular, we assume Ω_w to model the intima and media layers of the arterial wall. We denote by $\Gamma = \overline{\Omega}_f \cap \overline{\Omega}_w$ the interface between the fluid and solid subdomains, i.e. Γ models the endothelial layer.

For $\mathbf{x} \in \Omega_f$ and $t > 0$ we denote by $\boldsymbol{v}(\mathbf{x}, t)$ the velocity of the blood and by $p(\mathbf{x}, t)$ its kinematic pressure. We assume the blood to be an incompressible Newtonian fluid within rigid walls. Thus, the blood motion is described by the incompressible Navier-Stokes equations.

We denote by $C_f(\mathbf{x}, t)$ and $C_w(\mathbf{x}, t)$ the (dimensionless) concentrations of the solute in the lumen Ω_f and in the wall Ω_w, respectively. The dynamics of solutes is described by an advection-diffusion process. In the lumen, the convective field is provided by the blood velocity, while in the wall, because of the very low transmural velocity, we neglect the advection phenomena. This is particularly true when dealing with oxygen transport, since transmural velocity is one to two orders of magnitude smaller than oxygen diffusion velocity [ME97]. (A reaction term could also be considered in the wall domain to take into account solute consumption by cells within the arterial tissue, see, e.g., [CCC$^+$13, SLW$^+$14]). The interface Γ can be regarded as a permeable membrane whose permeability $\zeta = \zeta(\tau_w)$ is a positive function of the wall shear stress τ_w exerted by the blood on the wall [RP96b, RPP97, QVZ01]. In our approach, the solute flux through Γ is proportional to the difference of concentration between lumen and wall.

We end up with the following equations for the solute concentration in the lumen

$$\frac{\partial C_f}{\partial t} + \boldsymbol{v} \cdot \nabla C_f - \alpha_f \Delta C_f = 0 \qquad \text{in } \Omega_f \times (0, T) \tag{4.6}$$

and in the arterial wall

$$\frac{\partial C_w}{\partial t} - \alpha_w \Delta C_w = 0 \qquad \text{in } \Omega_w \times (0, T). \tag{4.7}$$

Here, α_f and α_w denote the solute diffusivity in the fluid and wall domains, respectively. The matching conditions at the interface read

$$\begin{cases} \alpha_f \nabla C_f \cdot \boldsymbol{n}_f + \zeta(\tau_w)(C_f - C_w) = 0 & \text{on } \Gamma \times (0, T) \\ \alpha_w \nabla C_w \cdot \boldsymbol{n}_w + \zeta(\tau_w)(C_w - C_f) = 0 & \text{on } \Gamma \times (0, T). \end{cases} \tag{4.8}$$

System (4.6)-(4.8) is then complemented with suitable Dirichlet and/or Neumann conditions at the inlet, outlet and wall outer boundaries (as well as initial conditions) specific for the problem at hand. In the following, we shall denote by $\Gamma_{f,d}$ and $\Gamma_{w,d}$ the Dirichlet portions of the fluid and solid boundaries, respectively. Moreover, we shall omit the τ_w-dependence of ζ.

4.3.2 Finite element approximation

In order to carry out a numerical discretization of the coupled problem (4.6)-(4.8), we first introduce its weak formulation. Upon defining the spaces $V_f = \{v \in H^1(\Omega_f) : v|_{\Gamma_{f,d}} = 0\}$ and $V_w = \{v \in H^1(\Omega_w) : v|_{\Gamma_{w,d}} = 0\}$, the weak formulation reads: for all $t \in (0, T)$, find $C_f(t) \in V_f$ and $C_w(t) \in V_w$ such that for all $\phi_f \in V_f$ and $\phi_w \in V_w$

$$\begin{cases} \left(\dfrac{\partial C_f}{\partial t} + \boldsymbol{v} \cdot \nabla C_f, \phi_f\right)_\Omega + (\alpha_f \nabla C_f, \nabla \phi_f)_\Omega + (\zeta(C_f - C_w), \phi_f)_\Gamma = g_f(\phi_f), \\ \left(\dfrac{\partial C_w}{\partial t}, \phi_w\right)_\Omega + (\alpha_w \nabla C_w, \nabla \phi_w)_\Omega + (\zeta(C_w - C_f), \phi_w)_\Gamma = g_w(\phi_w), \end{cases} \tag{4.9}$$

with $C_f(0) = C_f^0$ and $C_w(0) = C_w^0$. The functionals $g_f : V_f \to \mathbb{R}$ and $g_w : V_f \to \mathbb{R}$ encode the action of the nonhomogeneous Dirichlet conditions on $\Gamma_{f,d}$ and $\Gamma_{w,d}$, respectively. A well-posedness analysis of problem (4.9) has been carried out in [QVZ01].

For the space discretization of the equations at hand, we use the finite element method. In particular, for what concerns the Navier-Stokes equations, we consider the approximation method described in Sect. 3.7.2. Referring to the mass transport equations (4.9), we note that the solute dynamics in the fluid domain is dominated by advection effects. For instance, in the case of oxygen transport in the femoropopliteal bypass of Sect. 4.2, characteristic blood velocity is $V = 10$ cm s^{-1}, oxygen blood diffusivity is $\alpha_f \approx 10^{-5}$ cm^2 s^{-1} and the vessel radius is $r \approx 0.3$ cm, yielding a Péclet number

$$\mathrm{Pe} = \frac{2Vr}{\alpha_f} \approx 6 \cdot 10^5.$$

As it is well known, the Galerkin finite element method could be inaccurate when facing convection dominated problems and resorting to a stabilization technique becomes mandatory. Following [QVZ01, PZPQ05, CBBH08], we consider the streamline-upwind/Petrov–Galerkin (SUPG) stabilization already introduced in Sect. 3.2.2. We also remark that an SUPG stabilization augmented with a discontinuity-capturing operator was proposed in [BCTH07] to better resolve sharp interior and boundary layers.

We first introduce (conforming) finite element partitions $\mathcal{T}_{h,f}$ and $\mathcal{T}_{h,w}$ of the fluid and solid domain, respectively, and the finite element spaces

$$X_{h,f}^r = \{w_h \in C^0(\overline{\Omega_f}) : w_h|_K \in \mathbb{P}_r \ \forall K \in \mathcal{T}_{h,f}\},$$
$$X_{h,w}^r = \{w_h \in C^0(\overline{\Omega_w}) : w_h|_K \in \mathbb{P}_r \ \forall K \in \mathcal{T}_{h,w}\}.$$

Upon defining $V_{h,f} = V_f \cap X_{h,f}^r$ and $V_{h,w} = V_w \cap X_{h,w}^r$, we end up with the following semi-discrete weak formulation: for all $t \in (0, T)$, find $C_f(t) \in V_{h,f}$ and $C_w(t) \in V_{h,w}$ such that for all $\phi_f \in V_{h,f}$ and $\phi_w \in V_{h,w}$

$$\begin{cases} \left(\dfrac{\partial C_{h,f}}{\partial t} + \boldsymbol{v}_* \cdot \nabla C_{h,f}, \phi_f\right)_{\Omega_f} + (\alpha_f \nabla C_{h,f}, \nabla \phi_f)_{\Omega_f} + (\zeta(C_{h,f} - C_{h,w}), \phi_f)_\Gamma \\ \quad + \displaystyle\sum_{K \in \mathcal{K}_{h,f}} \left(\frac{\partial C_{h,f}}{\partial t} + \boldsymbol{v}_* \cdot \nabla C_{h,f} - \nabla \cdot (\alpha_f \nabla C_{h,f}), \tau_K \boldsymbol{v}_* \cdot \nabla \phi_f\right)_K = g_f(\phi_f), \\ \left(\dfrac{\partial C_{h,w}}{\partial t}, \phi_w\right)_{\Omega_w} + (\alpha_w \nabla C_{h,w}, \nabla \phi_f)_{\Omega_w} + (\zeta(C_{h,w} - C_{h,f}), \phi_w)_\Gamma = g_w(\phi_w), \end{cases} \tag{4.10}$$

where the stabilization parameter τ_K is defined as in Sect. 3.2.2 and $\boldsymbol{v}_* = \boldsymbol{v}_*(t)$ denotes a suitable (high-fidelity or reduced) approximation of the blood velocity field.

We denote by $\{\varphi_{f,i}\}_{i=1}^{N_{h,f}}$ and $\{\varphi_{w,i}\}_{i=1}^{N_{h,w}}$ Lagrangian finite element bases for $V_{h,f}$ and $V_{h,w}$, respectively. We also denote by $\mathbf{C}_{h,f} \in \mathbb{R}^{N_{h,f}}$ and $\mathbf{C}_{h,w} \in \mathbb{R}^{N_{h,w}}$ the vectors of coefficients in the expansions of $C_{h,f}$ and $C_{h,w}$ with respect to these bases. The algebraic formulation of (4.10) reads: for all $t \in (0,T)$, find $\mathbf{C}_{h,f}(t) \in \mathbb{R}^{N_{h,f}}$ and $\mathbf{C}_{h,w}(t) \in \mathbb{R}^{N_{h,w}}$ such that

$$\begin{cases} \mathbf{M}_f \dfrac{d\mathbf{C}_{h,f}}{dt} + \mathbf{A}_f \mathbf{C}_{h,f} + \mathbf{M}_{ff}\mathbf{C}_{h,f} - \mathbf{M}_{fw}\mathbf{C}_{h,w} = \mathbf{g}_f \\ \mathbf{M}_w \dfrac{d\mathbf{C}_{h,w}}{dt} + \mathbf{A}_w \mathbf{C}_{h,w} + \mathbf{M}_{ww}\mathbf{C}_{h,w} - \mathbf{M}_{wf}\mathbf{C}_{h,f} = \mathbf{g}_w, \end{cases} \tag{4.11}$$

with $\mathbf{C}_{h,f}(0) = \mathbf{C}^0_{h,f}$ and $\mathbf{C}_{h,w} = \mathbf{C}^0_{h,w}$. $\mathbf{M}_f$ and $\mathbf{M}_w$ are the mass matrices in the fluid and solid domains, respectively, while $\mathbf{A}_f$ and $\mathbf{A}_w$ are the matrices encoding the action of the differential operators. The weighted mass matrices $\mathbf{M}_{ff}, \mathbf{M}_{ss}, \mathbf{M}_{fs}$ and $\mathbf{M}_{sf}$ are defined as

$$(\mathbf{M}_{ff})_{i,j} = (\zeta\varphi_{f,j}, \varphi_{f,i})_\Gamma, \qquad (\mathbf{M}_{ww})_{k,l} = (\zeta\varphi_{w,l}, \varphi_{w,k})_\Gamma,$$
$$(\mathbf{M}_{fw})_{i,l} = (\zeta\varphi_{w,l}, \varphi_{f,i})_\Gamma, \qquad (\mathbf{M}_{wf})_{k,j} = (\zeta\varphi_{f,j}, \varphi_{w,k})_\Gamma.$$

for $i,j = 1,\dots,N_{h,f}$ and $k,l = 1,\dots,N_{h,w}$.

In a more compact form, (4.11) reads: for all $t \in (0,T)$, find $\mathbf{C}_h(t) \in \mathbb{R}^{N_h}$ such that

$$\mathbf{M}(t)\frac{d\mathbf{C}_h}{dt} + \mathbf{A}(t)\mathbf{C}_h = \mathbf{g}(t), \tag{4.12}$$

where $\mathbf{C}_h = (\mathbf{C}_{f,h}, \mathbf{C}_{h,w})^T$ and $N_h = N_{h,f} + N_{h,w}$. The block-partitioned matrices $\mathbf{M}(t)$ and $\mathbf{A}(t)$ are given by

$$\mathbf{M}(t) = \begin{pmatrix} \mathbf{M}_f(t) & 0 \\ 0 & \mathbf{M}_w \end{pmatrix}, \qquad \mathbf{A}(t) = \begin{pmatrix} \mathbf{A}_f(t) + \mathbf{M}_{ff}(t) & -\mathbf{M}_{fw}(t) \\ -\mathbf{M}_{wf}(t) & \mathbf{A}_w + \mathbf{M}_{ww}(t) \end{pmatrix}, \tag{4.13}$$

where the time-dependency is implicit through the blood velocity $\mathbf{v}_*(t)$ and the permeability function that depends on $\tau_w = \tau_w(\mathbf{v}_*(t))$.

If some of the problem coefficients – such as the Reynolds number in the fluid equations or the diffusivity constants α_f and α_w – are treated as parameters, we obtain the following parametrized formulation of (4.13): given $\boldsymbol{\mu} \in \mathcal{D}$, for all $t \in (0,T)$ find $\mathbf{C}_h(t) \in \mathbb{R}^{N_h}$ such that

$$\mathbf{M}(t;\boldsymbol{\mu})\frac{d\mathbf{C}_h}{dt} + \mathbf{A}(t;\boldsymbol{\mu})\mathbf{C}_h = \mathbf{g}(t;\boldsymbol{\mu}). \tag{4.14}$$

Since (4.14) is precisely of the form (3.46), a reduced-order model for the fluid-wall transport problem can be generated by applying the strategy developed in Sect. 3.6.

4.4 Oxygen transfer in a femoropopliteal bypass

In this section, we apply the fluid-wall model to simulate oxygen transfer in the femoropopliteal bypass already considered in Sect. 4.2.

4.4.1 Description of the domain and of the data

The geometry of the fluid domain Ω_f is the one already considered in Sect. 4.2.1, while the solid domain Ω_w is generated by extruding the arterial wall in the normal direction as shown in Fig. 4.19; see [MCG+12] for further details. The resulting wall thickness is equal to 10% of the vessel diameter. The fluid mesh is made of 65 893 vertices and 382 301 tetrahedral elements, while the solid one is made of 35 395 vertices and 168 684 tetrahedral elements; see Fig. 4.20.

We denote by $\Gamma_{f,g}$ the graft lumen inlet and by $\Gamma_{f,r}$ the occluded artery lumen inlet; $\Gamma_{f,n}$ is the outlet boundary of the fluid domain. Similarly, $\Gamma_{w,g}$ is the graft wall inlet, $\Gamma_{w,r}$ the occluded artery wall inlet, $\Gamma_{w,n}$ the outlet boundary of the solid domain, while $\Gamma_{w,o}$ denotes the outer surface of the wall (i.e. the media-adventitia interface).

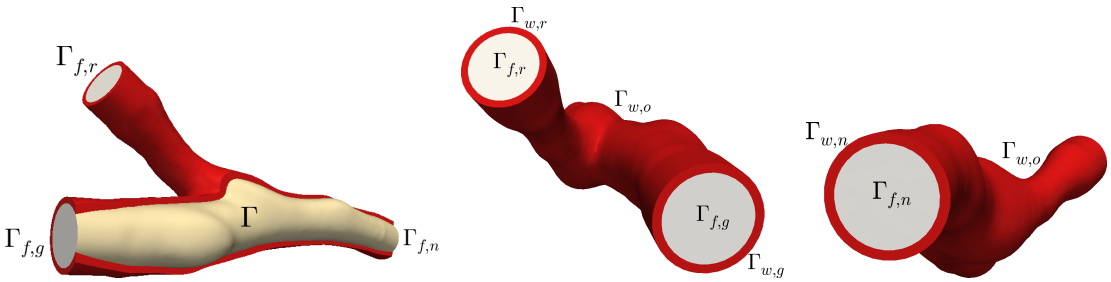
The setup of the fluid problem is the same as in Sect. 4.2.1. In particular, we consider as parameter μ_1 the percentage of residual flow in the occluded artery. For the fluid-wall model, we define C_f and C_w as the dimensionless oxygen concentration with respect to the physiological reference concentration $C_0 = 2.58 \cdot 10^{-3}$ ml cm^{-3}. The oxygen diffusivity in blood and arterial wall is given by $\alpha_f = 1.2 \cdot 10^{-5}$ cm^{2}s^{-1} and $\alpha_w = 0.9 \cdot 10^{-5}$ cm^{2}s^{-1} [ME97], respectively. However, as these values yield a very high local Péclet number for the fluid mesh at hand, we rather fix $\alpha_f = 1.2 \cdot 10^{-4}$ cm^{2}s^{-1} and $\alpha_w = 0.9 \cdot 10^{-4}$ cm^{2}s^{-1}. At the graft lumen we impose a uniform concentration value of oxygen $C_f = 1$, while at the outlet $\Gamma_{f,n}$ we impose a homogeneous Neumann condition $\alpha_f \nabla C_f \cdot \boldsymbol{n}_f = 0$. At the occluded artery inlet $\Gamma_{f,r}$, we impose $C_f = 1$ if $\mu_1 > 0$, $C_f = 0$ otherwise. At the inlet and outlet boundaries of the solid domain we impose a homogeneous Neumann condition $\alpha_w \nabla C_w \cdot \boldsymbol{n}_w = 0$, while a uniform concentration value of oxygen $C_w = 0.5$ is prescribed on the outer surface of the wall $\Gamma_{w,0}$ [BG82, QVZ01, CCC+13]. Following [RPP97, QVZ01], the wall permeability $\zeta(\tau_w)$ is assumed to vary linearly with respect to τ_w,

$$\zeta = \beta(1 + \tau_w). \tag{4.15}$$

The constant β is usually chosen such that the diffusive flux at the wall matches a physiological arterial flux. In a first test case we consider the reference value $\beta = 2.5 \cdot 10^{-4}$ cm^3 dyn^{-1} s^{-1} reported in [RPP97]. However, since there is only a limited amount of experimental data about how endothelial permeability depends on hemodynamics quantities, we then treat β as a parameter $\mu_2 \in [10^{-4}, 8 \cdot 10^{-4}]$ cm^3 dyn^{-1} s^{-1}.

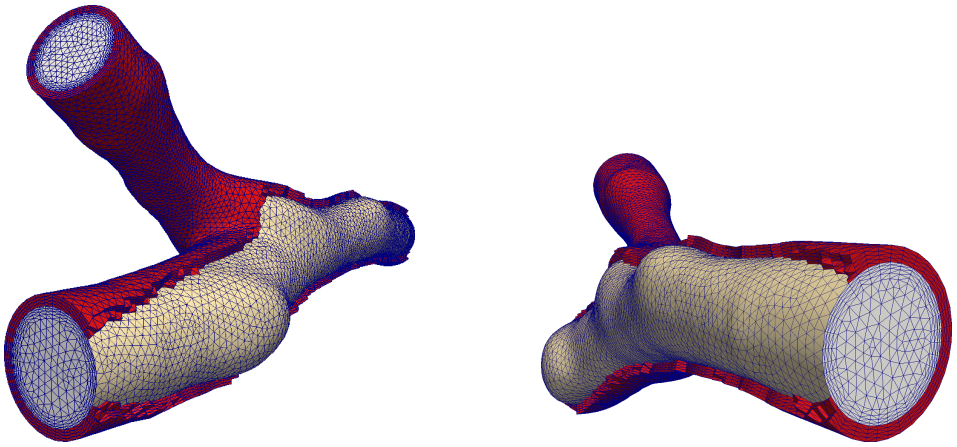
We consider as high-fidelity model the one resulting from the discretization (4.10) with $\boldsymbol{v}_* = \boldsymbol{v}_{N,m}(t, \boldsymbol{\mu})$. The latter denotes the velocity field solution of the hyper-ROM for the Navier-Stokes equations discussed in Sect. 4.2.2. For the space discretization we use linear finite elements, while a BDF2 scheme with a fixed time step $\Delta t = 0.01$ s is used to advance in time (two times bigger than the one used for blood flow computations). The dimension of the high-fidelity model is $N_h = 101\,288$. At each time step, the coupled linear system is solved monolithically by means of the GMRES with a two-level additive Schwarz method that preconditions the entire system together. In particular, we mimic the approach proposed in [BC10, WC14] dealing with fluid-structure interaction problems. To this end, we generate a decomposition of the mesh which is completely independent of the physical variables defined at a given mesh point. As a result, a subdomain may contain both fluid and solid elements.

Due to the periodicity of the underlying transport field, oxygen concentration has to



(a) lateral view (b) inlet (c) outlet

Fig. 4.19 Geometry and boundary condition for the oxygen transport problem in the femoropolipteal bypass: fluid domain in gray, solid domain in red.



(a) inlet (b) outlet

Fig. 4.20 Computational mesh for the oxygen transport problem in the femoropolipteal bypass.

reach a periodic regime as well. To shorten the initial transition phase, we use as initial condition the solution of the corresponding steady problem at $t = t_0$. Therefore, $\mathbf{C}_h^0(\boldsymbol{\mu})$ is the solution of the following problem,

$$\mathbf{A}(t_0;\boldsymbol{\mu})\mathbf{C}_h^0(\boldsymbol{\mu}) = \mathbf{g}(t_0;\boldsymbol{\mu}). \tag{4.16}$$

Then, six heartbeats are simulated, corresponding to a final time $T = 4.8$ s. Note that the underlying transport field is shifted forward in time of one heartbeat to exclude the initial transition phase of the fluid problem.

4.4.2 A reduced-order model for the initial condition

We construct a hyper-ROM for (4.14) by means of the procedure detailed in Algorithm 3.5 (see also Fig. 3.32). In particular, a block-partitioned POD basis for the fluid and wall concentration is constructed, i.e.

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_f & 0 \\ 0 & \mathbf{V}_w \end{pmatrix} \in \mathbb{R}^{(N_{h,f}+N_{h,w})\times(N_f+N_w)}$$

so that the reduced model preserves the structure of the high-fidelity one. We recall from Sect. 3.6.1 that the resulting fully discrete hyper-ROM reads: given $\mathbf{C}_{N,m}^{n}(\boldsymbol{\mu}), \dots,$ $\mathbf{C}_{N,m}^{n+1-\sigma}(\boldsymbol{\mu})$, for $n \geq \sigma - 1$, find $\mathbf{C}_{N,m}^{n+1}(\boldsymbol{\mu}) \in \mathbb{R}^N$ such that

$$\mathbf{M}_N^m(t_{n+1};\boldsymbol{\mu})\frac{\alpha\mathbf{C}_{N,m}^{n+1} - \mathbf{C}_{N,m}^{n,\sigma}}{\Delta t} + \mathbf{A}_N^m(t_{n+1};\boldsymbol{\mu})\mathbf{C}_{N,m}^{n+1} = \mathbf{g}_N^m(t_{n+1};\boldsymbol{\mu}) - \mathbf{V}^T\mathbf{A}_m(t_{n+1};\boldsymbol{\mu})\mathbf{C}_h^0(\boldsymbol{\mu}), \quad (4.17)$$

with $\mathbf{C}_{N,m}^0(\boldsymbol{\mu}) = \mathbf{0}$.

Because of the definition of $\mathbf{C}_h^0(\boldsymbol{\mu})$, evaluating (4.17) first requires a preprocessing phase which involves the solution of the high-fidelity steady problem (4.16). To avoid this computational bottleneck, we propose to build a reduced-order model also for system (4.16). Specifically, we seek an approximate initial condition

$$\mathbf{C}_h^0(\boldsymbol{\mu}) \approx \mathbf{V}_0\widetilde{\mathbf{C}}_{N,m}^0(\boldsymbol{\mu}), \quad (4.18)$$

where $\widetilde{\mathbf{C}}_{N,m}^0 \in \mathbb{R}^{N_0}$ satisfies

$$\mathbf{A}_N^m(t_0;\boldsymbol{\mu})\widetilde{\mathbf{C}}_{N,m}^0 = \mathbf{g}_N^m(t_0;\boldsymbol{\mu}). \quad (4.19)$$

The steady hyper-ROM (4.19) is generated following the procedure detailed in Algorithm 3.4. As a result, approximation (4.18) yields the following hyper-ROM for (4.14): given $\mathbf{C}_{N,m}^{n}(\boldsymbol{\mu}), \dots, \mathbf{C}_{N,m}^{n+1-\sigma}(\boldsymbol{\mu})$, for $n \geq \sigma - 1$, find $\mathbf{C}_{N,m}^{n+1}(\boldsymbol{\mu}) \in \mathbb{R}^N$ such that

$$\mathbf{M}_N^m(t_{n+1};\boldsymbol{\mu})\frac{\alpha\mathbf{C}_{N,m}^{n+1} - \mathbf{C}_{N,m}^{n,\sigma}}{\Delta t} + \mathbf{A}_N^m(t_{n+1};\boldsymbol{\mu})\mathbf{C}_{N,m}^{n+1} = \mathbf{g}_N^m(t_{n+1};\boldsymbol{\mu}) - \mathbf{A}_0^m(t_{n+1};\boldsymbol{\mu})\widetilde{\mathbf{C}}_{N,m}^0(\boldsymbol{\mu}), \quad (4.20)$$

with $\mathbf{C}_{N,m}^0(\boldsymbol{\mu}) = \mathbf{0}$ and

$$\mathbf{A}_0^m(t_{n+1};\boldsymbol{\mu}) = \mathbf{V}^T\mathbf{A}_m(t_{n+1};\boldsymbol{\mu})\mathbf{V}_0 \in \mathbb{R}^{N\times N_0}.$$

Once the reduced solution has been computed, its full-order representation at time t_n is given by $\mathbf{V}_0\widetilde{\mathbf{C}}_{N,m}^0(\boldsymbol{\mu}) + \mathbf{V}\mathbf{C}_{N,m}^n(\boldsymbol{\mu})$. The latter is used, for instance, to visualize the reduced solution over the underlying finite element mesh.

4.4.3 Numerical results: one parameter case

As a first test case we consider only one parameter, namely the percentage of residual flow in the occluded artery. We run Algorithm 3.5 to build the ROM using the same training set of $K = 3$ parameters of Sect. 4.2.2. At each step, we first solve the hyper-ROM for the Navier-Stokes equations and the high-fidelity model for the oxygen transfer. Each high-fidelity time step takes about 10 seconds using 4 cores[3]: parallelism is exploited for both the matrix assembly and the application of the preconditioner. Both solution and

[3]Offline computations are performed on a node (with two Intel Xeon E5-2660 processors and 64 GB of RAM) of the SuperB cluster at EPFL. Online computations are performed on a workstation with a Intel Core i5-2400S processor and 16 GB of RAM

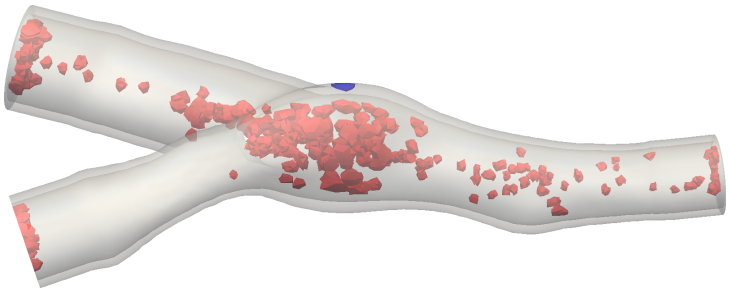


Fig. 4.21 Reduced mesh for the oxygen transport in the femoropolipteal bypass; red elements belong to the fluid domain, while blue elements belong to the solid one.

system snapshots are collected (each time step) only during the last heartbeat; then, we compress the local snapshots matrices by POD with $\varepsilon_u^{\text{loc}} = 10^{-3}$, $\varepsilon_a^{\text{loc}} = \varepsilon_m^{\text{loc}} = \varepsilon_g^{\text{loc}} = 10^{-4}$. After the snapshots collection we perform step (2) of Algorithm 3.5 obtaining a reduced model with $N_f = 112$, $N_w = 34$, $M_a = 126$, $M_m = 129$ and $M_g = 47$. Globally, the reduced mesh contains about 1.72% of the original elements, see Fig. 4.21; most of them ($\approx 99.7\%$) belong to the fluid subdomain.

We measure the accuracy of the resulting hyper-ROM at the training inputs by computing the time-history of the relative $H^1(\Omega)$ error between the full and reduced solutions. The results reported in Fig. 4.22 show that the relative error is always below 1%. Solving the hyper-ROM takes about 0.2 seconds per time step (using one core for the assembly on the reduced mesh), thus delivering a speedup of about 50 with respect to the high-fidelity model.

We present and compare the results of the simulations for different parameter values in terms of the Sherwood number. The latter represents the non-dimensional mass flux through the vessel wall and is computed as [ME97, CC08]

$$\text{Sh} = -\frac{2r\,(\nabla C_f \cdot \boldsymbol{n})}{C_{f,in} - C_{w,o}},$$

where $r = 0.3$ cm is a reference vessel diameter, $C_{f,in} = 1$ is the inlet oxygen concentration, and $C_{w,o} = 0.5$ is the oxygen concentration at the outer wall. A nodal reconstruction of $\nabla C_f \cdot \boldsymbol{n}$ is computed using the patch averaging technique described in Sect. 4.1.1. Figures 4.23 and 4.24 show the Sherwood number distribution for $\mu_1 = 0$ and $\mu_1 = 0.4$ at different time steps. As expected, we observe a significant correlation with the WSS distribution reported in Figs. 4.17 and 4.18. Figure 4.25 compares the wall concentration C_f on some interior cuts for $\mu_1 = 0$ and $\mu_1 = 0.4$ at $t = 0.21$ s. Moreover, Fig. 4.26 shows the wall concentration for $\mu_1 = 0$ at different times. While the Sherwood number distribution varies significantly over time, the oxygen distribution in the wall is far less affected by blood pulsatility.

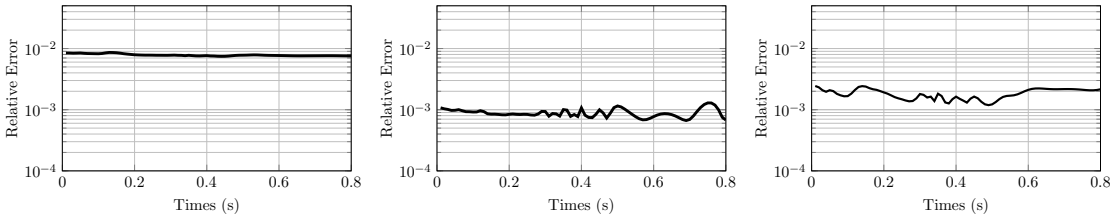
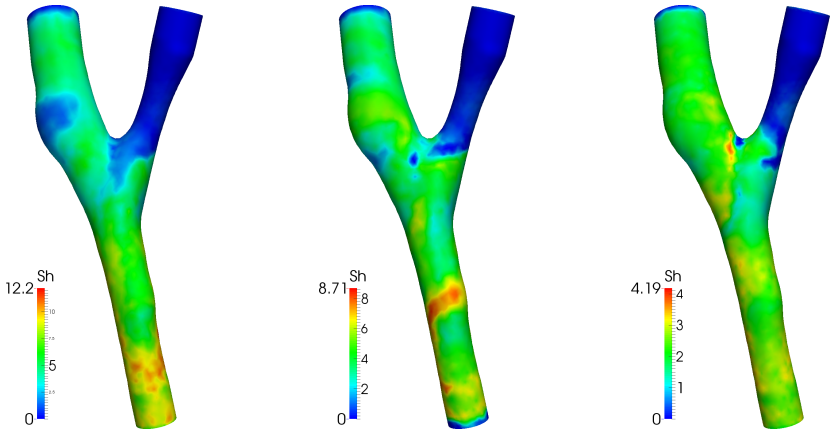
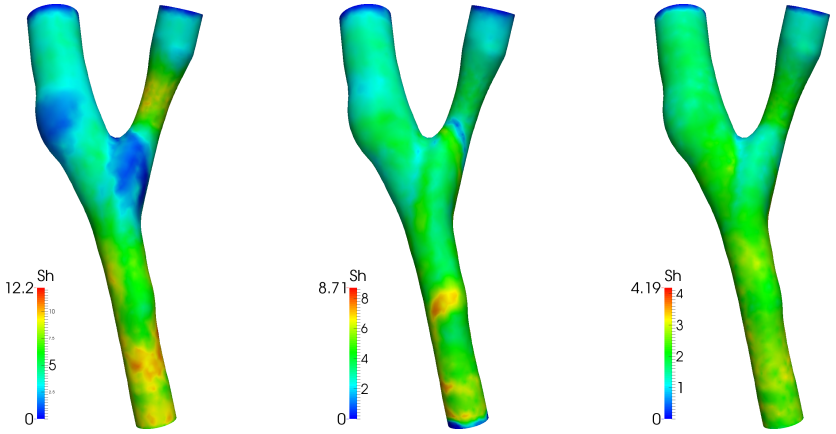


Fig. 4.22 Time-history of the relative $H^1(\Omega)$ error on the oxygen concentration at the the training parameters; the error is measured over the last heartbeat.



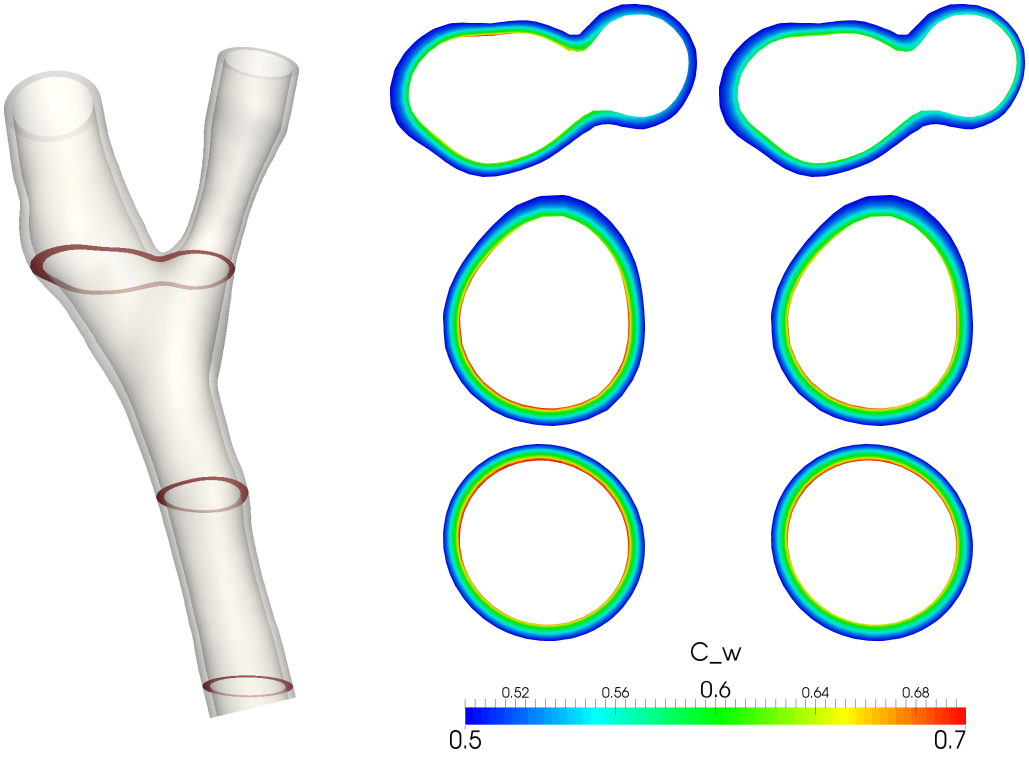
(a) $t = 0.21$ s (b) $t = 0.40$ s (c) $t = 0.65$ s

Fig. 4.23 Sherwood number distribution predicted by the hyper-ROM for $\mu_1 = 0$ at different time steps.



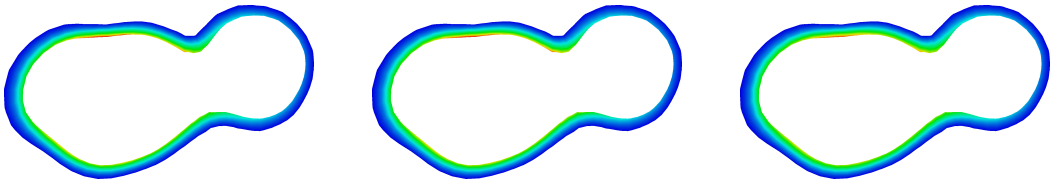
(a) $t = 0.21$ s (b) $t = 0.40$ s (c) $t = 0.65$ s

Fig. 4.24 Sherwood number distribution predicted by the hyper-ROM for $\mu_1 = 0.4$ at different time steps.



(a) interior cuts position (b) wall concentration for $\mu_1 = 0$ (left) and $\mu_1 = 0.4$ (right)

Fig. 4.25 Wall concentration at $t = 0.21$ s on interior cuts.

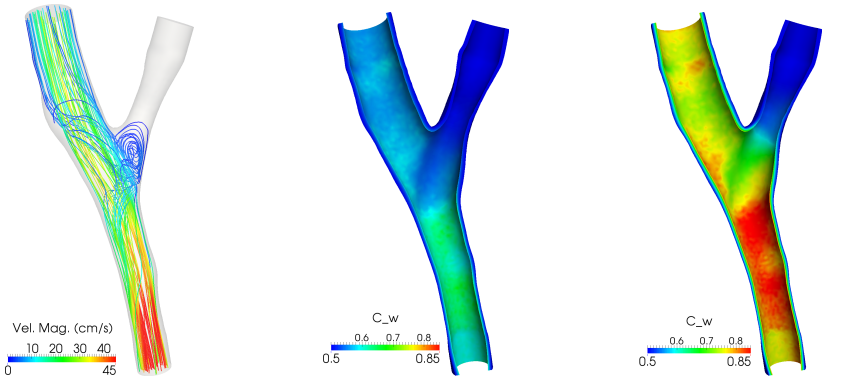


(a) $t = 0.21$ s (b) $t = 0.40$ s (c) $t = 0.65$ s

Fig. 4.26 Wall concentration at $t = 0.21, 0.40, 0,65$ s for $\mu_1 = 0$ (colormap given in Fig. 4.25).

4.4.4 Numerical results: two parameters case

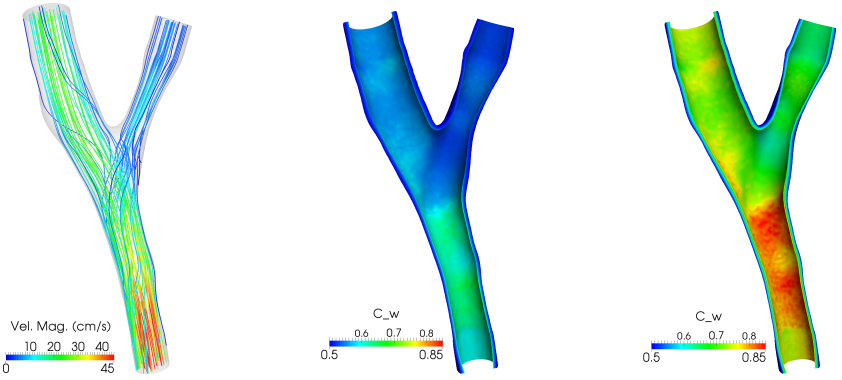
We now consider as parameter $\mu_2 \in [10^{-4}, 8{\cdot}10^{-4}]$ the constant β entering the expression of the permeability function (4.15). We generate the hyper-ROM by means of Algorithm 3.5 with $K = 8$ training inputs and the same setup of Sect. 4.4.3. We end up with a reduced model of dimension $N_f = 263$, $N_w = 83$, featuring $M_a = 180$, $M_m = 180$ and $M_g = 114$ affine terms; the corresponding reduced mesh contains about 3% of the original elements.



(a) velocity streamlines (b) C_w for $\mu_2 = 10^{-4}$ (c) C_w for $\mu_2 = 8 \cdot 10^{-4}$

Fig. 4.27 Oxygen concentration in the wall at $t = 0.21$ s for $\mu_1 = 0$.

Figures 4.27 and 4.28 compare the oxygen concentration predicted by the hyper-ROM at $t = 0.21$ s for different flow conditions and values of the permeability constant β. In particular, as the permeability increases the concentration flux at the wall significantly increases, resulting in higher oxygen concentration inside the vessel wall. Solving the hyper-ROM takes on average 0.35 seconds per time step (using one core for the assembly on the reduced mesh), thus delivering a speedup of about 30 with respect to the high-fidelity model. As a result, running a complete simulation of the blood flow and oxygen transfer using the hyper-ROMs takes about five minutes, when instead the high-fidelity solver needs more than five hours. As it enables a fast parameter exploration, the reduced model could thus be used, e.g., to calibrate (a posteriori) the permeability constant to match experimental or reference values of some more directly measurable quantity.



(a) velocity streamlines (b) C_w for $\mu_2 = 10^{-4}$ (c) C_w for $\mu_2 = 8 \cdot 10^{-4}$

Fig. 4.28 Oxygen concentration in the wall at $t = 0.21$ s for $\mu_1 = 0.2$.

5 A model order reduction framework for parametrized PDE-constrained optimization

In this chapter, we propose a model order reduction framework for parametrized quadratic optimization problems constrained by linear and nonlinear stationary PDEs. By characterizing the solutions of the optimization problem as the solutions of the corresponding optimality system, we build a ROM following a suitable all-at-once optimize-then-reduce paradigm. Low-dimensional spaces for the state, control and adjoint variables are simultaneously constructed by means of either the greedy algorithm or proper orthogonal decomposition. By this approach, we can easily estimate the error between the high-fidelity and reduced solutions; further, a bound for the error on the cost functional is obtained. For the sake of computational efficiency, we also integrate into this framework the ROMES method presented in [DC15] to generate tighter error indicators. Finally, we apply this general framework to the case of optimization problems constrained by linear elliptic, Stokes and Navier-Stokes equations.

5.1 Problem statement

The aim of this chapter is to propose a general approach for the construction of reduced-order models (ROMs) for parameterized PDE-constrained optimization problems of the following form: given $\boldsymbol{\mu} \in \mathcal{D}$,

$$\min_{(y,u)\in Y\times\mathcal{U}} \mathcal{J}(y,u;\boldsymbol{\mu}) \quad \text{subject to} \quad \mathcal{E}(y,u;\boldsymbol{\mu}) = 0 \text{ in } Q'. \tag{5.1}$$

Here Y, $\mathcal{U}$, Q are Hilbert spaces along with their duals Y', $\mathcal{U}'$, Q', $y \in Y$ denotes the state variable, while $u \in \mathcal{U}$ is the control (or design) variable. The equality constraint $\mathcal{E}(\cdot;\cdot;\boldsymbol{\mu}) : Y \times \mathcal{U} \to Q'$ represents a (or a system of) stationary nonlinear PDE, while the cost functional $\mathcal{J}(\cdot,\cdot;\boldsymbol{\mu}) : Y \times \mathcal{U} \to \mathbb{R}$ is assumed to be quadratic. Both $\mathcal{J}$ and $\mathcal{E}$ may depend on a vector $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^P$ of $P \geq 1$ input parameters representing either physical or geometrical features. Problem (5.1) represents the continuous counterpart of the discrete optimization problem (1.15) discussed in Sect. 1.3.

After introducing a suitable full-order (or high-fidelity) discretization of the associated optimality system, solving the optimization problem (5.1) requires to solve a large scale

system of nonlinear equations. To alleviate the computational burden, we aim at developing a reduction strategy based on a *all-at-once optimize-then-reduce* paradigm.

To this end, it is convenient to introduce the optimization variable $x = (y, u) \in X = Y \times \mathcal{U}$ and to express problem (5.1) in the so-called *full-space* formulation [HPUU09, ABG+06]

$$\min_{x \in X} \mathcal{J}(x; \boldsymbol{\mu}) \quad \text{subject to} \quad \mathcal{E}(x; \boldsymbol{\mu}) = 0 \text{ in } Q'. \tag{5.2}$$

We thus treat the state and control variables as independent variables, linked by the constraint equation.

5.1.1 Lagrange multipliers and first order optimality conditions

Let us first define the Lagrangian functional

$$\mathcal{L}(x, p; \boldsymbol{\mu}) = \mathcal{J}(x; \boldsymbol{\mu}) + \langle \mathcal{E}(x; \boldsymbol{\mu}), p \rangle, \tag{5.3}$$

being $p \in Q$ a Lagrange multiplier associated to the constraint. For any fixed $\boldsymbol{\mu} \in \mathcal{D}$, we make the following assumptions [HPUU09, IK08]:

(H1) problem (5.2) has at least a local optimal solution $\bar{x} \in X$;

(H2) the mappings $\mathcal{J}(\cdot; \boldsymbol{\mu}) \colon X \to \mathbb{R}$ and $\mathcal{E}(\cdot; \boldsymbol{\mu}) \colon X \to Q'$ are continuously Fréchet differentiable with Lipschitz continuous first derivatives $\mathcal{E}'(\cdot; \boldsymbol{\mu}) : X \to \mathcal{L}(X, Q')$ and $\mathcal{J}'(\cdot; \boldsymbol{\mu}) : X \to X'$, respectively;

(H3) the Fréchet derivative $\mathcal{E}'(\bar{x}; \boldsymbol{\mu})$ of the state operator is surjective, i.e. there exists a constant $\bar{\lambda} > 0$ such that

$$\lambda(\boldsymbol{\mu}) = \inf_{\delta p \in Q} \sup_{\delta x \in X} \frac{\langle \mathcal{E}'(\bar{x}; \boldsymbol{\mu}) \delta x, \delta p \rangle}{\|\delta p\|_Q \|\delta x\|_X} > \bar{\lambda}. \tag{5.4}$$

Under these assumptions, for any given $\boldsymbol{\mu} \in \mathcal{D}$, if $\bar{x}$ is an optimal solution of (5.2) there exists a Lagrange multiplier $\bar{p} \in Q$ such that $(\bar{x}, \bar{p})$ satisfies

$$\begin{cases} \mathcal{J}'(\bar{x}; \boldsymbol{\mu}) + \mathcal{E}'(\bar{x}; \boldsymbol{\mu})^* \bar{p} & = 0, \quad \text{in } X' \\ \mathcal{E}(\bar{x}; \boldsymbol{\mu}) & = 0, \quad \text{in } Q', \end{cases} \tag{5.5}$$

where $\mathcal{E}'(\bar{x}; \boldsymbol{\mu})^* \in \mathcal{L}(Q, X')$ denotes the adjoint of the Fréchet derivative of the state operator. The first order optimality system (5.5) can be also expressed in more compact form as

$$\mathcal{G}(U; \boldsymbol{\mu}) = \begin{pmatrix} \mathcal{J}'(x; \boldsymbol{\mu}) + \mathcal{E}'(x; \boldsymbol{\mu})^* p \\ \mathcal{E}(x; \boldsymbol{\mu}) \end{pmatrix} = 0 \qquad \text{in } \mathcal{X}', \tag{5.6}$$

being $U = (x, p) \in \mathcal{X}$, $\mathcal{X} = X \times Q$ and $\mathcal{G}(\cdot, \cdot; \boldsymbol{\mu}) : \mathcal{X} \to \mathcal{X}'$ defined as $\mathcal{G} = \nabla \mathcal{L}$.

5.1.2 Second order sufficient optimality condition

The optimality conditions (5.5) form a nonlinear system of equations which represents the starting point to approximate the optimization problem (5.2). However, since a solution of (5.5) is not guaranteed to be a local minimizer of (5.2), we also require the following second order sufficient optimality condition:

(H4) the mappings $\mathcal{J}(\cdot;\boldsymbol{\mu})\colon X \to \mathbb{R}$ and $\mathcal{E}(\cdot;\boldsymbol{\mu})\colon Y \to Q'$ are twice continuously Fréchet differentiable with Lipschitz continuous second derivatives and the operator $\mathcal{L}_{xx}(\bar{x}, \bar{p};\boldsymbol{\mu})$ is coercive on the null space of $\mathcal{E}'(\bar{x};\boldsymbol{\mu})$, i.e. there exists a constant $\bar{\alpha} > 0$ such that

$$\langle \mathcal{L}_{xx}(\bar{x},\bar{p};\boldsymbol{\mu})d, d\rangle \geq \bar{\alpha}\|d\|_X^2, \qquad \forall d \in \ker\, \mathcal{E}'(\bar{x};\boldsymbol{\mu}). \tag{5.7}$$

Under this condition, a solution $\bar{x}$ of (5.5) is a local minimizer of (5.2).

Remark 5.1. Notice that the Lipschitz continuity of second derivatives is not actually required to ensure that $\bar{x}$ is a local minimizer. However, since it will be required to guarantee the quadratic convergence of Newton method, we incorporate this condition in assumption (H4). •

5.2 Full-order approximation

Let us first introduce suitable finite-dimensional approximation spaces $X_h \subset X$ and $Q_h \subset Q$. We then set $\mathcal{X}_h = X_h \times Q_h$ and denote with $N_h = N_{h,x} + N_{h,p}$ its dimension. Following a *discretize-then-optimize* approach, see e.g. [CH12, Gun03, HPUU09], we consider the following full-order approximation of the optimization problem (5.2): given $\boldsymbol{\mu} \in \mathcal{D}$,

$$\min_{x_h \in X_h} \ \mathcal{J}_h(x_h;\boldsymbol{\mu}) \quad \text{subject to} \quad \langle \mathcal{E}_h(x_h;\boldsymbol{\mu}), \hat{p}\rangle = 0 \quad \forall \hat{p} \in Q_h. \tag{5.8}$$

By requiring the gradient of the discrete Lagrangian

$$\mathcal{L}_h(x_h, p_h;\boldsymbol{\mu}) = \mathcal{J}_h(x_h;\boldsymbol{\mu}) + \langle \mathcal{E}_h(x_h;\boldsymbol{\mu}), p_h\rangle$$

to vanish, we obtain the following optimality conditions

$$\mathcal{G}_h(U_h;\boldsymbol{\mu}) = \begin{pmatrix} \mathcal{J}_h'(x_h;\boldsymbol{\mu}) + \mathcal{E}_h'(x_h;\boldsymbol{\mu})^* p_h \\ \mathcal{E}_h(x_h;\boldsymbol{\mu}) \end{pmatrix} = 0 \qquad \text{in } \mathcal{X}_h'. \tag{5.9}$$

Upon defining the variational form

$$G(V; W;\boldsymbol{\mu}) =_{\mathcal{X}'} \langle \mathcal{G}_h(V;\boldsymbol{\mu}), W\rangle_{\mathcal{X}} \qquad \forall V, W \in \mathcal{X}_h,$$

problem (5.9) can be equivalently expressed in weak form: given $\boldsymbol{\mu} \in \mathcal{D}$, find $U_h(\boldsymbol{\mu}) \in \mathcal{X}_h$ such that

$$G(U_h; \widehat{U};\boldsymbol{\mu}) = 0 \qquad \forall \widehat{U} \in \mathcal{X}_h. \tag{5.10}$$

For the well-posedness of the full-order approximation (5.10), it is sufficient to require the assumptions (H1)-(H4) to hold at the discrete level. In particular, denoting with $U_h = (x_h, p_h)$ a solution of (5.10), we require the derivative of the discretized state operator to be surjective, i.e.

$$\exists \bar{\lambda} > 0 \quad \text{s.t.} \quad \lambda_h(\boldsymbol{\mu}) = \inf_{\hat{p} \in Q_h} \sup_{\hat{x} \in X_h} \frac{\langle \mathcal{E}_h'(x_h;\boldsymbol{\mu})\hat{x}, \hat{p}\rangle}{\|\hat{p}\|_Q \|\hat{x}\|_X} \geq \bar{\lambda}, \tag{5.11}$$

and the Hessian of the Lagrangian to satisfy a second order sufficient optimality condition

$$\exists \bar{\alpha} > 0 \quad \text{s.t.} \quad \langle \mathcal{L}_{h,xx}(x_h, p_h;\boldsymbol{\mu})d, d\rangle \geq \bar{\alpha}\|d\|_X^2, \qquad \forall d \in X_h^0, \tag{5.12}$$

where $X_h^0 = \{d \in X_h : \langle \mathcal{E}_h'(x_h;\boldsymbol{\mu})d, \hat{p}\rangle = 0,\ \forall \hat{p} \in Q_h\}$.

Remark 5.2. Conditions (5.11)-(5.12) are equivalent to the following [XZ03, Vol00]

$$\exists \bar{\beta} > 0 \quad \text{s.t.} \quad \beta_h(U_h;\boldsymbol{\mu}) = \inf_{\widehat{U}_1 \in \mathcal{X}_h} \sup_{\widehat{U}_2 \in X_h} \frac{dG[U_h](\widehat{U}_1, \widehat{U}_2;\boldsymbol{\mu})}{\|\widehat{U}_1\|_{\mathcal{X}} \|\widehat{U}_2\|_{\mathcal{X}}} \geq \bar{\beta}, \tag{5.13}$$

where $dG[U_h](\cdot,\cdot;\boldsymbol{\mu}) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ denotes the Fréchet derivative of $G(U_h;\cdot;\boldsymbol{\mu})$. •

5.2.1 Newton method

The optimality conditions (5.9) form a finite-dimensional system of nonlinear equations. The Newton method applied to (5.9) reads: given $\boldsymbol{\mu} \in \mathcal{D}$ and an initial guess $U_h^0 \in \mathcal{X}_h$, for $k = 0, 1, \ldots$ until convergence, find $\delta U_h \in \mathcal{X}_h$ such that

$$\mathcal{G}_h'(U_h^k;\boldsymbol{\mu})\, \delta U_h = -\mathcal{G}_h(U_h^k;\boldsymbol{\mu}) \quad \text{in } \mathcal{X}_h', \tag{5.14}$$

and then set $U_h^{k+1} = U_h^k + \delta U_h$. Here $\mathcal{G}_h'(V;\boldsymbol{\mu}) \in \mathcal{L}(\mathcal{X}_h; \mathcal{X}_h')$ denotes the Fréchet derivative of $\mathcal{G}_h$ at $V \in \mathcal{X}_h$. More in detail, the Newton step (5.14) reads: find $(\delta x_h^k, \delta p_h^k) \in X_h \times Q_h$ such that

$$\begin{pmatrix} \mathcal{L}_{h,xx}(x_h^k, p_h^k;\boldsymbol{\mu}) & \mathcal{E}_h'(x_h^k;\boldsymbol{\mu})^* \\ \mathcal{E}_h'(x_h^k;\boldsymbol{\mu}) & 0 \end{pmatrix} \begin{pmatrix} \delta x_h^k \\ \delta p_h^k \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_{h,x}(x_h^k, p_h^k;\boldsymbol{\mu}) \\ \mathcal{E}_h(x_h^k;\boldsymbol{\mu}) \end{pmatrix}. \tag{5.15}$$

If the initial guess U_h^0 is sufficiently close to an optimal solution $\bar{U}_h$, assumptions (H1)-(H4) provide sufficient conditions for local quadratic convergence of Newton method, see e.g. [IK08].

The k-th step (5.14) can be equivalently formulated in weak form as: find $\delta U_h \in \mathcal{X}_h$ such that

$$dG[U_h^k](\delta U_h, \widehat{U};\boldsymbol{\mu}) = -G(U_h^k;\widehat{U};\boldsymbol{\mu}) \qquad \forall \widehat{U} \in \mathcal{X}_h. \tag{5.16}$$

5.2.2 Algebraic formulation

At the algebraic level, problem (5.10) leads to the following nonlinear system for $\mathbf{U}_h = (\mathbf{x}_h, \mathbf{p}_h) \in \mathbb{R}^{N_h}$

$$\mathbf{G}(\mathbf{U}_h;\boldsymbol{\mu}) = \begin{pmatrix} \mathbf{g}(\mathbf{x}_h;\boldsymbol{\mu}) + \mathbf{B}^T(\mathbf{x}_h;\boldsymbol{\mu})\, \mathbf{p}_h \\ \mathbf{E}(\mathbf{x}_h;\boldsymbol{\mu}) \end{pmatrix} = \mathbf{0}. \tag{5.17}$$

Here $\mathbf{g} \in \mathbb{R}^{N_{h,p}}$ is the discretized gradient of $\mathcal{J}_h$ with respect to x; $\mathbf{E} \in \mathbb{R}^{N_{h,p}}$ denotes the discretized constraint equation, while $\mathbf{B} \in \mathbb{R}^{N_{h,p} \times N_{h,x}}$ denotes its Jacobian with respect to the optimization variable. The k-th step of the Newton method applied to (5.17) can be written in compact algebraic form as follows: find $\delta \mathbf{U}_h \in \mathbb{R}^{N_h}$ such that

$$d\mathbf{G}(\mathbf{U}_h^k;\boldsymbol{\mu})\, \delta \mathbf{U}_h = -\mathbf{G}(\mathbf{U}_h^k;\boldsymbol{\mu}), \tag{5.18}$$

being $d\mathbf{G}(\cdot;\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ the matrix resulting from the discretization of the Fréchet derivative $dG[\cdot](\cdot,\cdot;\boldsymbol{\mu})$. More in detail, (5.18) reads

$$\begin{pmatrix} \mathbf{H}(\mathbf{x}_h^k, \mathbf{p}_h^k;\boldsymbol{\mu}) & \mathbf{B}^T(\mathbf{x}_h^k;\boldsymbol{\mu}) \\ \mathbf{B}(\mathbf{x}_h^k;\boldsymbol{\mu}) & 0 \end{pmatrix} \begin{pmatrix} \delta \mathbf{x}_h \\ \delta \mathbf{p}_h \end{pmatrix} = - \begin{pmatrix} \mathbf{g}(\mathbf{x}_h^k;\boldsymbol{\mu}) + \mathbf{B}^T(\mathbf{x}_h^k;\boldsymbol{\mu})\mathbf{p}_h^k \\ \mathbf{E}(\mathbf{x}_h^k;\boldsymbol{\mu}) \end{pmatrix}, \tag{5.19}$$

where $\mathbf{H} \in \mathbb{R}^{N_{h,x} \times N_{h,x}}$ denotes the discretized Hessian of the Lagrangian with respect to x; $\delta\mathbf{x}_h \in \mathbb{R}^{N_{h,x}}$ and $\delta\mathbf{p}_h \in \mathbb{R}^{N_{h,p}}$ are the the search directions in the $\mathbf{x}$ and $\mathbf{p}$ variables, respectively.

For the sake of illustration, let us consider the following example of optimization problem with quadratic cost functional and nonlinear state equation:

$$\begin{aligned} &\min_{\mathbf{x}_h} \mathcal{J}_h(\mathbf{x}_h; \boldsymbol{\mu}) = \frac{1}{2}(\mathbf{y}_h - \mathbf{y}_d(\boldsymbol{\mu}))^T \mathbf{F}(\boldsymbol{\mu})(\mathbf{y}_h - \mathbf{y}_d(\boldsymbol{\mu})) + \frac{\sigma}{2}\mathbf{u}_h^T \mathbf{N}(\boldsymbol{\mu})\mathbf{u}_h \\ &\text{s.t. } \mathbf{E}(\mathbf{y}_h, \mathbf{u}_h; \boldsymbol{\mu}) = 0, \end{aligned} \tag{5.20}$$

where $\mathbf{y}_d(\boldsymbol{\mu}) \in \mathbb{R}^{N_{h,y}}$ is a given desired state, $\mathbf{F}(\boldsymbol{\mu}) \in \mathbb{R}^{N_{h,y} \times N_{h,y}}$ is a positive semidefinite matrix defining the objective of the optimization, $\sigma > 0$ is a given penalization constant, while $\mathbf{N}(\boldsymbol{\mu}) \in \mathbb{R}^{N_{h,u} \times N_{h,u}}$ is a symmetric positive definite matrix. The associated optimality system (5.17) reads

$$\begin{pmatrix} \mathbf{F}(\boldsymbol{\mu})\mathbf{y}_h - \mathbf{F}(\boldsymbol{\mu})\mathbf{y}_d(\boldsymbol{\mu}) + \mathbf{B}_y^T(\mathbf{x}_h; \boldsymbol{\mu})\, \mathbf{p}_h \\ \sigma\mathbf{N}(\boldsymbol{\mu})\mathbf{u}_h - \mathbf{B}_u^T(\mathbf{x}_h; \boldsymbol{\mu})\, \mathbf{p}_h \\ \mathbf{E}(\mathbf{y}_h, \mathbf{u}_h; \boldsymbol{\mu}) \end{pmatrix} = \mathbf{0}, \tag{5.21}$$

where we have partitioned the Jacobian matrix of the state equation

$$\mathbf{B}(\mathbf{x}_h; \boldsymbol{\mu}) = \begin{pmatrix} \mathbf{B}_y(\mathbf{x}_h; \boldsymbol{\mu}) & \mathbf{B}_u(\mathbf{x}_h; \boldsymbol{\mu}) \end{pmatrix} \in \mathbb{R}^{N_{h,y} \times (N_{h,y} + N_{h,u})} \tag{5.22}$$

according to state and control variables. The first equation in (5.21) is referred to as adjoint equation, the second one as optimality equation, while the last is nothing but the state equation. Moreover, the Hessian matrix $\mathbf{H}$ at $(\mathbf{x}_h^k, \mathbf{p}_h^k)$ is given by

$$\mathbf{H}(\mathbf{x}_h^k, \mathbf{p}_h^k; \boldsymbol{\mu}) = \begin{pmatrix} \mathbf{F}(\boldsymbol{\mu}) + \mathbf{D}_{yy}(\mathbf{x}_h^k, \mathbf{p}_h^k; \boldsymbol{\mu}) & \mathbf{D}_{yu}(\mathbf{x}_h^k, \mathbf{p}_h^k; \boldsymbol{\mu}) \\ \mathbf{D}_{uy}(\mathbf{x}_h^k, \mathbf{p}_h^k; \boldsymbol{\mu}) & \sigma\mathbf{N}(\boldsymbol{\mu}) + \mathbf{D}_{uu}(\mathbf{x}_h^k, \mathbf{p}_h^k; \boldsymbol{\mu}) \end{pmatrix} \tag{5.23}$$

where $\mathbf{D}(\mathbf{x}_h^k, \mathbf{p}_h^k; \boldsymbol{\mu}) \in \mathbb{R}^{N_{h,x} \times N_{h,x}}$ is the matrix containing the second derivatives of the state equation, block-partitioned according to the state and control variables. In case of a linear state equation, $\mathbf{E}(\mathbf{y}_h, \mathbf{u}_h; \boldsymbol{\mu})$ takes the form

$$\mathbf{E}(\mathbf{y}_h, \mathbf{u}_h; \boldsymbol{\mu}) = \mathbf{A}(\boldsymbol{\mu})\mathbf{y}_h - \mathbf{C}(\boldsymbol{\mu})\mathbf{u}_h - \mathbf{f}(\boldsymbol{\mu}), \tag{5.24}$$

where the matrix $\mathbf{A}(\boldsymbol{\mu})$ results from the discretization of a linear PDE operator, the matrix $\mathbf{C}(\boldsymbol{\mu})$ expresses the action of the control variable, while $\mathbf{f}(\boldsymbol{\mu})$ is a given source term. With respect to (5.22) and (5.23), here the Jacobian of the state equation is independent of $\mathbf{x}_h$ and therefore its second derivatives are identically zero, i.e.

$$\mathbf{B}(\mathbf{x}_h; \boldsymbol{\mu}) = \begin{pmatrix} \mathbf{A}(\boldsymbol{\mu}) & -\mathbf{C}(\boldsymbol{\mu}) \end{pmatrix}, \qquad \mathbf{D}(\mathbf{x}_h; \mathbf{p}_h; \boldsymbol{\mu}) = 0.$$

The optimality system (5.21) thus reduces to the following linear system

$$\begin{pmatrix} \mathbf{F}(\boldsymbol{\mu}) & 0 & \mathbf{A}^T(\boldsymbol{\mu}) \\ 0 & \sigma\mathbf{N}(\boldsymbol{\mu}) & -\mathbf{C}(\boldsymbol{\mu}) \\ \mathbf{A}(\boldsymbol{\mu}) & -\mathbf{C}(\boldsymbol{\mu}) & 0 \end{pmatrix} \begin{pmatrix} \mathbf{y}_h \\ \mathbf{u}_h \\ \mathbf{p}_h \end{pmatrix} = \begin{pmatrix} \mathbf{F}(\boldsymbol{\mu})\mathbf{y}_d(\boldsymbol{\mu}) \\ 0 \\ \mathbf{f}(\boldsymbol{\mu}) \end{pmatrix}. \tag{5.25}$$

For the resolution of the linear system (5.19) several strategies can be employed, see e.g. [IK08, ABG+06] and references therein. A popular approach is based on the so called *reduced-space* (or *reduced Hessian*) methods, in which block elimination on the state and adjoint variables yields a reduced[1] system for the control variable whose matrix is the Schur complement of the optimality system. A radically alternative strategy consists of using *full-space* (also called *all-at-once*) methods, where the optimality system is solved simultaneously for the state, adjoint and control variables. Both approaches present advantages and disadvantages and require problem-tailored design of suitable preconditioners and iterative linear solvers, see e.g. [BG05a, BGL05, BS09, PBC06, SZ07, BK05, RSW10, RDW10, RW11, Zul11]. Yet, beside the choice of the favorite solution algorithm, it is well known that the numerical solution of PDE-constrained optimization problems entails large computational costs and may be very time-consuming already in the non-parametric case. Therefore, when performing the optimization process for many different parameter values or else when, for a new given configuration, the solution has to be computed in a rapid way, reducing the computational complexity is mandatory. This is why we advocate using suitable model order reduction techniques.

5.3 Reduced-order approximation

The idea of reduced basis methods is to efficiently compute an approximate solution $x_N(\boldsymbol{\mu})$ of the full-order optimization problem (5.8) belonging to a low-dimensional space X_N generated by so called *snapshots*. For the problem at hand, since the optimal solutions $x_h(\boldsymbol{\mu})$ of (5.8) are characterized as the solutions of the optimality system (5.10), two possible approaches to build a ROM can be pursued: the *reduce-then-optimize* and the *optimize-then-reduce* approach. Here we follow the second one, i.e. we build the ROM directly on the optimality system.

Let us denote with $X_N \subset X_h$ and $Q_N \subset Q_h$ suitably defined (see Sect. 5.5) low-dimensional spaces generated by full-order snapshots of the optimization and adjoint variables, respectively. We then define the trial space $\mathcal{X}_N = X_N \times Q_N$ and we denote by $\mathcal{W}_N \subset \mathcal{X}_h$ a suitable test space possibly different from $\mathcal{X}_N$. The Petrov-Galerkin reduced basis approximation of (5.10) reads: find $U_N(\boldsymbol{\mu}) \in \mathcal{X}_N$ such that

$$G(U_N; \widehat{U}; \boldsymbol{\mu}) = 0 \qquad \forall \widehat{U} \in \mathcal{W}_N. \tag{5.26}$$

The reduced optimality system (5.26) is a system of N nonlinear equations, which can be solved by means of Newton method. The generic k-th step reads: find $\delta U_N \in \mathcal{X}_N$ such that

$$dG[U_N^k](\delta U_N, \widehat{U}; \boldsymbol{\mu}) = -G(U_N^k; \widehat{U}; \boldsymbol{\mu}) \qquad \forall \widehat{U} \in \mathcal{W}_N. \tag{5.27}$$

The well-posedness of the reduced problem (5.26) as well as the convergence of Newton method are ensured by Newton-Kantorovich theorem as long as the following inf-sup condition is fulfilled

$$\exists \bar{\beta} > 0 \quad \text{s.t.} \quad \beta_N(\boldsymbol{\mu}) = \inf_{V \in \mathcal{X}_N} \sup_{W \in \mathcal{W}_N} \frac{dG[U_N](V, W; \boldsymbol{\mu})}{\|V\|_{\mathcal{X}} \|W\|_{\mathcal{X}}} \geq \bar{\beta}. \tag{5.28}$$

[1] Here *reduced* must not be understood in the sense of *reduced-order model*.

The latter is equivalent to require assumptions (H3)-(H4) to hold at the reduced level. The fulfillment of these conditions has to be taken into account in the practical construction of the reduced spaces. Not only, we require the reduced spaces $\mathcal{X}_N, \mathcal{W}_N$ to provide

(A1) a *consistent approximation* of the optimality system, meaning that, if for some $\boldsymbol{\mu} \in \mathcal{D}$ we have $U_h(\boldsymbol{\mu}) \in \mathcal{X}_N$, then $U_N(\boldsymbol{\mu}) = U_h(\boldsymbol{\mu})$;

(A2) a *Lagrangian preserving approximation*, so that (5.26) represents the gradient of the reduced Lagrangian functional $\mathcal{L}_N(\cdot,\cdot;\boldsymbol{\mu}) : X_N \times Q_N \to \mathbb{R}$ defined as

$$\mathcal{L}_N(v, q; \boldsymbol{\mu}) = \mathcal{L}_h(v, q; \boldsymbol{\mu}) \qquad \forall v \in X_N, q \in Q_N; \tag{5.29}$$

(A3) a *stable approximation* in the sense of (5.28).

Provided (A3) holds (so that (5.26) admits a unique solution), (A1) only affects $\mathcal{X}_N$. On the other hand, the fulfillment of (A2) and (A3) only depends on the choice of $\mathcal{W}_N$. In particular, the ROM (5.26) enjoys property (A2) if and only if a Galerkin projection is employed, i.e. $\mathcal{W}_N = \mathcal{X}_N$. Indeed, requiring the gradient of (5.29) to vanish, we obtain

$$\langle \nabla \mathcal{L}_h(U_N(\boldsymbol{\mu}); \boldsymbol{\mu}), \widehat{U} \rangle = 0 \qquad \forall \widehat{U} \in \mathcal{X}_N, \tag{5.30}$$

which is equivalent to (5.26) if and only if $\mathcal{W}_N = \mathcal{X}_N$. In this case, however, as the Galerkin projection does not automatically generate a stable ROM, property (A3) has to be proved for the problem at hand. Conversely, a least-squares projection would automatically satisfy condition (A3) *and* violate property (A2). Thus, in the following we only consider Galerkin projections, i.e. we set $\mathcal{W}_N = \mathcal{X}_N$.

5.3.1 Algebraic formulation

Let us denote by $\mathbf{V}_x \in \mathbb{R}^{N_{h,x} \times N_x}$ and $\mathbf{V}_p \in \mathbb{R}^{N_{h,p} \times N_p}$ suitable bases for the reduced spaces X_N and Q_N, respectively; a basis for $\mathcal{X}_N$ is therefore given by

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_x & 0 \\ 0 & \mathbf{V}_p \end{pmatrix} \in \mathbb{R}^{(N_{h,x}+N_{h,p}) \times (N_x+N_p)}. \tag{5.31}$$

Problem (5.26) is thus equivalent to the following nonlinear system of $N_x + N_p$ equations

$$\mathbf{V}^T \mathbf{G}(\mathbf{V}\mathbf{U}_N; \boldsymbol{\mu}) = \begin{pmatrix} \mathbf{V}_x^T \mathbf{g}(\mathbf{V}_x \mathbf{x}_N; \boldsymbol{\mu}) + \mathbf{V}_x^T \mathbf{B}^T(\mathbf{V}_x \mathbf{x}_N; \boldsymbol{\mu})\, \mathbf{V}_p \mathbf{p}_N \\ \mathbf{V}_p^T \mathbf{E}(\mathbf{V}_x \mathbf{x}_N; \boldsymbol{\mu}) \end{pmatrix} = \mathbf{0}, \tag{5.32}$$

where $\mathbf{x}_N$ and $\mathbf{p}_N$ denote the vectors of coefficients in the expansions of x_N and p_N with respect to the reduced bases. As already mentioned, thanks to the Galerkin projection, (5.32) represents the optimality system of the following reduced Lagrangian functional

$$\mathcal{L}_N(\mathbf{x}_N, \mathbf{p}_N; \boldsymbol{\mu}) = \mathcal{L}_h(\mathbf{V}_x \mathbf{x}_N, \mathbf{V}_p \mathbf{p}_N; \boldsymbol{\mu}) = \mathcal{J}_h(\mathbf{V}_x \mathbf{x}_N; \boldsymbol{\mu}) + \mathbf{p}_N^T \mathbf{V}_p^T \mathbf{E}(\mathbf{V}_x \mathbf{x}_N; \boldsymbol{\mu}). \tag{5.33}$$

The Newton step (5.27) is equivalent to the following linear system

$$d\mathbf{G}_N(\mathbf{U}_N^k; \boldsymbol{\mu})\, \delta \mathbf{U}_N = -\mathbf{G}_N(\mathbf{U}_N^k; \boldsymbol{\mu}), \tag{5.34}$$

where

$$d\mathbf{G}_N(\mathbf{U}_N^k;\boldsymbol{\mu}) = \mathbf{V}^T d\mathbf{G}(\mathbf{V}\mathbf{U}_N^k;\boldsymbol{\mu})\mathbf{V} = \begin{pmatrix} \mathbf{H}_N(\mathbf{x}_N^k,\mathbf{p}_N^k;\boldsymbol{\mu}) & \mathbf{B}_N^T(\mathbf{x}_N^k;\boldsymbol{\mu}) \\ \mathbf{B}_N(\mathbf{x}_N^k;\boldsymbol{\mu}) & 0 \end{pmatrix}, \tag{5.35}$$

$$\mathbf{G}_N(\mathbf{U}_N^k;\boldsymbol{\mu}) = \mathbf{V}^T\mathbf{G}(\mathbf{V}\mathbf{U}_N^k;\boldsymbol{\mu}) = \begin{pmatrix} \mathbf{g}_N(\mathbf{x}_N^k;\boldsymbol{\mu}) + \mathbf{B}_N^T(\mathbf{x}_N^k;\boldsymbol{\mu})\mathbf{p}_N^k \\ \mathbf{E}_N(\mathbf{x}_N^k;\boldsymbol{\mu}) \end{pmatrix}, \tag{5.36}$$

and the reduced matrices and vectors are given by:

$$\mathbf{H}_N(\mathbf{x}_N^k,\mathbf{p}_N^k;\boldsymbol{\mu}) = \mathbf{V}_x^T\,\mathbf{H}(\mathbf{V}_x\mathbf{x}_N^k,\mathbf{V}_p\mathbf{p}_N^k;\boldsymbol{\mu})\,\mathbf{V}_x, \qquad \mathbf{B}_N(\mathbf{x}_N^k;\boldsymbol{\mu}) = \mathbf{V}_p^T\,\mathbf{B}(\mathbf{V}_x\mathbf{x}_N^k;\boldsymbol{\mu})\,\mathbf{V}_x,$$
$$\mathbf{E}_N(\mathbf{x}_N^k;\boldsymbol{\mu}) = \mathbf{V}_p^T\,\mathbf{E}(\mathbf{V}_x\mathbf{x}_N^k;\boldsymbol{\mu}), \qquad \mathbf{g}_N(\mathbf{x}_N^k;\boldsymbol{\mu}) = \mathbf{V}_x^T\,\mathbf{g}(\mathbf{V}_x\mathbf{x}_N^k;\boldsymbol{\mu}).$$

5.3.2 Computational efficiency: offline-online decomposition

The (hopefully small) dimension $N \ll N_h$ of the linear system (5.34) to be solved at each Newton step does not warren substantial computational savings, as the assembly of the reduced Jacobian matrix and residual vector still involves computations whose complexity depends on N_h. However, if the state equation features a low-order polynomial nonlinearity and the parametric dependence is affine, the assembly of both the residual $\mathbf{G}_N(\cdot;\boldsymbol{\mu})$ and the matrix $d\mathbf{G}_N(\cdot;\boldsymbol{\mu})$ admits an efficient offline-online decomposition. For instance, in the case of quadratic nonlinearity (as for the Navier-Stokes equations, see Sect. 5.9) we can express the residual in the form

$$\mathbf{G}(\mathbf{W};\boldsymbol{\mu}) = \widetilde{\mathbf{G}}(\mathbf{W},\mathbf{W};\boldsymbol{\mu}),$$

$\widetilde{\mathbf{G}}(\cdot;\cdot;\boldsymbol{\mu})$ being linear w.r.t. the first two arguments. Then, thanks to the affine parametric dependence, we can assume the reduced residual to be expressed as

$$\mathbf{G}_N(\mathbf{W}_N;\boldsymbol{\mu}) = \sum_{q=1}^{Q_g} \theta_q^g(\boldsymbol{\mu}) \sum_{i,j=1}^{N} \mathbf{W}_{Ni}\mathbf{W}_{Nj}\,\mathbf{V}^T\widetilde{\mathbf{G}}_q(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j), \tag{5.37}$$

for some suitable smooth functions $\theta_q^g : \mathcal{D} \to \mathbb{R}$ and $\boldsymbol{\mu}$-independent vectors $\widetilde{\mathbf{G}}_q(\cdot,\cdot) \in \mathbb{R}^{N_h}$ (denoting by $\boldsymbol{\phi}_i$ the elements of the basis $\mathbf{V}$). Similarly, we can express the Jacobian matrix as

$$d\mathbf{G}_N(\mathbf{W}_N;\boldsymbol{\mu}) = \sum_{q=1}^{Q_d} \theta_q^d(\boldsymbol{\mu}) \sum_{i=1}^{N} \mathbf{W}_{Ni}\,\mathbf{V}^T d\mathbf{G}_q(\boldsymbol{\phi}_i)\mathbf{V}, \tag{5.38}$$

for suitable smooth functions $\theta_q^d : \mathcal{D} \to \mathbb{R}$ and $\boldsymbol{\mu}$-independent matrices $d\mathbf{G}_q(\boldsymbol{\phi}_n) \in \mathbb{R}^{N_h \times N_h}$. The NQ_d reduced matrices $\mathbf{V}^T d\mathbf{G}_q(\boldsymbol{\phi}_i)\mathbf{V}$ and the N^2Q_g vectors $\mathbf{V}^T\widetilde{\mathbf{G}}_q(\boldsymbol{\phi}_i,\boldsymbol{\phi}_j)$ can be precomputed offline, so that online the reduced problem can be assembled and solved with complexity independent of N_h.

On the other hand, if the problem features a higher (or nonpolynomial) nonlinearity and possibly a nonaffine parametric dependence, we must resort to suitable system approximation techniques, see Sect. 3.1. By these approaches, we aim at obtaining an affine approximation of the residual of the form

$$\mathbf{G}(\mathbf{W}_N;\boldsymbol{\mu}) \approx \sum_{m=1}^{M} \alpha_m^g(\boldsymbol{\mu};\mathbf{W}_N)\,\mathbf{G}_m, \tag{5.39}$$

being $\{\mathbf{G}_m\}_{m=1}^M$ a basis for a suitable subspace of $\mathcal{M}_{\mathbf{G}} = \{\mathbf{G}(\mathbf{V}\mathbf{U}_N(\boldsymbol{\mu});\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathcal{D}\}$ and $\alpha_m^g(\cdot;\cdot)$ some interpolation coefficients to be determined. In Sect. 6.2 we shall present a successful application of MDEIM and DEIM to a linear-quadratic optimal control problem posed over a nonaffinely parametrized domain.

Remark 5.3. Note that (5.37) can be written in the form (5.39) with $M = Q_g N^2$, $\mathbf{G}_m = \widetilde{\mathbf{G}}_q(\boldsymbol{\phi}_j, \boldsymbol{\phi}_j)$ and $\alpha_m^g(\boldsymbol{\mu}; \mathbf{W}_N) = \theta_q^g(\boldsymbol{\mu})\mathbf{W}_{Ni}\mathbf{W}_{Nj}$. •

5.4 A posteriori error estimates

In order to derive an a posteriori error estimate for the error on the state, control and adjoint variables, we take advantage of Brezzi-Rappaz-Raviart theory [BRR80, CR97], as applied originally in [IR98a, VP05] to the reduced basis approximation of the Navier-Stokes equations.

We start by introducing all the involved quantities. First,

$$\varepsilon_N(\boldsymbol{\mu}) = \|G(U_N;\cdot;\boldsymbol{\mu})\|_{\mathcal{X}_h'} = \sup_{W\in\mathcal{X}_h} \frac{G(U_N;W;\boldsymbol{\mu})}{\|W\|_{\mathcal{X}}} \tag{5.40}$$

is the dual norm of the residual of the optimality system (5.26). Next, we need the inf-sup constant $\beta_h^N(\boldsymbol{\mu})$ of the Fréchet derivative dG at U_N

$$\beta_h^N(\boldsymbol{\mu}) = \inf_{V\in\mathcal{X}_h} \sup_{W\in\mathcal{X}_h} \frac{dG[U_N(\boldsymbol{\mu})](V,W;\boldsymbol{\mu})}{\|V\|_{\mathcal{X}}\|W\|_{\mathcal{X}}}, \tag{5.41}$$

and an upper bound $K_h^N(\boldsymbol{\mu})$ for its Lipschitz constant[2] such that

$$\|dG[U_N(\boldsymbol{\mu})](\cdot,\cdot;\boldsymbol{\mu}) - dG[V](\cdot,\cdot;\boldsymbol{\mu})\|_{\mathcal{L}(\mathcal{X}_h,\mathcal{X}_h')} \le K_h^N(\boldsymbol{\mu})\|U_N(\boldsymbol{\mu}) - V\|_{\mathcal{X}} \tag{5.42}$$

for all $V \in \overline{B}\left(U_N(\boldsymbol{\mu}), 2\varepsilon_N(\boldsymbol{\mu})/\beta_h^N(\boldsymbol{\mu})\right)$. Here $\overline{B}(x,\alpha) \subset \mathcal{X}_h$ denotes a closed ball with center x and radius α. Finally, we define the *proximity indicator*

$$\tau_N(\boldsymbol{\mu}) = \frac{4K_h^N(\boldsymbol{\mu})\varepsilon_N(\boldsymbol{\mu})}{\left(\beta_h^N(\boldsymbol{\mu})\right)^2}. \tag{5.43}$$

A straightforward application of [CR97, Theorem 2.1] (see also [IR98a, VP05]) provides the following bound for the norm of the error $E_N(\boldsymbol{\mu}) = U_h(\boldsymbol{\mu}) - U_N(\boldsymbol{\mu})$.

Proposition 5.1. *If $\tau_N(\boldsymbol{\mu}) < 1$, then there exists a unique solution*

$$U_h(\boldsymbol{\mu}) \in B\left(U_N(\boldsymbol{\mu}), 2\varepsilon_N(\boldsymbol{\mu})/\beta_h^N(\boldsymbol{\mu})\right)$$

of (5.10)*. Moreover,*

$$\|E_N(\boldsymbol{\mu})\|_{\mathcal{X}} \le \Delta_N(\boldsymbol{\mu}) = \frac{2}{2-\tau_N(\boldsymbol{\mu})}\,\frac{\varepsilon_N(\boldsymbol{\mu})}{\beta_h^N(\boldsymbol{\mu})}. \tag{5.44}$$

[2]The Lipschitz continuity of dG follows from assumption (H2).

Remark 5.4. If the state equation is linear, we recover the error estimate

$$\|E_N(\boldsymbol{\mu})\|_{\mathcal{X}} \leq \Delta_N^L(\boldsymbol{\mu}) = \frac{\varepsilon_N(\boldsymbol{\mu})}{\beta_h(\boldsymbol{\mu})}$$

obtained in [NRMQ13] by means of Babuška stability theory, which is valid for all $N \geq 1$, since $K_h^N(\boldsymbol{\mu}) = 0$ and thus $\tau_N(\boldsymbol{\mu}) = 0$. Moreover, in the nonlinear case, if the ROM is consistent, $\varepsilon_N(\boldsymbol{\mu})$ tends to zero as N increases and therefore also $\tau_N(\boldsymbol{\mu})$ tends to zero, so that the nonlinear error estimator $\Delta_N(\boldsymbol{\mu})$ reduces to $\Delta_N^L(\boldsymbol{\mu})$. The latter can thus be conveniently used as error indicator in place of $\Delta_N(\boldsymbol{\mu})$. •

Remark 5.5. From an algebraic standpoint, for $\tau_N(\boldsymbol{\mu}) < 1$ the error estimate (5.44) reads

$$\|\mathbf{U}_h(\boldsymbol{\mu}) - \mathbf{V}\mathbf{U}_N(\boldsymbol{\mu})\|_{\mathbf{X}} \leq \frac{2}{\sigma_{\min}(\mathbf{X}^{-1/2} d\mathbf{G}(\mathbf{V}\mathbf{U}_N(\boldsymbol{\mu}); \boldsymbol{\mu})\mathbf{X}^{-1/2})} \|\mathbf{G}(\mathbf{V}\mathbf{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu})\|_{\mathbf{X}^{-1}}. \tag{5.45}$$

•

5.4.1 Efficient evaluation of the error estimate

The error estimate (5.44)-(5.45) admits the usual efficient offline-online decomposition [RHP08], that we briefly summarize here. We start by evaluating the dual norm of the residual $\varepsilon_N(\boldsymbol{\mu})$ efficiently, thanks to the affine decomposition (5.39):

$$\varepsilon_N(\boldsymbol{\mu})^2 = \|\mathbf{G}(\mathbf{V}\mathbf{U}_N; \boldsymbol{\mu})\|_{\mathbf{X}^{-1}}^2 = \sum_{m_1, m_2 = 1}^{M} \underbrace{\alpha_{m_1}^g(\boldsymbol{\mu}; \mathbf{U}_N)\, \alpha_{m_2}^g(\boldsymbol{\mu}; \mathbf{U}_N)}_{\text{online}} \underbrace{\mathbf{G}_{m_1}^T \mathbf{X}^{-1} \mathbf{G}_{m_2}}_{\text{offline}},$$

$\mathbf{X} \in \mathbb{R}^{N_h \times N_h}$ being a matrix realizing the $\mathcal{X}_h$-norm.

The second step involves the evaluation of the stability factor $\beta_h^N(\boldsymbol{\mu})$, for which we provide the heuristic approximation presented in Sect. 2.5.2. As a result, $\beta_h^N(\boldsymbol{\mu})$ is approximated by a surrogate $\beta_I(\boldsymbol{\mu})$ whose online evaluation has a computational complexity independent of the dimension N_h.

Finally, we have to provide an upper bound $K_h^N(\cdot; \boldsymbol{\mu})$ to the Lipshitz constant of dG, which however depends on the problem at hand. We shall see an example when dealing with the optimal control of Navier-Stokes equations in Sect. 5.9.

5.4.2 Error estimate on the cost functional

In order to obtain a bound for the error on the cost functional, we combine the error bound (5.44) with some results from goal-oriented a posteriori error analysis. Let us first define the following estimator

$$\Delta_N^{\mathcal{J}} = \frac{1}{2} \Delta_N(\boldsymbol{\mu})\, \varepsilon_N(\boldsymbol{\mu}). \tag{5.46}$$

Proposition 5.2. *For the reduced basis approximation* (5.26) *of the full-order problem* (5.10) *with nonlinear state equation and quadratic cost functional, if* $\tau_N(\boldsymbol{\mu}) < 1$ *there holds*

$$|\mathcal{J}_h(x_h; \boldsymbol{\mu}) - \mathcal{J}_h(x_N; \boldsymbol{\mu})| \leq \Delta_N^{\mathcal{J}}(\boldsymbol{\mu}) + |\mathcal{R}(E_N(\boldsymbol{\mu}); \boldsymbol{\mu})|, \tag{5.47}$$

where the remainder term $\mathcal{R}(E_N(\boldsymbol{\mu});\boldsymbol{\mu})$ *can be estimated by*

$$|\mathcal{R}(E_N(\boldsymbol{\mu});\boldsymbol{\mu})| \leq \sup_{W\in[U_N,U_h]} |d^2G[W](E_N(\boldsymbol{\mu}),E_N(\boldsymbol{\mu}),E_N(\boldsymbol{\mu});\boldsymbol{\mu})|. \tag{5.48}$$

Proof. We adapt here the result proved in [BKR00, Prop. 6.1]. If the ROM (5.26) preserves the Lagrangian structure (assumption A2), we have that $\mathcal{J}_h(x_h;\boldsymbol{\mu}) - \mathcal{J}_h(x_N;\boldsymbol{\mu}) = \mathcal{L}_h(U_h;\boldsymbol{\mu}) - \mathcal{L}_h(U_N;\boldsymbol{\mu})$. By applying the mean value theorem we then obtain,

$$\mathcal{L}_h(U_h;\boldsymbol{\mu}) - \mathcal{L}_h(U_N;\boldsymbol{\mu}) = \int_0^1 \nabla\mathcal{L}_h(U_h + s(U_N - U_h);\boldsymbol{\mu})(E_N)\,ds.$$

Approximating the integral with the trapezoidal rule and by the problem statement (5.10) we get,

$$\mathcal{L}_h(U_h;\boldsymbol{\mu}) - \mathcal{L}_h(U_N;\boldsymbol{\mu}) = \tfrac{1}{2}\nabla\mathcal{L}_h(U_N;\boldsymbol{\mu})(E_N) + \mathcal{R}(E_N(\boldsymbol{\mu});\boldsymbol{\mu}),$$

where the remainder term $\mathcal{R}$ is given by

$$\mathcal{R}(E_N;\boldsymbol{\mu}) = -\tfrac{1}{12}\nabla_h^3\mathcal{L}(\hat{W};\boldsymbol{\mu})(E_N,E_N,E_N;\boldsymbol{\mu}) = -\tfrac{1}{12}d^2G[\hat{W}](E_N,E_N,E_N;\boldsymbol{\mu}),$$

for a suitable $\hat{W}$ lying on the segment whose extremes are U_h and U_N. Then, since $\nabla\mathcal{L}_h(U_N;\boldsymbol{\mu})(E_N) = G(U_N;E_N);\boldsymbol{\mu})$, if $\tau_N(\boldsymbol{\mu}) < 1$ we can bound the first term as,

$$|\nabla\mathcal{L}_h(U_N;\boldsymbol{\mu})(E_N)| = |G(U_N;E_N);\boldsymbol{\mu})| \leq \varepsilon_N(\boldsymbol{\mu})\|E_N(\boldsymbol{\mu})\|_{\mathcal{X}} \leq \varepsilon_N(\boldsymbol{\mu})\Delta_N(\boldsymbol{\mu}),$$

which yields (5.47). □

Remark 5.6. As for $\Delta_N(\boldsymbol{\mu})$, if the state equation is linear we recover the error estimate

$$|\mathcal{J}_h(x_h;\boldsymbol{\mu}) - \mathcal{J}_h(x_N;\boldsymbol{\mu})| \leq \frac{1}{2}\Delta_N^L(\boldsymbol{\mu})\varepsilon_N(\boldsymbol{\mu}),$$

obtained in [NRMQ13] by means of Babuška stability theory, since $\Delta_N(\boldsymbol{\mu}) \equiv \Delta_N^L(\boldsymbol{\mu})$ and the remainder term is identically zero. •

5.4.3 Error estimate for the control variable

An alternative bound for the error in the control variable can be obtained by considering the following (equivalent) reduced-space formulation of the optimization problem (5.8)

$$\min_{u_h\in\mathcal{U}_h} \widehat{\mathcal{J}}_h(u_h;\boldsymbol{\mu}) = \mathcal{J}_h(y_h(u_h;\boldsymbol{\mu}),u_h;\boldsymbol{\mu}), \tag{5.49}$$

where we have denoted by $u_h \mapsto y_h(u_h;\boldsymbol{\mu})$ the solution operator of $\mathcal{E}_h(y_h,u_h;\boldsymbol{\mu}) = 0$.

The first order optimality condition for the unconstrained optimization problem (5.49) reads: given $\boldsymbol{\mu}\in\mathcal{D}$, find $u_h\in\mathcal{U}_h$ such that

$$\widehat{\mathcal{J}}_h'(u_h;\boldsymbol{\mu}) = 0 \ \text{ in } \mathcal{U}_h'. \tag{5.50}$$

The first derivative of $\widehat{\mathcal{J}}_h$ is given by

$$\widehat{\mathcal{J}}_h'(u_h;\boldsymbol{\mu}) = \mathcal{E}_{h,u}(y_h,u_h;\boldsymbol{\mu})^* p_h + \mathcal{J}_{h,u}(y_h,u_h;\boldsymbol{\mu}),$$

where $y_h = y_h(u_h;\boldsymbol{\mu})$, and $p_h = p_h(u_h;\boldsymbol{\mu})$ satisfies the adjoint equation

$$\mathcal{E}_{h,y}(y_h, u_h;\boldsymbol{\mu})^* p_h = -\mathcal{J}_{h,y}(y_h, u_h;\boldsymbol{\mu}).$$

By applying Brezzi-Rappaz-Raviart theory (or equivalently Newton-Kantorovich theorem, see e.g. [Zei85]) to the nonlinear problem (5.50) we obtain the following bound[3]

$$\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_{\mathcal{U}} \leq \frac{2}{\lambda_h^N(\boldsymbol{\mu})}\|\widehat{\mathcal{J}}_h'(u_N(\boldsymbol{\mu});\boldsymbol{\mu})\|_{\mathcal{U}'}, \tag{5.51}$$

where

$$\lambda_h^N(\boldsymbol{\mu}) = \inf_{v\in\mathcal{U}_h} \frac{\langle \widehat{\mathcal{J}}_h''(u_N(\boldsymbol{\mu});\boldsymbol{\mu})v, v\rangle}{\|v\|_{\mathcal{U}}^2}$$

is the coercivity constant of the Hessian $\widehat{\mathcal{J}}_h''$. For any $v_h \in \mathcal{U}_h$, the latter is given by [HPUU09, IK08]

$$\widehat{\mathcal{J}}_h''(v_h;\boldsymbol{\mu}) = T_h(v_h;\boldsymbol{\mu})^*\ \mathcal{L}_{h,xx}(y_h(v_h;\boldsymbol{\mu}), v_h, p(v_h;\boldsymbol{\mu});\boldsymbol{\mu})\ T_h(v_h;\boldsymbol{\mu}).$$

with

$$T_h(v_h;\boldsymbol{\mu}) = \begin{pmatrix} -\mathcal{E}_{h,y}(y_h(v_h;\boldsymbol{\mu}), v_h)^{-1}\mathcal{E}_{h,u}(y_h(v_h;\boldsymbol{\mu}), v_h) \\ I_{\mathcal{U}_h} \end{pmatrix} \in \mathcal{L}(\mathcal{U}_h, Y_h \times \mathcal{U}_h)$$

and $I_{\mathcal{U}_h} : \mathcal{U}_h \to \mathcal{U}_h$ being the identity operator.

The error bound (5.51) was already obtained in different forms in [HV05, HV08, TV09, KV12, KTV13, KG14a, DH13]. However, since $\widehat{\mathcal{J}}_h'(u_N(\boldsymbol{\mu});\boldsymbol{\mu})$ is an implicit operator, we cannot set up a straightforward offline-online strategy for the computation of its norm. In fact, from an algebraic standpoint (5.51) becomes

$$\|\mathbf{u}_h(\boldsymbol{\mu}) - \mathbf{V}_u\mathbf{u}_N(\boldsymbol{\mu})\|_{\mathbf{X}} \leq \frac{2}{\lambda_{\min}(\mathbf{X}_u^{-1/2} d\widehat{\mathbf{G}}(\mathbf{V}_u\mathbf{u}_N(\boldsymbol{\mu});\boldsymbol{\mu})\mathbf{X}_u^{-1/2})}\|\widehat{\mathbf{G}}(\mathbf{V}_u\mathbf{u}_N(\boldsymbol{\mu});\boldsymbol{\mu})\|_{\mathbf{X}_u^{-1}}, \tag{5.52}$$

where $\mathbf{X}_u \in \mathbb{R}^{N_{h,u}\times N_{h,u}}$ is a matrix norm for $\mathcal{U}_h$ and $\mathbf{V}_u \in \mathbb{R}^{N_{h,u}\times N_u}$ is a basis for U_N. Moreover, $\widehat{\mathbf{G}}(\cdot;\boldsymbol{\mu})$ denotes the algebraic counterpart of the reduced gradient $\widehat{\mathcal{J}}_h'(\cdot;\boldsymbol{\mu})$, while $d\widehat{\mathbf{G}}(\cdot;\boldsymbol{\mu})$ is the algebraic representation of the reduced Hessian $\widehat{\mathcal{J}}''(\cdot;\boldsymbol{\mu})$. In the case of problem (5.20), for any $\mathbf{v}_h \in \mathbb{R}^{N_{h,u}}$ they are given by, respectively:

$$\widehat{\mathbf{G}}(\mathbf{v}_h) = \sigma\mathbf{N}\mathbf{v}_h + \mathbf{B}_u^T\mathbf{B}_y^{-T}\mathbf{F}\,\mathbf{y}_h(\mathbf{v}_h),$$

$$d\widehat{\mathbf{G}}(\mathbf{v}_h) = \sigma\mathbf{N} + \mathbf{D}_{uu} + \mathbf{B}_u^T\mathbf{B}_y^{-T}(\mathbf{F} + \mathbf{D}_{yy})\mathbf{B}_y^{-1}\mathbf{B}_u - \mathbf{B}_u^T\mathbf{B}_y^{-T}\mathbf{D}_{yu} - \mathbf{D}_{uy}\mathbf{B}_y^{-1}\mathbf{B}_u,$$

where we have omitted the $\boldsymbol{\mu}$, $\mathbf{v}_h$ and $\mathbf{y}_h$-dependence in the matrices for the sake of clarity. Moreover, we have denoted by $\mathbf{v}_h \mapsto \mathbf{y}_h(\mathbf{v}_h)$ the solution operator of the state equation for a given control $\mathbf{v}_h$.

Therefore, the main disadvantage of working in the control space is that the involved operators become:

[3] More rigorously, the bound only holds if $u_N(\boldsymbol{\mu})$ satisfies a *proximity condition* similar to the one of Proposition 5.1.

1. *non-sparse*: $\widehat{\mathbf{G}}(\mathbf{V}_u\mathbf{u}_N)$ cannot be formed explicitly, rather we can compute its action on a vector in the control space by performing a state ($\mathbf{y}_h(\mathbf{V}_u\mathbf{u}_N)$) and an adjoint ($\mathbf{B}_y^{-T}$) full-order solve;
2. *nonaffinely parametrized*: even if all matrices and vectors in (5.20) admit an affine decomposition, the vector $\widehat{\mathbf{G}}$ and the matrix $d\widehat{\mathbf{G}}$ cannot be expressed as, e.g., in (5.37)-(5.38).

In particular, the residual $\widehat{\mathbf{G}}$ becomes a non-sparse and nonaffine operator, so that the usual offline-online computational procedure cannot be performed. A possible way to obtain an efficiently computable error estimate is to bound the norm of the residual in terms of quantities depending explicitly on the full and reduced-order state and adjoint solutions, as done in [GK11, KG14a]. Here, we rely instead on the *full-space* estimates (5.44)-(5.47).

5.5 Reduced bases construction

We still need to specify how to construct the reduced space $\mathcal{X}_N$ and its transformation matrix $\mathbf{V}$ (defined in (5.31)). Depending on the dimension of the control space $\mathcal{U}$, different strategies can be employed to construct $\mathcal{X}_N$. In particular:

1. if $\mathcal{U}$ (and thus $\mathcal{U}_h$) is finite and low-dimensional, say $\mathcal{U} = \mathbb{R}^C$ for some $C \geq 1$, then we can set $\mathcal{U}_N = \mathcal{U}$, i.e no control reduction is performed. Moreover, the C components of the control variable $u \in \mathcal{U}$ can be treated themselves as parameters for the purpose of reduction;
2. if instead $\mathcal{U}$ is infinite-dimensional and thus $\mathcal{U}_h$ is possibly high-dimensional, a suitable reduction of the control variable has to be carried out in order to generate a low-dimensional space $\mathcal{U}_N \subset \mathcal{U}_h$.

We further address these aspects in the next sections.

5.5.1 Low-dimensional control spaces

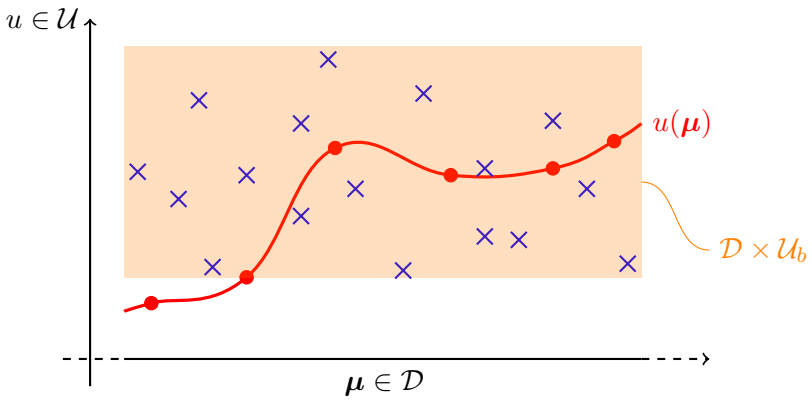


Fig. 5.1 RB spaces construction in the case of low dimensional controls. If the control variables are considered as parameters, offline we sample from the product space $\mathcal{D} \times \mathcal{U}_b$ (blue crosses); alternatively, we could sample only from $\mathcal{D}$ to build a basis of optimal solutions (red circles).

As anticipated, in the case of low-dimensional control variables, a control reduction is not required. Moreover, we can treat in the offline phase the control variables as input parameters of the state equation. The problem is thus cast in the form of a parametric optimization problem, see Sect. 1.3. However, since in our framework $\mathcal{U}$ is an unbounded set, we first have to (a priori) assume that $u \in \mathcal{U}_b$ for some bounded $\mathcal{U}_b \subset \mathcal{U}$. Ideally, $\mathcal{U}_b$ should contain the entire set of optimal controls $\{u(\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathcal{D}\}$ solutions of (5.1). Then, the state and adjoint reduced spaces Y_N and Q_N can be generated by means of either POD or greedy algorithms as

$$Y_N \subset \operatorname{span}\left\{y_h^i\right\}_{i=1}^{N_s}, \qquad Q_N \subset \operatorname{span}\left\{p_h^i\right\}_{i=1}^{N_s},$$

where the i-th state snapshot $y_h^i = y_h(u^i; \boldsymbol{\mu}^i)$ is the solution of the following (nonlinear) state equation

$$\mathcal{E}_h(y_h^i, u^i; \boldsymbol{\mu}^i) = 0,$$

and the i-th adjoint snapshot $p_h^i = p_h(u^i; \boldsymbol{\mu}^i)$ is the solution of the following (linear) adjoint equation

$$\mathcal{E}_{h,y}(y_h^i, u^i; \boldsymbol{\mu}^i)^* p_h^i = -\mathcal{J}_{h,y}(y_h^i, u^i; \boldsymbol{\mu}^i).$$

The $N_s \geq N$ points $(\boldsymbol{\mu}^i, u^i)$ (see Fig. 5.1) are either a priori chosen by a proper sampling of the space $\mathcal{D} \times \mathcal{U}_b$ in the case of POD or iteratively selected when using the greedy algorithm.

Following this procedure, the offline phase does not require the solution of any full-order optimization problem (5.8), since only uncoupled state and adjoint solves are performed. Unfortunately, this strategy has some significant drawbacks:

- effectively sampling the $(P+C)$-dimensional space $\mathcal{D}\times\mathcal{U}_b$ can be far more challenging than sampling only $\mathcal{D}$;
- the offline training phase is blind to the purpose of the optimization, possibly yielding a ROM of unnecessary large dimension;
- conversely, if $\mathcal{U}_b$ does not contain the entire set of optimal controls, the resulting ROM might be unable to correctly predict the optimal solution in some (untrained) portion of the parameter domain.

An alternative strategy which allows to overcome these shortcomings – yet increasing the offline costs – consists in generating a basis made of optimal state and adjoint solutions simultaneously (see Fig. 5.1). This is precisely the approach we follow in the case of high-dimensional control variables, the low-dimensional one being a particular instance.

5.5.2 High-dimensional control spaces

In the case of high-dimensional control variables, a suitable reduction of the control space $\mathcal{U}_h$ has to be carried out. To this end, we propose to simultaneously build a reduced basis $\mathbf{V}$ made of optimal state, adjoint and control solutions, i.e. solutions of the optimization problem (5.26) for some selected parameter samples.

The first option is to rely on the greedy algorithm [PRV$^+$02, RHP08] guided by the a posteriori error estimate (5.44). At each iteration N, we select the parameter $\boldsymbol{\mu}^{N+1}$

which maximizes $\Delta_N(\boldsymbol{\mu})$ over a training set $\Xi_{\text{train}} \subset \mathcal{D}$, and then enrich the current basis $\mathbf{V}$ using $\mathbf{U}_h(\boldsymbol{\mu}^N)$. The algorithm stops when a desired accuracy (or a maximum number of iterations) is reached. The procedure requires to solve only N full-order optimization problems; however the update (at each iteration) of the ingredients required to evaluate the dual norm of the residual can highly affect the offline costs (both in terms of computational time and memory storage), especially for large N, Q_d, Q_g.

An alternative approach relies on proper orthogonal decomposition, see e.g. [Sir87, HLB96]. First, a set of N_s snapshots $\{\mathbf{U}_h(\boldsymbol{\mu}^n)\}_{n=1}^{N_s}$ is computed for some configurations $\boldsymbol{\mu}^n \in \mathcal{D}$ (selected either a priori guided by physical intuition or by sampling techniques like latin hypercube sampling (LHS) or sparse grids). Then, the basis $\mathbf{V}$ is constructed by retaining the first N left singular vectors in the singular value decomposition of the snapshot matrix $S = [\mathbf{U}_h(\boldsymbol{\mu}^1) \cdots \mathbf{U}_h(\boldsymbol{\mu}^{N_s})]$. With respect to the previous option, here we usually have to solve a higher number ($N_s > N$) of full-order optimization problems. However, since the error estimate in this case only serves for the online certification, its ingredients are assembled only once and for all after the ROM construction. Then, the error estimate can be evaluated over a test sample $\Xi_{\text{test}} \subset \mathcal{D}$ to check whether the maximum (or average) error is below a desired tolerance. If not, the POD basis can be suitably enriched by either including some of the discarded singular vectors or computing new snapshots.

In both cases, we end up with a basis $\mathbf{V}$ block-partitioned according to the state, control and adjoint variables, i.e.

$$\mathbf{V} = \begin{pmatrix} \mathbf{V}_y & 0 & 0 \\ 0 & \mathbf{V}_u & 0 \\ 0 & 0 & \mathbf{V}_p \end{pmatrix} \in \mathbb{R}^{(N_{h,y}+N_{h,u}+N_{h,p})\times(N_y+N_u+N_p)}.$$

Moreover, both approaches are suitable also when the control variable is already low-dimensional. In this case, we simply set $\mathbf{V}_u = \mathbf{I}$ with $N_u = N_{h,u} = C$.

The choice between these two strategies is influenced by several factors: besides the dimension N_h of the problem, the number of affine terms Q_g and Q_d and the dimension of the parameter space $\mathcal{D}$, also the software implementation and the available computational resources play an important role. However, in both cases the availability of a tight error estimate is crucial to (i) avoid oversampling in the offline phase and (ii) effectively quantify the accuracy of the reduced approximation in the online phase. We further address these aspects in the next section.

5.6 Tight error indicator by Gaussian Process regression

The sharpness of the error estimator $\Delta_N(\boldsymbol{\mu})$ is measured by the effectivity factor

$$\eta_N(\boldsymbol{\mu}) = \frac{\Delta_N(\boldsymbol{\mu})}{\|E_N(\boldsymbol{\mu})\|_{\mathcal{X}}}.$$

Proposition 5.3. *If $\tau_N(\boldsymbol{\mu}) < 1$, we have the following bounds on the effectivity factor*

$$1 \leq \eta_N(\boldsymbol{\mu}) \leq 4\frac{\gamma_h^N(\boldsymbol{\mu})}{\beta_h^N(\boldsymbol{\mu})}, \tag{5.53}$$

where $\gamma_h^N(\boldsymbol{\mu})$ denotes the continuity constant of the Fréchet derivative dG at $U_N(\boldsymbol{\mu})$.

Proof. The lower bound directly follows from Proposition 5.1. We now prove the upper bound (omitting the $\boldsymbol{\mu}$-dependence for clarity). Using equations (5.10), (5.26) and the mean value theorem, we have for all $V \in \mathcal{X}_h$

$$G(U_N; V) = -dG[U_N](U_h - U_N, V) + \int_0^1 [dG[U_N](U_h - U_N, V) - dG[U_N + s(U_h - U_N)](U_h - U_N, V)]\, ds.$$

Then, since $U_h \in B(U_N, 2\varepsilon_N/\beta_h^N)$ and $\tau_N < 1$, by (5.43) and (5.44) we obtain

$$\varepsilon_N \leq \gamma_h^N \|E_N\|_{\mathcal{X}} + \frac{K_h^N}{2}\|E_N\|_{\mathcal{X}}^2 \leq \gamma_h^N \|E_N\|_{\mathcal{X}} + \frac{1}{2}\varepsilon_N,$$

whence $\varepsilon_N \leq 2\gamma_h^N \|E_N\|_{\mathcal{X}}$. Thus,

$$\Delta_N = \frac{2}{2 - \tau_N}\,\frac{\varepsilon_N}{\beta_h^N} \leq 4\frac{\gamma_h^N}{\beta_h^N}\|E_N\|_{\mathcal{X}},$$

which proves the second inequality in (5.53). □

The upper bound of the estimate (5.53) is closely related to the condition number $\kappa_h(\boldsymbol{\mu}) = \gamma_h(\boldsymbol{\mu})/\beta_h(\boldsymbol{\mu})$ of the Hessian dG. Since PDE-constrained optimization problems are known to be often severely ill-conditioned (see, e.g., [ABG+06] and references therein), we cannot expect the error estimator $\Delta_N(\boldsymbol{\mu})$ to be sufficiently tight. However, as already mentioned, over-conservative error estimates lead to the construction of unnecessary large reduced spaces, thus affecting both the offline costs and the online efficiency.

To overcome this shortcoming, we resort to the ROM Error Surrogates (ROMES) method proposed in [DC15] (see also [PDTA14, MPL14]). The key assumption, that will be numerically verified in Sects. 6.4 and 6.5, is that both $\Delta_N(\boldsymbol{\mu})$ and $\Delta_N^L(\boldsymbol{\mu})$ strongly correlate with $\|E_N(\boldsymbol{\mu})\|_{\mathcal{X}}$. Exploiting this property, the ROMES method allows to generate a tight estimate $\widehat{\Delta}_N(\boldsymbol{\mu})$ of the error by approximating the one-dimensional map $\Delta_N^L(\boldsymbol{\mu}) \mapsto \|E_N(\boldsymbol{\mu})\|_{\mathcal{X}}$ using Gaussian Process (GP) regression [RW06]. We briefly summarize here how to construct such an estimator, referring to [DC15] for further details.

For the time being, we assume to have a set $\mathcal{T}_N$ of $M_r \geq 2$ training points

$$\mathcal{T}_N = \left\{ \left(\Delta_n^L(\boldsymbol{\mu}^{i_n}), \|E_n(\boldsymbol{\mu}^{i_n})\|_{\mathcal{X}} \right) : \{(n, \boldsymbol{\mu}^{i_n})\} \subseteq \{1, \ldots, N\} \times \mathcal{D} \right\} \subset \mathbb{R}^2, \tag{5.54}$$

obtained by evaluating the error and its estimate for different parameter values and dimensions of the ROM; we will discuss later how to generate this training set. Based on $\mathcal{T}_N$, we construct by means of Gaussian Process (GP) regression [RW06] a statistical model of the unknown deterministic error $\|E_N(\boldsymbol{\mu})\|_{\mathcal{X}}$ as a function of $\Delta_N^L(\boldsymbol{\mu})$. To this end, let us denote by $d = \log \Delta_N^L$ the independent variable and by $f = \log \|E_N\|_{\mathcal{X}}$ the dependent one. Moreover we define a vector of training points $\mathbf{d} \in \mathbb{R}^{M_r}$ and a vector $\mathbf{f} \in \mathbb{R}^{M_r}$ of training targets as

$$\mathbf{d}_i = \log(r_i), \qquad \mathbf{f}_i = \log(s_i), \qquad (r_i, s_i) \in \mathcal{T}_N,\ i = 1, \ldots, M_r.$$

Using these training data, for any given test point $d_* \in \mathbb{R}$, the GP method generates a predictive Gaussian distribution

$$f_*|\mathbf{d}, \mathbf{f}, d_* \sim \mathcal{N}(\nu(d_*), \bar{\sigma}(d_*)) \tag{5.55}$$

with mean $\nu(d_*)$ and covariance $\bar{\sigma}(d_*)$. These latter are given by

$$\nu(d_*) = \mathbf{k}_*^T(\mathbf{K} + \sigma_{M_r}^2 \mathrm{I})^{-1}\mathbf{f}, \qquad \bar{\sigma}(d_*) = k_* - \mathbf{k}_*^T(\mathbf{K} + \sigma_{M_r}^2 \mathbf{I})^{-1}\mathbf{k}_*,$$

where the matrix $\mathbf{K} \in \mathbb{R}^{M_r \times M_r}$, the vector $\mathbf{k}_* \in \mathbb{R}^{M_r}$ and $k_* \in \mathbb{R}$ are defined as

$$\mathbf{K}_{ij} = k(\mathbf{d}_i, \mathbf{d}_j), \qquad (\mathbf{k}_*)_i = k(\mathbf{d}_i, d_*), \qquad k_* = k(d_*, d_*),$$

with

$$k(x, y) = \sigma^2 \exp\left(-\frac{|x-y|^2}{2l^2}\right).$$

The free parameters $(l, \sigma, \sigma_{M_r})$ are determined as the maximizers of the log-likelihood function (see e.g. [RW06])

$$g(l, \sigma, \sigma_{M_r}) = \tfrac{1}{2}\mathbf{f}^T(\mathbf{K}(l,\sigma) + \sigma_{M_r}^2\mathbf{I})^{-1}\mathbf{f} - \tfrac{1}{2}\log|\mathbf{K}(l,\sigma) + \sigma_{M_r}^2\mathbf{I}| - \tfrac{M_r}{2}\log(2\pi).$$

We can now define a *probabilistic upper bound* $f_*^+ : \mathbb{R} \to \mathbb{R}$ of the unknown error as the $100(1-\alpha)\%$ upper predictive interval of the GP regression model

$$f_*^+(d_*) = \nu(d_*) + \sqrt{2}\,\mathrm{erf}^{-1}(1-\alpha)\;\bar{\sigma}(d_*),$$

where erf^{-1} denotes the inverse Gauss error function. This latter implicitly defines the error indicator $\widehat{\Delta}_N : \mathcal{D} \to \mathbb{R}^+$,

$$\widehat{\Delta}_N(\boldsymbol{\mu}) = \exp\left(\nu(\log\Delta_N^L(\boldsymbol{\mu})) + \sqrt{2}\,\mathrm{erf}^{-1}(1-\alpha)\;\bar{\sigma}(\log\Delta_N^L(\boldsymbol{\mu}))\right). \tag{5.56}$$

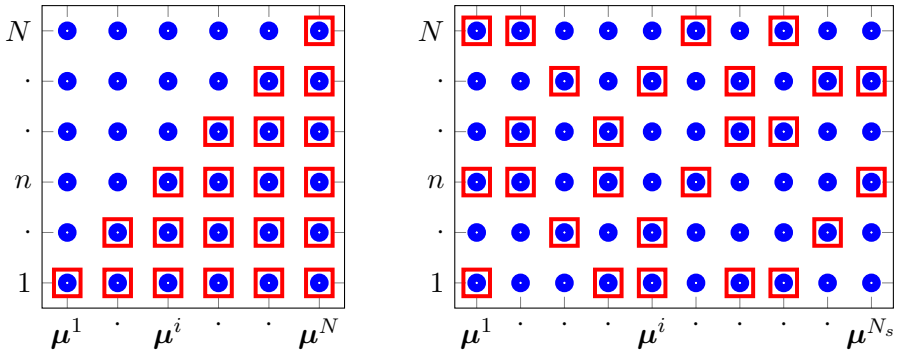


Fig. 5.2 Scheme for the construction of the regression data set $\mathcal{T}_N$ when using the greedy algorithm (left) and POD (right). The blue circles represent the points where the exact error $\|E_n(\boldsymbol{\mu}^i)\|_{\mathcal{X}}$ is available, while red squares represent those used to build $\mathcal{T}_N$.

The training set $\mathcal{T}_N$ is constructed in a different way depending on the strategy used to build the ROM. In the case the greedy algorithm is used, at the N-th generic iteration

Input: Maximum number of iterations $N_{\max}$, stopping tolerance $\varepsilon_g > 0$, training sample $\Xi_{\text{train}} \subset \mathcal{D}$, starting point $\boldsymbol{\mu}^1 \in \mathcal{D}$

$N = 0$, $\delta_0 = \varepsilon_g + 1$, $\mathbf{V} = \emptyset$, $\mathcal{T}_0 = \emptyset$
while $N < N_{\max}$ and $\delta_N > \varepsilon_g$
  $N \leftarrow N + 1$
  compute $\mathbf{U}_h(\boldsymbol{\mu}^N)$ by solving the FOM (5.17)
  update reduced basis $\mathbf{V} \leftarrow \mathbf{V} \cup \mathbf{U}_h(\boldsymbol{\mu}^N)$
  set $\mathcal{T}_N = \mathcal{T}_{N-1}$
  for $n = 1 : N$
    $\mathcal{T}_N \leftarrow \mathcal{T}_N \cup \left(\Delta_n^L(\boldsymbol{\mu}^N), \|E_n(\boldsymbol{\mu}^N)\|_{\mathcal{X}}\right)$
  end for
  if $N \geq 2$ **then**
    build surrogate error model and associated upper bound $\widehat{\Delta}_N(\boldsymbol{\mu})$
  else
    $\widehat{\Delta}_N(\boldsymbol{\mu}) = \Delta_N(\boldsymbol{\mu})$
  end if
  $[\delta_N, \boldsymbol{\mu}^{N+1}] = \max_{\boldsymbol{\mu} \in \Xi_{\text{train}}} \widehat{\Delta}_N(\boldsymbol{\mu})$
end while

Algorithm 5.1 Greedy algorithm guided by the error indicator $\widehat{\Delta}_N(\boldsymbol{\mu})$.

we have N^2 candidate training points $(\Delta_n^L(\boldsymbol{\mu}^n), \|E_n(\boldsymbol{\mu}^n)\|)$, $1 \leq n \leq N$, where both the estimate and the exact error are available.[4] However, thanks to assumption (A1), $\|E_n(\boldsymbol{\mu}^i)\|_{\mathcal{X}} \approx 0$ for $n \geq i$. On one hand, the points where the error vanishes are not as important as the ones corresponding to unexplored regions of the parameter space; on the other hand, discarding all these points from the training set would yield a very poor estimator in these (untrained) regions. We thus propose to discard the points corresponding to the first $n-1$ training inputs of the ROM of dimension n, so that $\mathcal{T}_N$ is only made of the following $M_r = N(N+1)/2$ points (see also Fig. 5.2)

$$\mathcal{T}_N = \left\{ \left(\Delta_n^L(\boldsymbol{\mu}^i),\, \|E_n(\boldsymbol{\mu}^i)\|_{\mathcal{X}}\right), \quad n = 1, \dots, i, \quad i = 1, \dots, N \right\}. \tag{5.57}$$

The GP regression model is then embedded in the usual greedy procedure [PRV+02, RHP08] as described in Algorithm 5.1.

If a POD approach is used, the regression model is built only once and for all after the snapshots collection and the construction of the ROM. In this case, there are $N_s N$ candidate training points where computing the exact error and its estimate is relatively cheap. In fact, already for a moderate number of snapshots and basis functions, say $N_s \approx 100$ and $N \approx 20$, generating the entire training set $\mathcal{T}_N$ and the corresponding regression model can be quite time-consuming. Therefore, we define $\mathcal{T}_N$ as a random

[4] Given a ROM of dimension N, all the ROMs of dimension $n = 1, \dots, N-1$ are readily available as long as the reduced spaces are hierarchical. This is the case when using both greedy and POD procedures.

subset of the available $N_s N$ points (see also Fig. 5.2)

$$\mathcal{T}_N \subseteq \left\{ \left(\Delta_n^L(\boldsymbol{\mu}^i),\, \|E_n(\boldsymbol{\mu}^i)\|_{\mathcal{X}} \right), \quad n = 1, \dots, N, \quad i = 1, \dots, N_s \right\}. \tag{5.58}$$

5.7 Optimal control of linear elliptic PDEs

In this section, we specify the general framework presented so far to the case of optimal control problems governed by linear, coercive, elliptic PDEs, such as advection-diffusion-reaction and linear elasticity equations.

5.7.1 Problem statement

With reference to (5.2), we define a linear state equation of the form

$$\langle \mathcal{E}(x; \boldsymbol{\mu}), \hat{p} \rangle = a(y, \hat{p}; \boldsymbol{\mu}) - c(u, \hat{p}; \boldsymbol{\mu}) \qquad \forall \hat{p} \in Q, \tag{5.59}$$

where the bilinear form $a(\cdot, \cdot; \boldsymbol{\mu}) : Y \times Q \to \mathbb{R}$ represents a linear elliptic operator and the bilinear form $c(\cdot, \cdot; \boldsymbol{\mu}) : \mathcal{U} \times Q \to \mathbb{R}$ expresses the action of the control. The quadratic cost functional to be minimized is given by

$$\mathcal{J}(y, u; \boldsymbol{\mu}) = \frac{1}{2} m(y - y_d(\boldsymbol{\mu}), y - y_d(\boldsymbol{\mu}); \boldsymbol{\mu}) + \frac{\sigma}{2} n(u, u; \boldsymbol{\mu}), \tag{5.60}$$

where $\sigma > 0$ is a given constant (which can be viewed as the cost needed to implement the control), $y_d(\boldsymbol{\mu}) \in Z$ is a given parameter-dependent observation function, $Z \supset Y$ is a (Hilbert) observation space, the bilinear form $m(\cdot, \cdot; \boldsymbol{\mu})$ defines the objective of the minimization, while the bilinear form $n(\cdot, \cdot; \boldsymbol{\mu})$ acts as a penalization term for the control variable.

Since we consider second-order coercive elliptic equation as constraint, we can assume without loss of generality that $Q \equiv Y$. Then, we assume that the bilinear form $a(\cdot, \cdot; \boldsymbol{\mu})$ is bounded over $Y \times Q$ and coercive over $Y \equiv Q$ for any $\boldsymbol{\mu} \in \mathcal{D}$. We assume that the bilinear form $c(\cdot, \cdot; \boldsymbol{\mu})$ is symmetric and bounded, and the bilinear form $n(\cdot, \cdot; \boldsymbol{\mu}) : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ is symmetric, bounded and coercive. Moreover, we assume the bilinear form $m(\cdot, \cdot; \boldsymbol{\mu})$ to be symmetric, continuous and non-negative in the norm induced by the space Z. Holding these assumptions, conditions (H1)-(H4) (see Sect. 5.1.1) are easily verified, see e.g. [HPUU09, Trö10]. In particular, by formulating the optimality system as a saddle-point problem, condition (5.13) can be verified by means of Brezzi theorem [Bre74, BBF13], see e.g. [GB04, GB09].

An example of PDE constraint which can be formulated as in (5.59) is provided by the following advection-diffusion-reaction equation

$$\begin{cases} -\nabla \cdot (k(\boldsymbol{\mu}) \nabla y) + \boldsymbol{b}(\boldsymbol{\mu}) \cdot \nabla y + c_0(\boldsymbol{\mu}) y = \rho_1 u_1 & \text{in } \Omega(\boldsymbol{\mu}) \\ y = \rho_2 u_2 & \text{on } \Gamma_D(\boldsymbol{\mu}) \\ k(\boldsymbol{\mu}) \nabla y \cdot \boldsymbol{n} = \rho_3 u_3 & \text{on } \Gamma_N(\boldsymbol{\mu}). \end{cases} \tag{5.61}$$

Here, $\Omega(\boldsymbol{\mu}) \subset \mathbb{R}^d$ $(d = 2, 3)$ is a parametrized domain with boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, being Γ_D the Dirichlet portion and Γ_N the Neumann portion. Moreover, $k(\boldsymbol{\mu}) : \mathbb{R}^d \to \mathbb{R}$

is the diffusion coefficient, $\boldsymbol{b}(\boldsymbol{\mu}) : \mathbb{R}^d \to \mathbb{R}^d$ is the convective field, while $c_0(\boldsymbol{\mu})$ is a reaction coefficient. The Boolean variables ρ_i satisfy $\rho_1+\rho_2+\rho_3 = 1$, so that the control variable u may represent either a forcing term (u_1), a Dirichlet condition on Γ_D (u_2) or a Neumann flux on Γ_N (u_3).

A second example is given by the following linear (isotropic) elasticity problem

$$\begin{cases} -\nabla \cdot (\boldsymbol{\sigma}(\boldsymbol{y})) = \rho_1 \boldsymbol{u}_1 & \text{in } \Omega(\boldsymbol{\mu}) \\ \boldsymbol{y} = \rho_2 \boldsymbol{u}_2 & \text{on } \Gamma_D(\boldsymbol{\mu}) \\ \boldsymbol{\sigma}(\boldsymbol{y})\boldsymbol{n} = \rho_2 \boldsymbol{u}_3 & \text{on } \Gamma_N(\boldsymbol{\mu}), \end{cases} \tag{5.62}$$

where $\boldsymbol{y} : \mathbb{R}^d \to \mathbb{R}^d$ denotes the displacement field,

$$\boldsymbol{\varepsilon}(\boldsymbol{y}) = \frac{1}{2}(\nabla \boldsymbol{y} + \nabla \boldsymbol{y}^T), \qquad \boldsymbol{\sigma}(\boldsymbol{y}) = 2\mu\boldsymbol{\varepsilon}(\boldsymbol{y}) + \lambda(\nabla \cdot \boldsymbol{y})\boldsymbol{I},$$

are the strain and stress tensors, respectively, while μ and λ are the Lamé coefficients, which can be expressed in terms of the Young modulus E and the Poisson coefficient ν as

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)}.$$

The latter can be treated as parameters. As before, the Boolean variables ρ_i satisfy $\rho_1 + \rho_2 + \rho_3 = 1$, so that the control variable $\boldsymbol{u}$ may represent either a distributed load ($\boldsymbol{u}_1$), an imposed displacement on Γ_D ($\boldsymbol{u}_2$), or a traction on Γ_N ($\boldsymbol{u}_3$).

5.7.2 Aggregated reduced spaces

In order to build reduced spaces satisfying assumptions (A1)-(A3), we employ the aggregated strategy proposed in [Neg11], see also [HV08, Ded10].

We first describe the construction in combination with the greedy algorithm. We denote by $S_N = \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N\}$ the parameter samples selected by the greedy algorithm and consider the corresponding full-order solutions $(y_h(\boldsymbol{\mu}^n), u_h(\boldsymbol{\mu}^n), p_h(\boldsymbol{\mu}^n))$, $1 \le n \le N$. We define the aggregated state and adjoint spaces Y_N, Q_N as

$$Y_N = Q_N = \text{span}\Big\{ y_h(\boldsymbol{\mu}^n),\, p_h(\boldsymbol{\mu}^n) \Big\}_{n=1}^{N}.$$

Then, we define the reduced space $\mathcal{U}_N = \text{span}\{u_h(\boldsymbol{\mu}^n)\}_{n=1}^N$ for the control variable. Finally, a suitable orthonormalization is performed separately on both spaces.

If a POD approach is used instead, we first compute full-order solutions $(y_h(\boldsymbol{\mu}^i)$, $u_h(\boldsymbol{\mu}^i), p_h(\boldsymbol{\mu}^i))$, $i = 1, \cdots, N_s$ of the optimality system for N_s parameters value $\boldsymbol{\mu}_i$. Then, we perform POD separately on each variable in order to obtain the reduced state space Y_N^*, the reduced adjoint spaces Q_N^*, and the reduced control space $\mathcal{U}_N$. We define the aggregated space as $Y_N = Q_N = Y_N^s \cup Q_N^a$.

In both cases, we finally set $X_N = Y_N \times \mathcal{U}_N$, $\mathcal{X}_N = X_N \times Q_N$, so that the reduced space $\mathcal{X}_N$ has dimension $5N$. Thanks to the use of aggregated spaces, the resulting ROM can be proved to be well-posed by applying Brezzi theorem on the optimality system, see [Neg11].

5.8 Optimal control of the Stokes equations

In this section we introduce a general formulation for parametrized optimization problems governed by the Stokes equations and analyze their RB approximation. Specifically, we consider the following parametrized optimal control problem: find a triple $(\boldsymbol{v}, \pi, \boldsymbol{u})$ such that the cost functional

$$\mathcal{J}(\boldsymbol{v}, \pi, \boldsymbol{u}; \boldsymbol{\mu}) = \mathcal{F}_1(\boldsymbol{v}, \pi; \boldsymbol{\mu}) + \mathcal{F}_2(\boldsymbol{u}; \boldsymbol{\mu}) \tag{5.63}$$

is minimized, subject to the steady Stokes equations:

$$\begin{cases} -\nu\Delta\boldsymbol{v} + \nabla\pi = \rho_1\boldsymbol{u}_1 & \text{in } \Omega(\boldsymbol{\mu}) \\ \operatorname{div}\boldsymbol{v} = 0 & \text{in } \Omega(\boldsymbol{\mu}) \\ \boldsymbol{v} = \rho_2\boldsymbol{u}_2 & \text{on } \Gamma_D(\boldsymbol{\mu}) \\ -\pi\boldsymbol{n} + \nu(\nabla\boldsymbol{v})\mathbf{n} = \rho_3\boldsymbol{u}_3 & \text{on } \Gamma_N(\boldsymbol{\mu}). \end{cases} \tag{5.64}$$

Here, $\Omega(\boldsymbol{\mu}) \subset \mathbb{R}^d$ $(d = 2, 3)$ is a spatial domain with Lipschitz boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$, where Γ_D and Γ_N are the (disjoint) Dirichlet and the Neumann portions of the boundary; the state variables $(\boldsymbol{v},\pi)$ denote the velocity and pressure fields, respectively, while $\nu > 0$ is the kinematic viscosity. The Boolean variables ρ_i satisfy $\rho_1 + \rho_2 + \rho_3 = 1$, so that the control variable $\boldsymbol{u}$ may represent either a source term $(\boldsymbol{u}_1)$, a boundary velocity on Γ_D $(\boldsymbol{u}_2)$ or a Neumann flux on Γ_N $(\boldsymbol{u}_3)$. Possible choices for $\mathcal{F}_1$ are (see, e.g., [GHS93, Hei98]) the viscous energy dissipation

$$\mathcal{F}_1(\boldsymbol{v}; \boldsymbol{\mu}) = \frac{\nu}{2} \int_{\Omega(\boldsymbol{\mu})} |\nabla\boldsymbol{v}|^2 \, d\Omega,$$

vorticity

$$\mathcal{F}_1(\boldsymbol{v}; \boldsymbol{\mu}) = \frac{1}{2} \int_{\Omega(\boldsymbol{\mu})} |\nabla \times \boldsymbol{v}|^2 \, d\Omega,$$

or velocity tracking type functionals

$$\mathcal{F}_1(\boldsymbol{v}; \boldsymbol{\mu}) = \frac{1}{2} \int_{\Omega(\boldsymbol{\mu})} |\boldsymbol{v} - \boldsymbol{v}_d(\boldsymbol{\mu})|^2 \, d\Omega,$$

where $\boldsymbol{v}_d(\boldsymbol{\mu})$ is a desired velocity field.

5.8.1 Weak formulation

In order to cast the optimal control problem (5.63)-(5.64) in the general formulation (5.1)-(5.2), we first define the velocity space V

$$V = \mathbf{H}^1_D(\Omega) = \left\{ \boldsymbol{v} \in [H^1(\Omega)]^d \colon \boldsymbol{v} = 0 \text{ on } \Gamma_D \right\},$$

the pressure space $M = L^2(\Omega)$, and the state space $Y = V \times M$. Moreover, we assume the control space $\mathcal{U}$ to be a Hilbert space. For example, if we consider a distributed forcing term $\boldsymbol{u}_1$ as control variable, the natural choice is $\mathcal{U} = \mathbf{L}^2(\Omega_o)$, whereas if we consider as control variable $\boldsymbol{u}_3$, $\mathcal{U} = \mathbf{L}^2(\Gamma_N^o)$. In Sect. 5.9.1, a deeper discussion will be

devoted to the case of a Dirichlet control $\boldsymbol{u}_2$. Then, we set $X = Y \times \mathcal{U}$ (endowed with the usual ℓ^2-norm) and $Q = Y$, since the Stokes operator (5.64) can be considered with values in $Q' = Y'$.

To take into account a possible Dirichlet control $\boldsymbol{u}_2$, we split the velocity field as $\boldsymbol{v} = \boldsymbol{v}_0 + \rho_2 \mathbf{R}(\boldsymbol{u}_2)$, where $\boldsymbol{v} \in \mathbf{H}^1_D(\Omega)$, $\mathbf{R}\colon \mathbf{H}^{1/2}(\Gamma_D) \to \mathbf{H}^1(\Omega)$ is a bounded extension operator such that $\mathbf{R}(\boldsymbol{u}_2) \in \mathbf{H}^1(\Omega)$ and $\mathbf{R}(\boldsymbol{u}_2)|_{\Gamma_D} = \boldsymbol{u}_2$. For the sake of simplicity, we still denote in the following $\boldsymbol{v}_0$ with $\boldsymbol{v}$, as no ambiguity occurs, while we use the notation below for test and trial functions, respectively:

$$\begin{aligned} &\text{trial:} && x = (y, u) \in X, && y = (\boldsymbol{v}, \pi) \in Y, && p = (\boldsymbol{\lambda}, \eta) \in Q, \\ &\text{test:} && \hat{x} = (\hat{y}, \hat{u}) \in X, && \hat{y} = (\hat{\boldsymbol{v}}, \hat{\pi}) \in Y, && \hat{p} = (\hat{\boldsymbol{\lambda}}, \hat{\eta}) \in Q. \end{aligned}$$

Thus, the optimization variable $x = (y, u)$ denotes the aggregated state and control variables, the former being given by velocity $\boldsymbol{v}$ and pressure π.

We define the bilinear form $S(\cdot,\cdot)\colon Y \times Q \to \mathbb{R}$ associated to the Stokes operator [QV94, ESW04]

$$S(y, \hat{p}; \boldsymbol{\mu}) = a(\boldsymbol{v}, \hat{\boldsymbol{\lambda}}; \boldsymbol{\mu}) + b(\hat{\boldsymbol{\lambda}}, \pi; \boldsymbol{\mu}) + b(\boldsymbol{v}, \hat{\eta}; \boldsymbol{\mu}), \tag{5.65}$$

where $a(\cdot,\cdot;\boldsymbol{\mu})\colon V \times V \to \mathbb{R}$ and $b(\cdot,\cdot;\boldsymbol{\mu})\colon V \times M \to \mathbb{R}$ are defined by

$$a(\boldsymbol{v}, \hat{\boldsymbol{\lambda}}; \boldsymbol{\mu}) = \int_{\Omega(\boldsymbol{\mu})} \nu \nabla \boldsymbol{v} \cdot \nabla \hat{\boldsymbol{\lambda}} \, d\Omega, \qquad b(\boldsymbol{v}, \hat{\eta}; \boldsymbol{\mu}) = -\int_{\Omega(\boldsymbol{\mu})} \hat{\eta} \nabla \cdot \boldsymbol{v} \, d\Omega.$$

The bilinear form $C(\cdot,\cdot;\boldsymbol{\mu})\colon U \times Q \to \mathbb{R}$ associated to the control variable is given by (after integration by parts and introducing the lifting operator, see e.g. [Gun03])

$$\begin{aligned} C(u, \hat{p}; \boldsymbol{\mu}) = \rho_1 \int_{\Omega(\boldsymbol{\mu})} \boldsymbol{u}_1 \cdot \hat{\boldsymbol{\lambda}} \, d\Omega + \rho_3 \int_{\Gamma_N(\boldsymbol{\mu})} \boldsymbol{u}_3 \cdot \hat{\boldsymbol{\lambda}} \, d\Gamma \\ - \rho_2 a(\mathbf{R}(\boldsymbol{u}_2), \hat{\boldsymbol{\lambda}}) - \rho_2 b(\mathbf{R}(\boldsymbol{u}_2), \hat{\eta}). \end{aligned} \tag{5.66}$$

Then, the state operator $\mathcal{E}(\cdot;\boldsymbol{\mu})\colon X \to Q'$ is given by

$$_{Q'}\langle \mathcal{E}(x;\boldsymbol{\mu}), \hat{p} \rangle_Q = S(y, \hat{p}; \boldsymbol{\mu}) - C(u, \hat{p}; \boldsymbol{\mu}) \qquad \forall\, \hat{p} \in Q. \tag{5.67}$$

Finally, let us express the quadratic functionals $\mathcal{F}_1$ and $\mathcal{F}_2$ appearing in (5.63) as

$$\mathcal{F}_1(\boldsymbol{v};\boldsymbol{\mu}) = \frac{1}{2} m(\boldsymbol{v} - \boldsymbol{v}_d(\boldsymbol{\mu}), \boldsymbol{v} - \boldsymbol{v}_d(\boldsymbol{\mu})) \qquad \mathcal{F}_2(\boldsymbol{u};\boldsymbol{\mu}) = \frac{\sigma}{2} n(\boldsymbol{u}, \boldsymbol{u}), \tag{5.68}$$

where $\mathcal{Z} \supset V$ is a Hilbert (observation) space, $m(\cdot,\cdot;\boldsymbol{\mu})\colon Z \times Z \to \mathbb{R}$ is a symmetric, continuous, non-negative bilinear form, and $\boldsymbol{v}_d(\boldsymbol{\mu}) \in Z$ is a given observation function. Furthermore, $\sigma > 0$ is a given constant, while $n(\cdot,\cdot;\boldsymbol{\mu})\colon \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ is a symmetric, bounded and coercive bilinear form. Note that in case of Dirichlet control the functional $\mathcal{F}_1(\boldsymbol{v};\boldsymbol{\mu})$ is slightly modified. Indeed, by recalling that the velocity field is $\boldsymbol{v} = \boldsymbol{v}_0 + \mathbf{R}(\boldsymbol{u}_2)$, we can rewrite $\mathcal{F}_1$ as

$$\mathcal{F}_1(\boldsymbol{v}, \boldsymbol{u};\boldsymbol{\mu}) = \frac{1}{2} m(\boldsymbol{v}_0 - \boldsymbol{v}_d(\boldsymbol{\mu}), \boldsymbol{v}_0 - \boldsymbol{v}_d(\boldsymbol{\mu})) + \frac{1}{2} m(\mathbf{R}(\boldsymbol{u}_2), \mathbf{R}(\boldsymbol{u}_2)) + m(\boldsymbol{v}_0 - \boldsymbol{v}_d(\boldsymbol{\mu}), \mathbf{R}(\boldsymbol{u}_2)).$$

5.8.2 Optimality conditions

By using the definitions (5.67) and (5.68), we can express the optimality system as: find $(y, u, p) \in Y \times \mathcal{U} \times Q$ such that

$$\begin{array}{cc|ccc|l} m(\boldsymbol{v}, \hat{\boldsymbol{v}}; \boldsymbol{\mu}) & 0 & S(\hat{y}, p; \boldsymbol{\mu}) & = & m(\boldsymbol{v}_d(\boldsymbol{\mu}), \hat{\boldsymbol{v}}; \boldsymbol{\mu}) & \forall \hat{y} \in Y \\ 0 & \sigma n(u, \hat{u}; \boldsymbol{\mu}) & -C(\hat{u}, p; \boldsymbol{\mu}) & = & 0 & \forall \hat{u} \in U \\ \hline S(y, \hat{p}; \boldsymbol{\mu}) & -C(u, \hat{p}; \boldsymbol{\mu}) & 0 & = & 0 & \forall \hat{p} \in Q. \end{array} \tag{5.69}$$

Formulation (5.69) highlights the structure of the optimality system, featuring two nested saddle-point problems: an *outer* saddle-point problem (given by the optimization problem) and an *inner* one (given by the Stokes constraint). The stability properties of the whole system reflect this particular structure, which must be carefully taken into account when designing suitable reduced spaces, as shown in Sect. 5.8.4.

As in the case of Sect. 5.7, condition (5.13) can be easily verified by formulating the optimality system as a saddle-point problem and applying Brezzi theorem, see e.g. [GB04, GB09]. To this end, it is useful to define the following bilinear forms:

$$\mathcal{A}(x, \hat{x}; \boldsymbol{\mu}) = m(\boldsymbol{v}, \hat{\boldsymbol{v}}) + \sigma n(u, \hat{u}) \qquad \forall x, \hat{x} \in X,$$

$$\mathcal{B}(x, \hat{p}; \boldsymbol{\mu}) = S(y, \hat{p}; \boldsymbol{\mu}) - C(u, \hat{p}; \boldsymbol{\mu}) \qquad \forall x = (y, u) \in X, \hat{p} \in Q.$$

Then, the left-hand side of (5.69) can be expressed in the following compact form

$$dG((x, p), (\hat{x}, \hat{p}); \boldsymbol{\mu}) = \mathcal{A}(x, \hat{x}; \boldsymbol{\mu}) + \mathcal{B}(\hat{x}, p; \boldsymbol{\mu}) + \mathcal{B}(x, \hat{p}; \boldsymbol{\mu}). \tag{5.70}$$

Proposition 5.4. *The bilinear form $dG(\cdot; \cdot; \boldsymbol{\mu})$ is continuous and inf-sup stable over $\mathcal{X} \times \mathcal{X}$.*

Proof. To prove that $dG(\cdot; \cdot; \boldsymbol{\mu})$ is continuous and inf-sup stable it is sufficient (see e.g. [Nic82, XZ03]) to prove that $\mathcal{A}(\cdot; \cdot; \boldsymbol{\mu})$ and $\mathcal{B}(\cdot; \cdot; \boldsymbol{\mu})$ fulfill the assumptions of Brezzi theorem. For the complete proof we refer to [GB09, Ch. 11]; here we briefly recall its basic ingredients. The continuity of $\mathcal{A}(\cdot, \cdot; \boldsymbol{\mu})$ and $\mathcal{B}(\cdot, \cdot; \boldsymbol{\mu})$ follows from the continuity of the bilinear forms $S(\cdot, \cdot; \boldsymbol{\mu})$, $C(\cdot, \cdot; \boldsymbol{\mu})$, $m(\cdot, \cdot; \boldsymbol{\mu})$ and $n(\cdot, \cdot; \boldsymbol{\mu})$. To prove the coercivity of $\mathcal{A}(\cdot, \cdot; \boldsymbol{\mu})$ on $X^0 = \{\hat{x} \in X : \mathcal{B}(\hat{x}, \hat{p}; \boldsymbol{\mu}) = 0 \; \forall \hat{p} \in Q\}$ it is sufficient to exploit the weak coercivity of the Stokes operator, the coercivity of $n(\cdot, \cdot; \boldsymbol{\mu})$ and the non-negativeness of $m(\cdot, \cdot; \boldsymbol{\mu})$. In particular, for any prescribed $\boldsymbol{\mu} \in \mathcal{D}$, for all $x = (y, u) \in X^0$ the following estimate holds

$$\|y\|_Y \leq \frac{\gamma^C(\boldsymbol{\mu})}{\beta^S(\boldsymbol{\mu})} \|u\|_U, \tag{5.71}$$

where $\gamma^C(\boldsymbol{\mu})$ is the continuity constant of the bilinear form $C(\cdot, \cdot; \boldsymbol{\mu})$ and $\beta^S(\boldsymbol{\mu})$ defined as the Babuška inf-sup constant of the Stokes operator, i.e.

$$\beta^S(\boldsymbol{\mu}) = \inf_{y_1 \in Y} \sup_{y_2 \in Y} \frac{S(y_1, y_2; \boldsymbol{\mu})}{\|y_1\|_Y \|y_2\|_Y} = \inf_{y_2 \in Y} \sup_{y_1 \in Y} \frac{S(y_1, y_2; \boldsymbol{\mu})}{\|y_1\|_Y \|y_2\|_Y}. \tag{5.72}$$

Denoting with $\alpha^n(\boldsymbol{\mu})$ the coercivity constant of $n(\cdot, \cdot; \boldsymbol{\mu})$ and exploiting (5.71), we obtain that, for any $x \in X^0$

$$\mathcal{A}(x, x; \boldsymbol{\mu}) \geq \sigma n(u, u; \boldsymbol{\mu}) \geq \frac{\sigma}{2} \alpha^n(\boldsymbol{\mu}) \|u\|_U^2 + \frac{\sigma \alpha^n(\boldsymbol{\mu})}{2} \left(\frac{\beta^S(\boldsymbol{\mu})}{\gamma^C(\boldsymbol{\mu})} \right)^2 \|y\|_Y^2 \geq C(\boldsymbol{\mu}) \|x\|_X^2,$$

where

$$C(\boldsymbol{\mu}) = \frac{\sigma}{2}\alpha^n(\boldsymbol{\mu}) \min\left\{1, \left(\frac{\beta^S(\boldsymbol{\mu})}{\gamma^C(\boldsymbol{\mu})}\right)^2\right\}.$$

Finally, by exploiting again the weak coercivity of the Stokes operator and the fact that $Y \equiv Q$, we verify that $\mathcal{B}(\cdot,\cdot;\boldsymbol{\mu})$ is inf-sup stable over $X \times Q$

$$\sup_{0\neq\hat{x}\in X} \frac{\mathcal{B}(\hat{x},\hat{p};\boldsymbol{\mu})}{\|\hat{x}\|_X} = \sup_{0\neq(\hat{y},\hat{u})\in Y\times\mathcal{U}} \frac{S(\hat{y},\hat{p};\boldsymbol{\mu}) - C(\hat{u},\hat{p};\boldsymbol{\mu})}{(\|\hat{y}\|_Y^2 + \|\hat{u}\|_{\mathcal{U}}^2)^{1/2}}$$
$$\geq \sup_{0\neq\hat{y}\in Y} \frac{S(\hat{y},\hat{p};\boldsymbol{\mu})}{\|\hat{y}\|_Y} \geq \beta^S(\boldsymbol{\mu})\|\hat{p}\|_Y = \beta^S(\boldsymbol{\mu})\|\hat{p}\|_Q. \quad \square$$

5.8.3 Finite element approximation

We first introduce a stable pair of FE spaces $V_h \subset \mathbf{H}^1(\Omega)$ and $M_h \subset L^2(\Omega)$ [QV94, ESW04] such that

$$\exists\, \beta_0^b > 0: \quad \beta_h^b(\boldsymbol{\mu}) = \inf_{\pi_h\in M_h} \sup_{\boldsymbol{v}_h\in\mathbf{V}_h} \frac{b(\boldsymbol{v}_h,\pi_h;\boldsymbol{\mu})}{\|\boldsymbol{v}_h\|_V\|\pi_h\|_M} \geq \beta_0^b, \qquad \forall\boldsymbol{\mu}\in\mathcal{D}. \tag{5.73}$$

For instance, Taylor-Hood $\mathbb{P}_2$-$\mathbb{P}_1$ or $\mathbb{P}_1^b$-$\mathbb{P}_1$ elements meet this requirement. The stability assumption (5.73) on the velocity and pressure spaces implies the fulfillment of the following Babuška inf-sup condition on $S(\cdot,\cdot;\boldsymbol{\mu})$, i.e.

$$\exists\, \beta_0^S > 0: \quad \beta_h^S(\boldsymbol{\mu}) = \inf_{y_1\in Y_h} \sup_{y_2\in Y_h} \frac{S(y_1,y_2;\boldsymbol{\mu})}{\|y_1\|_Y\|y_2\|_Y} = \inf_{y_2\in Y_h} \sup_{y_1\in Y_h} \frac{S(y_1,y_2;\boldsymbol{\mu})}{\|y_1\|_Y\|y_2\|_Y} \geq \beta_0^S, \tag{5.74}$$

where $Y_h = V_h \times M_h$. Furthermore, we assume $\mathcal{U}_h$ to be a suitable FE subspace of $\mathcal{U}$ and we set $X_h = Y_h \times \mathcal{U}_h$, $Q_h = Y_h$.

Provided that $Y_h = Q_h$, it can be shown that the bilinear form $\mathcal{A}(\cdot,\cdot;\boldsymbol{\mu})$ is continuous over $X_h \times X_h$ and coercive over $X_h^0 = \{\hat{x}\in X_h \colon \mathcal{B}(\hat{x},\hat{p};\boldsymbol{\mu}) = 0\ \forall\,\hat{p}\in Q_h\}$, with continuity constant $\gamma_h^a(\boldsymbol{\mu})$ and coercivity constant $\alpha_h^a(\boldsymbol{\mu})$, respectively. In the same way, $\mathcal{B}(\cdot,\cdot;\boldsymbol{\mu})$ is continuous and inf-sup stable over $X_h \times Q_h$, i.e.

$$\exists\, \hat{\beta}_0 > 0: \quad \hat{\beta}_h(\boldsymbol{\mu}) = \inf_{\hat{p}\in Q_h} \sup_{\hat{x}\in X_h} \frac{\mathcal{B}(\hat{x},\hat{p};\boldsymbol{\mu})}{\|\hat{x}\|_X\|\hat{p}\|_Q} \geq \beta_0 \qquad \forall\boldsymbol{\mu}\in\mathcal{D}. \tag{5.75}$$

In particular, $\hat{\beta}_h(\boldsymbol{\mu}) \geq \beta_h^S(\boldsymbol{\mu})$. Therefore, thanks to Brezzi theory, also the FE approximation of the optimality system is well-posed.

5.8.4 Construction of aggregated RB spaces by the greedy algorithm

We now define a stable pair of aggregated reduced spaces X_N, Q_N for state, control and adjoint variables. To take into account the double saddle-point structure, we first have to ensure the stability of the RB approximation of the Stokes state operator. Then, we have to verify the stability of the whole optimality system, i.e. we have to guarantee the

coercivity of the bilinear form $\mathcal{A}(\cdot,\cdot;\boldsymbol{\mu})$ over $X_N^0 = \{\hat{x} \in X_N \colon \mathcal{B}(\hat{x},\hat{p};\boldsymbol{\mu}) = 0,\ \forall \hat{p} \in Q_N\}$, and the fulfillment of an equivalent RB inf-sup condition on $\mathcal{B}(\cdot,\cdot;\boldsymbol{\mu})$, i.e.

$$\exists\, \hat{\beta}_0 > 0 : \quad \hat{\beta}_N(\boldsymbol{\mu}) = \inf_{\hat{p}\in Q_N} \sup_{\hat{x}\in X_N} \frac{\mathcal{B}(\hat{x},\hat{p};\boldsymbol{\mu})}{\|\hat{x}\|_X \|\hat{p}\|_Q} \geq \beta_0, \qquad \forall \boldsymbol{\mu} \in \mathcal{D}. \tag{5.76}$$

This also implies the fulfillment of an equivalent RB Babuška inf-sup condition on the whole optimality system [Nic82, XZ03]. We employ the following strategy: we achieve stability of the Stokes operator by enriching the velocity space with suitably defined *supremizer solutions* [RV07]. Then to guarantee the stability of the optimality system, we define suitable aggregated spaces for state and adjoint variables, as shown in Sect. 5.7.2. The two recipes are combined together as described below.

Let us denote by $S_N = \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N\}$ the parameter samples selected by the greedy algorithm and consider the corresponding full-order solutions $(x_h(\boldsymbol{\mu}^n),\, p_h(\boldsymbol{\mu}^n))$, $1 \leq n \leq N$. We introduce the *supremizer functions* $\mathbf{T}_\pi^{\boldsymbol{\mu}} \in V_h$ defined as [Rov03, RV07]

$$(\mathbf{T}_\pi^{\boldsymbol{\mu}}, \hat{\boldsymbol{v}})_V = b(\hat{\boldsymbol{v}}, \pi; \boldsymbol{\mu}) \qquad \forall \hat{\boldsymbol{v}} \in V_h, \tag{5.77}$$

i.e. as the Riesz representers of the linear functional $b(\cdot,\pi;\boldsymbol{\mu})$ restricted on V_h. In this way, we can define an aggregated (state and adjoint) pressure space M_N

$$M_N = \operatorname{span}\{\pi_h(\boldsymbol{\mu}^n),\, \eta_h(\boldsymbol{\mu}^n), \quad n = 1, \dots, N\}, \tag{5.78}$$

and an aggregated (state and adjoint) velocity space V_N, including also the corresponding supremizer snapshots

$$V_N^{\boldsymbol{\mu}} = \operatorname{span}\Big\{\boldsymbol{v}_h(\boldsymbol{\mu}^n), \mathbf{T}^{\boldsymbol{\mu}}_{\pi_h(\boldsymbol{\mu}^n)}, \boldsymbol{\lambda}_h(\boldsymbol{\mu}^n), \mathbf{T}^{\boldsymbol{\mu}}_{\eta_h(\boldsymbol{\mu}^n)}, \quad n = 1, \dots, N\Big\}. \tag{5.79}$$

Then, we define the reduced space for the control variable

$$\mathcal{U}_N = \operatorname{span}\{u_h(\boldsymbol{\mu}^n), \quad n = 1, \dots, N\}. \tag{5.80}$$

Finally, let us define

$$Y_N = V_N^{\boldsymbol{\mu}} \times M_N, \qquad X_N = Y_N \times \mathcal{U}_N, \qquad Q_N = Y_N, \tag{5.81}$$

as the reduced spaces for the state, state and control, and adjoint variables, respectively.

Remark 5.7. The need to enrich the reduced velocity space with supremizer solutions stems from the use of stable finite element velocity-pressure spaces, see Sect. 5.8.3. An alternative option relying on a stabilized formulation shall be discussed in Sect. 5.9 dealing with the optimal control of Navier-Stokes equations. •

5.8.5 Stability properties

In order to analyze the stability of the proposed scheme, let us introduce the following RB inf-sup constant on the Stokes operator

$$\beta_N^S(\boldsymbol{\mu}) = \inf_{\hat{y}\in Y_N} \sup_{\hat{p}\in Q_N} \frac{S(\hat{y},\hat{p};\boldsymbol{\mu})}{\|\hat{y}\|_Y \|\hat{p}\|_Q}, \tag{5.82}$$

and the following RB inf-sup constant on the optimality system

$$\beta_N(\boldsymbol{\mu}) = \inf_{(x,p)\in\mathcal{X}_N} \sup_{(\hat{x},\hat{p})\in\mathcal{X}_N} \frac{dG((x,p),(\hat{x},\hat{p});\boldsymbol{\mu})}{\|(x,p)\|_{\mathcal{X}}\|(\hat{x},\hat{p})\|_{\mathcal{X}}}. \tag{5.83}$$

The well-posedness of the RB approximation is ensured by the following proposition.

Proposition 5.5. *If X_N and Q_N are chosen accordingly to (5.78)-(5.81), then $\mathcal{A}(\cdot,\cdot;\boldsymbol{\mu})$ and $\mathcal{B}(\cdot,\cdot;\boldsymbol{\mu})$ satisfy the assumptions of Brezzi theorem over X_N, Q_N. Moreover, the following inequalities for the RB stability factors hold, for any $\boldsymbol{\mu}\in\mathcal{D}$:*

$$\hat{\beta}_N(\boldsymbol{\mu}) \geq \beta_N^S(\boldsymbol{\mu}) \geq \frac{\alpha_h^a(\boldsymbol{\mu})}{1+\left(\gamma_h^a(\boldsymbol{\mu})/\beta_h^b(\boldsymbol{\mu})\right)^2}, \tag{5.84}$$

$$\alpha_N(\boldsymbol{\mu}) \geq \frac{\sigma}{2}\alpha_h^n(\boldsymbol{\mu}) \min\left\{1, \left(\frac{\beta_N^S(\boldsymbol{\mu})}{\gamma_h^C(\boldsymbol{\mu})}\right)^2\right\}, \tag{5.85}$$

$$\beta_N(\boldsymbol{\mu}) \geq \frac{\alpha_N(\boldsymbol{\mu})}{1+\left(\gamma_h^{\mathcal{A}}(\boldsymbol{\mu})/\hat{\beta}_N(\boldsymbol{\mu})\right)^2}, \tag{5.86}$$

where $\gamma_h^{\mathcal{A}}$ is the continuity constant of $\mathcal{A}(\cdot,\cdot;\boldsymbol{\mu})$ over $X_h\times X_h$, and $\alpha_N(\boldsymbol{\mu})$ is the coercivity constant of $\mathcal{A}(\cdot,\cdot;\boldsymbol{\mu})$ over $X_N^0\times X_N^0$.

Proof. The continuity of $\mathcal{A}(\cdot,\cdot:\boldsymbol{\mu})$ and $\mathcal{B}(\cdot,\cdot:\boldsymbol{\mu})$ are automatically fulfilled, since $X_N\subset X_h$ and $Q_N\subset Q_h$. Then, thanks to the enrichment of the RB velocity space $V_N^{\boldsymbol{\mu}}$ by supremizer solutions and to the fact that $Y_N = Q_N$, the RB approximation of the Stokes operator satisfies the assumptions of Brezzi theorem [RV07], which implies [Nic82, XZ03] the existence of a constant $\beta_{N0}^S > 0$ such that

$$\beta_N^S(\boldsymbol{\mu}) = \inf_{\hat{y}\in Y_N}\sup_{\hat{p}\in Q_N}\frac{S(\hat{y},\hat{p};\boldsymbol{\mu})}{\|\hat{y}\|_Y\|\hat{p}\|_Q} = \inf_{\hat{p}\in Q_N}\sup_{\hat{y}\in Y_N}\frac{S(\hat{y},\hat{p};\boldsymbol{\mu})}{\|\hat{y}\|_Y\|\hat{p}\|_Q} \geq \beta_{N0}^S;$$

in other words $S(\cdot,\cdot;\boldsymbol{\mu})$ is inf-sup stable on $Y_N\times Q_N$. By exploiting this property, the coercivity of $\mathcal{A}(\cdot,\cdot;\boldsymbol{\mu})$ over X_N^0, and the fulfillment of the inf-sup condition (5.76) can be easily proved.
To obtain the inequality (5.84), we first apply to $\beta_N^S(\boldsymbol{\mu})$ the estimate

$$\beta_N^S(\boldsymbol{\mu}) \geq \frac{\alpha_N^a(\boldsymbol{\mu})}{1+\left(\gamma_N^a(\boldsymbol{\mu})/\beta_N^b(\boldsymbol{\mu})\right)^2} \qquad \forall\boldsymbol{\mu}\in\mathcal{D},$$

proved in [KSZ13, Th. 1], where $\gamma_N^a(\boldsymbol{\mu})$ and $\alpha_N^a(\boldsymbol{\mu})$ are the RB approximation of the continuity and coercivity constants of $a(\cdot,\cdot;\boldsymbol{\mu})$. Then, since $\gamma_N^a(\boldsymbol{\mu})\leq\gamma_h^a(\boldsymbol{\mu})$, $\alpha_N^a(\boldsymbol{\mu})\geq\alpha_h^a(\boldsymbol{\mu})$ and $\beta_N^b(\boldsymbol{\mu})\geq\beta_h^b(\boldsymbol{\mu})$ (see e.g. [RHM13, RV07] for the proof), (5.84) directly follows. To prove (5.85), we proceed as we did in the proof of Proposition 5.4, to obtain

$$\alpha_N(\boldsymbol{\mu}) \geq \frac{\sigma}{2}\alpha_N^n(\boldsymbol{\mu})\min\left\{1,\left(\frac{\beta_N^S(\boldsymbol{\mu})}{\gamma_N^C(\boldsymbol{\mu})}\right)^2\right\}.$$

Since $\gamma_N^C(\boldsymbol{\mu}) \leq \gamma_h^C(\boldsymbol{\mu})$ and $\alpha_N^n(\boldsymbol{\mu}) \geq \alpha_h^n(\boldsymbol{\mu})$, (5.85) directly follows. Finally, we apply once again the inequality provided in [KSZ13, Th. 1] to get

$$\beta_N(\boldsymbol{\mu}) \geq \frac{\alpha_N(\boldsymbol{\mu})}{1 + \left(\gamma_N^{\mathcal{A}}(\boldsymbol{\mu})/\hat{\beta}_N(\boldsymbol{\mu})\right)^2}, \qquad \forall \boldsymbol{\mu} \in \mathcal{D},$$

from which (5.86) direcly follows, since $\gamma_N^{\mathcal{A}}(\boldsymbol{\mu}) \leq \gamma_h^{\mathcal{A}}(\boldsymbol{\mu})$. □

Because of the definition of the supremizer solutions $\mathbf{T}_\pi^{\boldsymbol{\mu}}$, the RB velocity space $V_N^{\boldsymbol{\mu}}$ (and therefore also the spaces Y_N and Q_N) still depends on $\boldsymbol{\mu}$. To get rid of this $\boldsymbol{\mu}$-dependence, following [RHM13, GV12], we rather define

$$V_N = \text{span}\Big\{\boldsymbol{v}_h(\boldsymbol{\mu}^n), \mathbf{T}^{\boldsymbol{\mu}^n}_{\pi_h(\boldsymbol{\mu}^n)}, \boldsymbol{\lambda}_h(\boldsymbol{\mu}^n), \mathbf{T}^{\boldsymbol{\mu}^n}_{\eta_h(\boldsymbol{\mu}^n)}, \quad n = 1, \dots, N\Big\}, \tag{5.87}$$

so that we consider only $2N$ parameter independent supremizer snapshots. This enables a full decoupling of the offline-online stages (and thus substantial computational savings). We cannot rigorously demonstrate that the approximation stability of the Stokes operator is preserved, despite being numerically verified [RHM13, GV12]. Thus, we obtain a RB velocity space V_N of dimension $4N$ and a RB pressure space Q_N of dimension $2N$. Therefore, since $Q_N = Y_N$, the RB state and adjoint spaces Y_N and Q_N have dimension $6N$, while the control space U_N has dimension N.

Remark 5.8. If the state operator is a general noncoercive operator (different from the Stokes operator) we can still employ the strategy above to define stable RB spaces. To this end, it is sufficient to combine a stable approximation for the state equation with the definition of aggregated state/adjoint spaces. •

5.9 Dirichlet boundary control of the Navier-Stokes equations

We now consider the following parametrized boundary control problem [HR99, GB97, GHS91]: find a triple $(\boldsymbol{v}, \pi, \boldsymbol{u})$ such that the cost functional

$$\mathcal{J}(\boldsymbol{v}, \pi, \boldsymbol{u}; \boldsymbol{\mu}) = \mathcal{F}_1(\boldsymbol{v}; \boldsymbol{\mu}) + \mathcal{F}_2(\boldsymbol{u}; \boldsymbol{\mu}) \tag{5.88}$$

is minimized subject to the steady Navier-Stokes equations

$$\begin{aligned} -\nu\Delta\boldsymbol{v} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + \nabla\pi &= \mathbf{0} && \text{in } \Omega(\boldsymbol{\mu}) \\ \nabla\cdot\boldsymbol{v} &= 0 && \text{in } \Omega(\boldsymbol{\mu}), \end{aligned} \tag{5.89}$$

with boundary conditions

$$\begin{aligned} \boldsymbol{v} &= \boldsymbol{u} && \text{on } \Gamma_C(\boldsymbol{\mu}) \\ \boldsymbol{v} &= \mathbf{0} && \text{on } \Gamma_w(\boldsymbol{\mu}) \\ -\pi\boldsymbol{n} + \nu(\nabla\boldsymbol{v})\boldsymbol{n} &= \mathbf{0} && \text{on } \Gamma_N(\boldsymbol{\mu}). \end{aligned} \tag{5.90}$$

Here $\Omega(\boldsymbol{\mu}) \subset \mathbb{R}^d$ $(d = 2, 3)$ is a parametrized spatial domain with Lipschitz boundary $\partial\Omega = \Gamma_C \cup \Gamma_N \cup \Gamma_w$, being $\Gamma_D = \Gamma_C \cup \Gamma_w$ the Dirichlet portion of the boundary and Γ_N

the Neumann one; the state variables $\boldsymbol{v}$ and π denote the velocity and pressure fields, respectively, while $\boldsymbol{u}$ acts as a Dirichlet boundary control. In the cost functional $\mathcal{J}$, $\mathcal{F}_1(\boldsymbol{v};\boldsymbol{\mu})$ represents the objective to be minimized (see Sect. 5.8), while $\mathcal{F}_2(\boldsymbol{u};\boldsymbol{\mu})$ is a regularization term.

In view of the application presented in Sect. 6.4, we also consider an additional constraint on the control variable, described by a bounded linear functional $l(\cdot;\boldsymbol{\mu}) \in U'$

$$l(\boldsymbol{u};\boldsymbol{\mu}) = C(\boldsymbol{\mu}) \qquad \forall \boldsymbol{u} \in U, \tag{5.91}$$

$C(\boldsymbol{\mu}) \in \mathbb{R}$ being a given function of the parameters. The constraint (5.91) can be used for instance to impose a desired flux on the control boundary Γ_C.

5.9.1 Weak formulation

In order to cast the optimal control problem (5.88)-(5.90) in the general formulation (5.1)-(5.2), we first define the velocity space $V = \mathbf{H}^1_D(\Omega)$, the pressure space $M = L^2(\Omega)$, and the state space $Y = V \times M$. The definition of the control space deserves a special consideration in this case. In fact, the natural choice for the control space would be $\mathcal{U} = \mathbf{H}^{1/2}(\Gamma_C) = [H^{1/2}(\Gamma_C)]^d$. However, the realization of the $\mathbf{H}^{1/2}(\Gamma_C)$-inner product would introduce undesirable computational complexities, see e.g. [OPS14] for further details. Several alternatives have been proposed to overcome this issue, for instance:

(i) considering weaker solutions by choosing $\mathcal{U} = \mathbf{L}^2(\Gamma_C)$, which requires to cast the state equation in the ultra-weak variational formulation, see e.g. [Vex07, MRV13];

(ii) adopting a penalty approach [HR99], in which the Dirichlet condition is approximated by a Robin condition which allows to set $\mathcal{U} = \mathbf{L}^2(\Gamma_C)$;

(iii) requiring further regularity by choosing $\mathcal{U} = \mathbf{H}^1_0(\Gamma_C)$, see e.g. [GHS91].

Here we follow the third approach, i.e. we assume the control variable to belong to $\mathcal{U} = \mathbf{H}^1(\Gamma_C)$ ($\mathcal{U} = \mathbf{H}^1_0(\Gamma_C)$ if Γ_C has a boundary), and then we treat the Dirichlet control employing a lifting approach; an alternative would be to employ a Lagrange multiplier approach, see e.g. [GHS91, Ded07]. To this end, we split the velocity field as $\boldsymbol{v} = \boldsymbol{v}_0 + \mathbf{R}(\boldsymbol{u})$, where $\boldsymbol{v}_0 \in \mathbf{H}^1_D(\Omega)$, $\mathbf{R}\colon \mathcal{U} \to \mathbf{H}^1_w(\Omega)$ is a bounded extension operator such that $\mathbf{R}(\boldsymbol{u}) \in \mathbf{H}^1_w(\Omega)$ and $\mathbf{R}(\boldsymbol{u})|_{\Gamma_C} = \boldsymbol{u}$. For the sake of simplicity, we still denote $\boldsymbol{v}_0$ with $\boldsymbol{v}$, as no ambiguity occurs.

We define the following trilinear forms associated with the convective operator

$$c(\boldsymbol{v}_1, \boldsymbol{v}_2, \hat{\boldsymbol{\lambda}}; \boldsymbol{\mu}) = \int_{\Omega(\boldsymbol{\mu})} (\boldsymbol{v}_1 \cdot \nabla)\boldsymbol{v}_2 \cdot \hat{\boldsymbol{\lambda}}\, d\Omega \qquad \forall \boldsymbol{v}_1, \boldsymbol{v}_2, \hat{\boldsymbol{\lambda}} \in V,$$

$$d(\boldsymbol{v}_1, \boldsymbol{v}_2, \hat{\boldsymbol{\lambda}}; \boldsymbol{\mu}) = c(\boldsymbol{v}_1, \boldsymbol{v}_2, \hat{\boldsymbol{\lambda}}; \boldsymbol{\mu}) + c(\boldsymbol{v}_2, \boldsymbol{v}_1, \hat{\boldsymbol{\lambda}}; \boldsymbol{\mu}) \qquad \forall \boldsymbol{v}_1, \boldsymbol{v}_2, \hat{\boldsymbol{\lambda}}, \in V.$$

Since the Navier-Stokes equations can be considered with values in $\widetilde{Q}' = Y'$, we set $\widetilde{Q} = Y$ and define the operator $\mathcal{E}_1(\cdot;\boldsymbol{\mu})\colon X \to \widetilde{Q}'$ as

$${}_{\widetilde{Q}'}\langle \mathcal{E}_1(x;\boldsymbol{\mu}), \hat{p} \rangle_{\widetilde{Q}} = S(\boldsymbol{v} + \mathbf{R}(\boldsymbol{u}), \pi; \hat{\boldsymbol{\lambda}}, \hat{\eta}; \boldsymbol{\mu}) + c(\boldsymbol{v} + \mathbf{R}(\boldsymbol{u}), \boldsymbol{v} + \mathbf{R}(\boldsymbol{u}), \hat{\boldsymbol{\lambda}}; \boldsymbol{\mu}), \tag{5.92}$$

where

$$S(\boldsymbol{v}, \pi; \boldsymbol{\lambda}, \eta; \boldsymbol{\mu}) = a(\boldsymbol{v}, \boldsymbol{\lambda}; \boldsymbol{\mu}) + b(\boldsymbol{\lambda}, \pi; \boldsymbol{\mu}) + b(\boldsymbol{v}, \eta; \boldsymbol{\mu})$$

represents the Stokes operator already introduced in Sect. 5.8.1. Then, we define the operator $\mathcal{E}_2(\cdot;\boldsymbol{\mu})\colon \mathcal{U} \to \mathbb{R}$ as $\mathcal{E}_2(x;\boldsymbol{\mu}) = l(\boldsymbol{u};\boldsymbol{\mu}) - C(\boldsymbol{\mu})$. This fits the abstract formulation (5.2), upon defining the operator $\mathcal{E}(\cdot;\boldsymbol{\mu}) : X \to Q'$, with $Q = \widetilde{Q} \times \mathbb{R}$, as

$$\mathcal{E}(x;\boldsymbol{\mu}) = \begin{pmatrix} \mathcal{E}_1(x;\boldsymbol{\mu}) \\ \mathcal{E}_2(x;\boldsymbol{\mu}) \end{pmatrix}. \tag{5.93}$$

Finally, let us express the quadratic functionals $\mathcal{F}_1$ and $\mathcal{F}_2$ appearing in (5.88) as

$$\mathcal{F}_1(\boldsymbol{v};\boldsymbol{\mu}) = \frac{1}{2} m(\boldsymbol{v},\boldsymbol{v};\boldsymbol{\mu}) \qquad \mathcal{F}_2(\boldsymbol{u};\boldsymbol{\mu}) = \frac{\sigma}{2} n(\boldsymbol{u},\boldsymbol{u};\boldsymbol{\mu}),$$

where $m(\cdot,\cdot;\boldsymbol{\mu})\colon\ Z \times Z \to \mathbb{R}$ is a symmetric, continuous, non-negative bilinear form over a Hilber space $Z \supset V$, $\sigma > 0$ is a given constant, while $n(\cdot,\cdot;\boldsymbol{\mu})\colon \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ is a symmetric, bounded and coercive bilinear form.

From the existence of solutions of the steady Navier-Stokes equations (5.89) (under the assumption of *small data*), see e.g. [GR86, Tem01], and the properties of $\mathcal{J}$, it follows by standard arguments (see e.g. [DI94, GHS91, Hei98]) that for each $\boldsymbol{\mu} \in \mathcal{D}$ there exists at least one optimal solution $\bar{x} \in X$.

5.9.2 Optimality conditions

Let us introduce the Lagrange functional[5] $\mathcal{L}(\cdot;\boldsymbol{\mu}) : X \times Q_1 \times \mathbb{R} \to \mathbb{R}$,

$$\mathcal{L}(x,p,\kappa;\boldsymbol{\mu}) = \mathcal{J}(x;\boldsymbol{\mu}) + \langle p, \mathcal{E}_1(x;\boldsymbol{\mu})\rangle + \kappa\, \mathcal{E}_2(x;\boldsymbol{\mu}). \tag{5.94}$$

In order to ensure the existence of Lagrange multipliers associated to an optimal solution $\bar{x}$, we have to verify the fulfillment of assumptions (H2) and (H3). As regards the former, we start by computing the Fréchet derivatives of $\mathcal{E}_1$ at $x \in X$ in directions $\delta x, \delta^2 x \in X$ (we omit the $\boldsymbol{\mu}$-dependence of the variational forms for clarity):

$$\langle \mathcal{E}_1'(x;\boldsymbol{\mu})\delta x, p\rangle = S(\delta\boldsymbol{v} + \mathbf{R}(\delta\boldsymbol{u}), \delta\pi; \boldsymbol{\lambda}, \eta) + d(\delta\boldsymbol{v} + \mathbf{R}(\delta\boldsymbol{u}), \boldsymbol{v} + \mathbf{R}(\boldsymbol{u}), \boldsymbol{\lambda}), \quad \forall p \in \widetilde{Q},$$
$$\langle \mathcal{E}_1''(x;\boldsymbol{\mu})(\delta x, \delta^2 x), p\rangle = d(\delta\boldsymbol{v} + \mathbf{R}(\delta\boldsymbol{u}), \delta^2\boldsymbol{v} + \mathbf{R}(\delta^2\boldsymbol{u}), \boldsymbol{\lambda}), \quad \forall p \in \widetilde{Q}.$$

As the nonlinearities are quadratic, all higher derivatives are identically equal to zero. For the linear constraint $\mathcal{E}_2$, it holds

$$\mathcal{E}_2'(x;\boldsymbol{\mu})\delta x = l(\delta\boldsymbol{u};\boldsymbol{\mu}), \qquad \mathcal{E}_2''(x;\boldsymbol{\mu})(\delta x, \delta^2 x) = 0.$$

Therefore, $\mathcal{E}$ is twice continuously differentiable and its second derivative is Lipschitz continuous since it does not depend on $x \in X$. The first and second derivatives of $\mathcal{J}$ at $x \in X$ are given by

$$\mathcal{J}'(x;\boldsymbol{\mu})\delta x = m(\boldsymbol{v} + \mathbf{R}(\boldsymbol{u}), \delta\boldsymbol{v} + \mathbf{R}(\delta\boldsymbol{u});\boldsymbol{\mu}) + \sigma n(\boldsymbol{u}, \delta\boldsymbol{u};\boldsymbol{\mu}), \tag{5.95}$$

$$\mathcal{J}''(x;\boldsymbol{\mu})(\delta x, \delta^2 x) = m(\delta^2\boldsymbol{v} + \mathbf{R}(\delta^2\boldsymbol{u}), \delta\boldsymbol{v} + \mathbf{R}(\delta\boldsymbol{u});\boldsymbol{\mu}) + \sigma n(\delta^2\boldsymbol{u}, \delta\boldsymbol{u};\boldsymbol{\mu}), \tag{5.96}$$

so that also $\mathcal{J}$ satisfies assumption (H2). Finally, assumption (H3), i.e. the surjectivity of the linearization of the constraint $\mathcal{E}$, can be proved by following the arguments in [GHS91, Hei98]. We end up with the following

[5] Here we denote with p the Lagrange multiplier associated to $\mathcal{E}_1$ rather than $\mathcal{E}$ as in the previous sections, as no ambiguity occurs.

Proposition 5.6. *Suppose that, for a given $\boldsymbol{\mu} \in \mathcal{D}$, $\bar{x} \in X$ is a local solution to* (5.88)-(5.89)*. Then there exist unique Lagrange multipliers $\bar{p} \in \widetilde{Q}, \bar{\kappa} \in \mathbb{R}$ such that $(\bar{x}, \bar{p}, \bar{\kappa})$ satisfy*

$$\begin{cases} \mathcal{J}'(\bar{x};\boldsymbol{\mu}) + \mathcal{E}_1'(\bar{x};\boldsymbol{\mu})^* \bar{p} + \mathcal{E}_2'(\bar{x};\boldsymbol{\mu})^* \bar{\kappa} & = 0, \quad \text{in } X' \\ \mathcal{E}_1(\bar{x};\boldsymbol{\mu}) & = 0, \quad \text{in } Q' \\ \mathcal{E}_2(\bar{x};\boldsymbol{\mu}) & = 0, \quad \text{in } \mathbb{R}. \end{cases} \tag{5.97}$$

This means that, given $\boldsymbol{\mu} \in \mathcal{D}$, solving the optimization problem (5.88)-(5.90) *requires to find $(\boldsymbol{v}, \pi; \boldsymbol{u}; \boldsymbol{\lambda}, \eta; \kappa) \in Y \times \mathcal{U} \times \widetilde{Q} \times \mathbb{R}$ such that*

$$\begin{cases} S(\hat{\boldsymbol{v}}, \hat{\pi}; \boldsymbol{\lambda}, \eta) + d(\hat{\boldsymbol{v}}, \boldsymbol{v} + \mathbf{R}(\boldsymbol{u}), \boldsymbol{\lambda}) = -m(\boldsymbol{v} + \mathbf{R}(\boldsymbol{u}), \hat{\boldsymbol{v}}), & \forall (\hat{\boldsymbol{v}}, \hat{\pi}) \in Y \\ \sigma n(\boldsymbol{u}, \hat{\boldsymbol{u}}) + d(\mathbf{R}(\hat{\boldsymbol{u}}), \boldsymbol{v} + \mathbf{R}(\boldsymbol{u}), \boldsymbol{\lambda}) + m(\boldsymbol{v} + \mathbf{R}(\boldsymbol{u}), \mathbf{R}(\hat{\boldsymbol{u}})) \\ \qquad = -\kappa l(\hat{\boldsymbol{u}};\boldsymbol{\mu}) - S(\mathbf{R}(\hat{\boldsymbol{u}}), 0; \boldsymbol{\lambda}, \eta), & \forall \hat{\boldsymbol{u}} \in \mathcal{U} \\ S(\boldsymbol{v} + \mathbf{R}(\boldsymbol{u}), \pi; \hat{\boldsymbol{\lambda}}, \hat{\eta}) + c(\boldsymbol{v} + \mathbf{R}(\boldsymbol{u}), \boldsymbol{v} + \mathbf{R}(\boldsymbol{u}), \hat{\boldsymbol{\lambda}}) = 0, & \forall (\hat{\boldsymbol{\lambda}}, \hat{\eta}) \in \widetilde{Q} \\ l(\boldsymbol{u};\boldsymbol{\mu}) = C(\boldsymbol{\mu}). \end{cases} \tag{5.98}$$

For the proof of the coercivity of the Lagrangian (necessary for the fulfillment of condition (H4)) we refer to [DI94, Hei98] and references therein. We simply report the expression of the Hessian of the Lagrangian at $x \in X$ in the direction $\delta x \in X$

$$\begin{aligned} \langle \mathcal{L}_{xx}(x, p; \boldsymbol{\mu}) \delta x, \delta x \rangle = \mathcal{J}''(x;\boldsymbol{\mu})(\delta x, \delta x) + \langle \mathcal{E}_1''(x;\boldsymbol{\mu})(\delta x, \delta x), p \rangle = \sigma n(\delta \boldsymbol{u}, \delta \boldsymbol{u}; \boldsymbol{\mu}) \\ + m(\delta \boldsymbol{v} + \mathbf{R}(\delta \boldsymbol{u}), \delta \boldsymbol{v} + \mathbf{R}(\delta \boldsymbol{u}); \boldsymbol{\mu}) + d(\delta \boldsymbol{v} + \mathbf{R}(\delta \boldsymbol{u}), \delta \boldsymbol{v} + \mathbf{R}(\delta \boldsymbol{u}), \boldsymbol{\lambda}; \boldsymbol{\mu}). \end{aligned}$$

5.9.3 Finite element approximation

For the high-fidelity approximation of (5.88)-(5.90), we use $\mathbb{P}_1$-$\mathbb{P}_1$ finite element spaces with Dohrmann-Bochev stabilization [DB04, BDG06] to approximate velocity and pressure variables, as well as $\mathbb{P}_1$ finite elements to discretize the control variable. With respect to the choice of Sect. 5.8.3, here we use stabilized low-order (rather than stable) velocity and pressure spaces to lower the computational costs entailed by the solution of the high-fidelity problem. We remark that since the pressure stabilization is symmetric, optimize-then-discretize and discretize-then-optimize approaches commute in this case (see, e.g., [ABH04, Bra09]).

Moreover, for the numerical experiments discussed in Chap. 6, a continuation strategy with respect to Reynolds number (see e.g. [HR99, BG05b]) proves to be sufficient to globalize Newton method, without resorting to suitable line search or trust region algorithms (see e.g. [NW06, DHV98] and reference therein).

5.9.4 Reduced spaces definition

In order to build reduced spaces satisfying assumptions (A1)-(A3), we employ a slightly modified version of the aggregated strategy presented in Sect. 5.8.4: we use aggregated pressure and velocity spaces for the state and adjoint variables as in [NMR15], but without enriching the velocity space by supremizer solutions. Indeed, in all our numerical experiments the pressure stabilization demonstrates to guarantee the solvability of the state operator also at the reduced level, without the need of enriching the velocity space.

We first describe the construction using the greedy algorithm. We denote by $S_N = \{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N\}$ the parameter samples selected by the greedy algorithm and consider the corresponding (suitably orthonormalized) full-order solutions $(x_h(\boldsymbol{\mu}^n),\, p_h(\boldsymbol{\mu}^n))$, $1 \le n \le N$. We define the aggregated pressure and velocity spaces M_N, V_N as

$$M_N = \text{span}\Big\{\pi_h(\boldsymbol{\mu}^n),\, \eta_h(\boldsymbol{\mu}^n)\Big\}_{n=1}^N, \qquad V_N = \text{span}\Big\{\boldsymbol{v}_h(\boldsymbol{\mu}^n), \boldsymbol{\lambda}_h(\boldsymbol{\mu}^n)\Big\}_{n=1}^N.$$

Then, we define the reduced space $\mathcal{U}_N = \text{span}\{\boldsymbol{u}_h(\boldsymbol{\mu}^n)\}_{n=1}^N$ for the control variable.

If a POD approach is used instead, we first compute full-order solutions $(x_h(\boldsymbol{\mu}_i),\, p_h(\boldsymbol{\mu}_i))$, $i = 1, \cdots, N_s$ of the optimality system for N_s parameters value $\boldsymbol{\mu}_i$ chosen by LHS sampling. Then, we perform POD separately on each variable in order to obtain (i) the reduced state velocity and pressure spaces V_N^s, M_N^s; (ii) the reduced adjoint velocity and pressure spaces V_N^a, M_N^a; (iii) the reduced control space $\mathcal{U}_N$. Finally, we define the aggregated spaces as:

$$M_N = M_N^s \cup M_N^a, \qquad V_N = V_N^s \cup V_N^a.$$

In both cases, we finally set

$$Y_N = V_N \times M_N, \qquad X_N = Y_N \times \mathcal{U}_N, \qquad \widetilde{Q}_N = Y_N, \qquad \mathcal{X}_N = X_N \times \widetilde{Q}_N \times \mathbb{R},$$

so that the reduced space $\mathcal{X}_N$ has dimension $9N + 1$.

5.9.5 Error estimates

Let us denote by $\rho_h = \rho_h(\Omega)$ the *discrete* $L^4(\Omega)$-$H^1(\Omega)$ Sobolev embedding constant [Man14]; moreover we denote by V_h a suitable finite element subspace for V. Then, the continuity of the trilinear form $c(\cdot,\cdot,\cdot;\boldsymbol{\mu})$ yields

$$c(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{\lambda}; \boldsymbol{\mu}) \le \rho_h^2 M_c(\boldsymbol{\mu}) \|\boldsymbol{v}_1\|_V \|\boldsymbol{v}_2\|_V \|\boldsymbol{\lambda}\|_V, \qquad \forall \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{\lambda} \in V_h,$$

where $M_c(\boldsymbol{\mu})$ is a function depending on the parametrization (see [Man14] for further details).

Proposition 5.7. *The Lipschitz constant of the Fréchet derivative $dG[\cdot](\cdot,\cdot;\boldsymbol{\mu})$ associated to problem* (5.88)-(5.89) *is bounded by the positive function*

$$K_h^N(\boldsymbol{\mu}) = 6\rho_h^2 M_c(\boldsymbol{\mu}). \tag{5.99}$$

Moreover, the error on the cost functional is bounded by

$$|\mathcal{J}_h(\boldsymbol{\mu}) - \mathcal{J}_N(\boldsymbol{\mu})| \le \Delta_N^{\mathcal{J}}(\boldsymbol{\mu}) + 6\rho_h^2 M_c(\boldsymbol{\mu})(\Delta_N(\boldsymbol{\mu}))^3, \tag{5.100}$$

with $\Delta_N^{\mathcal{J}}$ defined as in (5.46).

Proof. In order to estimate the Lipschitz constant $K_h^N(\boldsymbol{\mu})$ it is sufficient to exploit the continuity of the trilinear form $c(\cdot,\cdot,\cdot;\boldsymbol{\mu})$ and the definition of $d(\cdot,\cdot,\cdot;\boldsymbol{\mu})$,

$$\begin{aligned}
|dG[U_1](\delta U, \widehat{U}; \boldsymbol{\mu}) - dG[U_2](\delta U, \widehat{U}; \boldsymbol{\mu})| &= |dG[U_1 - U_2](\delta U, \widehat{U}; \boldsymbol{\mu})| \\
&\le |d(\delta\boldsymbol{v} + \mathbf{R}(\delta\boldsymbol{u}), \boldsymbol{v}_1 - \boldsymbol{v}_2 + \mathbf{R}(\boldsymbol{u}_1) - \mathbf{R}(\boldsymbol{u}_2), \hat{\boldsymbol{\lambda}}; \boldsymbol{\mu})| \\
&\quad + |d(\delta\boldsymbol{v} + \mathbf{R}(\delta\boldsymbol{u}), \hat{\boldsymbol{v}} + \mathbf{R}(\hat{\boldsymbol{u}}), \boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2; \boldsymbol{\mu})| \\
&\quad + |d(\hat{\boldsymbol{v}} + \mathbf{R}(\hat{\boldsymbol{u}}), \boldsymbol{v}_1 - \boldsymbol{v}_2 + \mathbf{R}(\boldsymbol{u}_1) - \mathbf{R}(\boldsymbol{u}_2), \delta\boldsymbol{\lambda}; \boldsymbol{\mu})| \\
&\le 6\rho_h^2 M_c(\boldsymbol{\mu})\, \|U_1 - U_2\|_{\mathcal{X}} \|\delta U\|_{\mathcal{X}} \|\widehat{U}\|_{\mathcal{X}},
\end{aligned}$$

so that $K_h^N(\boldsymbol{\mu}) = 6\rho_h^2 M_c(\boldsymbol{\mu})$ (which actually does not depend on N). In order to estimate the remainder term in (5.48), we first compute the second derivative of $G(\cdot,\cdot;\boldsymbol{\mu})$ at U,

$$d^2G[U](\widehat{U},\widehat{U},\widehat{U};\boldsymbol{\mu}) = 3\,d(\hat{\boldsymbol{v}}+\mathbf{R}(\hat{\boldsymbol{u}}),\hat{\boldsymbol{v}}+\mathbf{R}(\hat{\boldsymbol{u}}),\hat{\boldsymbol{\lambda}};\boldsymbol{\mu}).$$

Then, by the continuity of $d(\cdot,\cdot,\cdot;\boldsymbol{\mu})$ we obtain

$$\begin{aligned}|\mathcal{R}(E_N(\boldsymbol{\mu});\boldsymbol{\mu})| &\leq \sup_{W\in[U_N,U_h]} |d^2G[W](E_N(\boldsymbol{\mu}),E_N(\boldsymbol{\mu}),E_N(\boldsymbol{\mu});\boldsymbol{\mu})| \\ &\leq 6\rho_h^2 M_c(\boldsymbol{\mu})\|E_N(\boldsymbol{\mu})\|_{\mathcal{X}}^3,\end{aligned}$$

and (5.100) easily follows. □

6 Parametrized optimization problems in fluid dynamics

In this chapter, we apply the methodology presented in Chap. 5 to a variety of optimization problems arising in fluid dynamics. To begin with, we consider the optimization of a steady heat conduction-convection problem which models the temperature of a fluid flowing into a heat exchanger device. The second problem we consider deals with the reconstruction, from experimental observations provided by eco-doppler measurements, of the blood velocity field across a two dimensional section of a carotid artery. Finally, we apply our reduction strategy to a vorticity minimization problem for a bluff body immersed in a two and three-dimensional flow, and a two-dimensional Dirichlet boundary control problem modeling the optimization of a simplified arterial bypass graft.

6.1 An optimal heat transfer problem

We consider a steady heat conduction-convection problem which models the temperature of a fluid flowing into a heat exchanger device, like the one shown in Fig. 6.1. Our goal is to regulate the temperature $u = u(\mathbf{x})$ imposed on the three baffles of the heat exchanger in such a way that the temperature distribution $y = y(\mathbf{x})$ approaches as much as possible

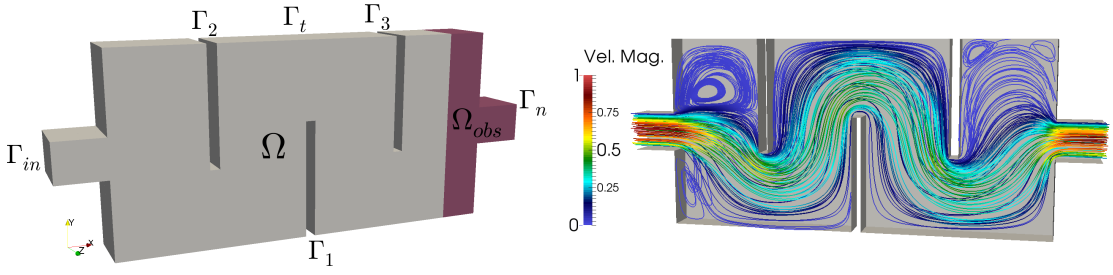


Fig. 6.1 Left: Computational domain Ω and boundaries; the dark red portion of the domain identifies the observation subdomain Ω_{obs}, while the control boundary Γ_C is given by the union of the three baffles Γ_1, Γ_2 and Γ_3. Right: streamlines of the convective field $\boldsymbol{b}$, which is obtained from solving a stationary Navier-Stokes problem for a Reynolds number equal to 100.

a desired temperature y_d in the outflow region Ω_{obs} of the domain Ω. To this end, we consider the following cost functional,

$$\mathcal{J}(y, u; \boldsymbol{\mu}) = \frac{1}{2} \int_{\Omega_{\text{obs}}} (y - y_d)^2 \, d\Omega + \frac{\sigma}{2} \int_{\Gamma_C} \left(|\nabla_\Gamma u|^2 + u^2 \right) d\Gamma, \tag{6.1}$$

where ∇_Γ denotes the surface gradient operator on Γ_C, see e.g. [DZ11]. The first terms penalizes the misfit between the desired and the predicted temperature, while the second one penalizes rapid variations and high values of the control variable u. The state and control variables are linked together by the following state problem,

$$\begin{cases} -\alpha \Delta y + \boldsymbol{v} \cdot \nabla y = 0 & \text{in } \Omega \\ y = u & \text{on } \Gamma_C \\ y = 0 & \text{on } \Gamma_w \cup \Gamma_{in} \\ \alpha \nabla y \cdot \boldsymbol{n} = h & \text{on } \Gamma_t \\ \alpha \nabla y \cdot \boldsymbol{n} = 0 & \text{on } \Gamma_n, \end{cases} \tag{6.2}$$

where the domain $\Omega \subset \mathbb{R}^3$ and its boundaries are displayed in Fig. 6.1, α is the thermal diffusivity, while the convective field $\boldsymbol{v}$ represents the (prescribed) velocity of the flow field across the exchanger. For low Reynolds numbers, the latter can be obtained as the solution of the stationary Navier-Stokes equations (see Fig. 6.1). The control variable u acts as a Dirichlet datum and we impose a (given) non-zero heat flux h on the top wall Γ_t. Problem (6.1)-(6.2) thus belongs to the class of optimization problems constrained by linear elliptic PDEs described in Sect. 5.7.

We consider $P = 4$ parameters: $\mu_1 = u_d \in [2, 12]$ is the desired temperature, $\mu_2 = 1/\sigma \in [5, 50]$ is the penalization constant in the cost functional, $\mu_3 = h \in [0, 0.5]$ is the heat flux imposed on Γ_t, while $\mu_4 \in [1, 500]$ is the Péclet number. The latter is defined as $\text{Pe} = VL/\nu$, where $V = 1$ is a characteristic velocity, $L = 1$ is a characteristic length and α is the diffusion coefficient. The fluid enters the channel with a reference temperature $y = 0$, then the heating process is regulated by the temperature values u imposed on the baffles Γ_1, Γ_2 and Γ_3, respectively.

The problem is discretized by piecewise linear finite elements for the state, control and adjoint variables, leading to a discretized optimality system (5.25) of dimension 90 408 which admits an affine decomposition with $Q_g = 7$ and $Q_d = 3$ terms.

For the construction of the RB spaces we employ the greedy procedure detailed in Sect. 5.7.2. The algorithm selects $N = 37$ sample points with a fixed tolerance $\varepsilon_{tol} = 10^{-3}$ so that $\Delta_N(\boldsymbol{\mu}) \leq \varepsilon_{tol} \ \forall \boldsymbol{\mu} \in \Xi_{\text{train}}$, where Ξ_{train} is a training set of $5 \cdot 10^3$ random points. In Fig. 6.3 we compare the error estimate $\Delta_N(\boldsymbol{\mu})$ with the true error between the high-fidelity and RB approximations, as well as the error estimate $\Delta_N^{\mathcal{J}}(\boldsymbol{\mu})$ with the error on the cost functional $|\mathcal{J}_h(\boldsymbol{\mu}) - \mathcal{J}_N(\boldsymbol{\mu})|$. In Fig. 6.2 some representative RB solutions are shown, while the computational details are reported in Table 6.1. We remark that a speedup of three orders of magnitude is achieved, still ensuring a very high accuracy.

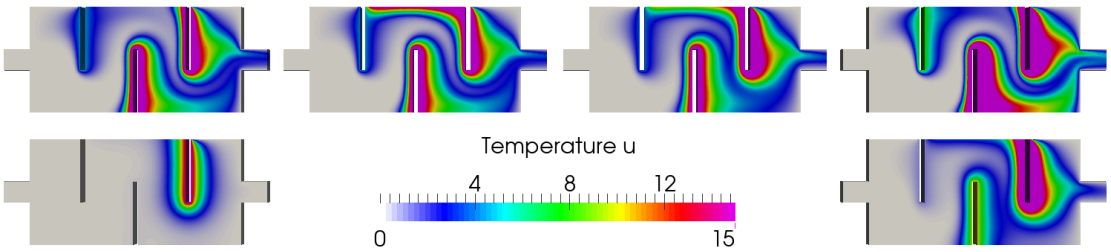


Fig. 6.2 RB state solutions of (6.1)-(6.2) for different parameter values (with $\mu_1 = 8$ fixed).

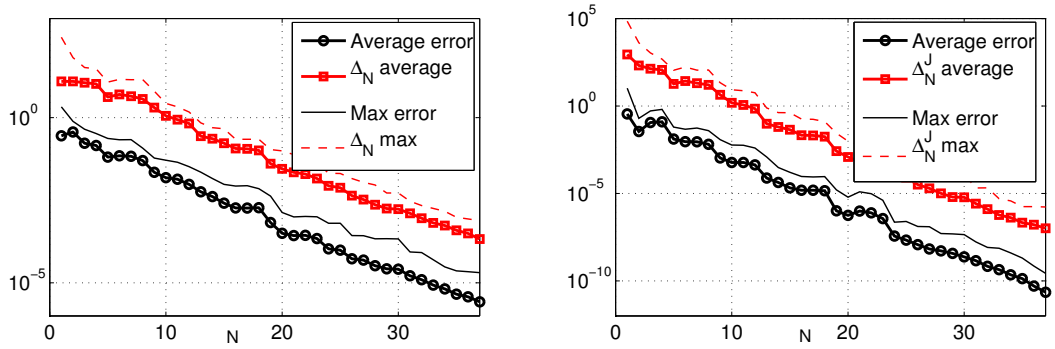


Fig. 6.3 Left: average and max computed errors and estimate between the high-fidelity and RB approximations. Right: average and max error and estimate between $\mathcal{J}_h$ and $\mathcal{J}_N$. Computations have been performed over a test sample set of 200 random points.

Table 6.1 Computational details for the high-fidelity and RB approximations of (6.1)-(6.2). Both offline and online computations are performed on a workstation with a Intel Core i5-2400S processor and 16 GB of RAM.

High-fidelity model		Reduced order model	
Number of FE dofs	90 408	Number of RB dofs	$37 \cdot 5$
Number of parameters P	4	Dofs reduction	488:1
Affine components Q_g	7	Offline greedy time	1317 s
Error tolerance greedy ε_{tol}	10^{-3}	RB online solution	3 ms
FE solution time (assembling + solution)	≈ 10 s	RB online estimation	7 ms

6.2 A surface reconstruction problem

The second problem we consider, originally proposed in [ADSS11, ANSS14], deals with the reconstruction, from areal data provided by eco-dopplers measurements, of the blood velocity field in a section of a carotid artery. Specifically, given a set of velocity measurements (in the horizontal direction for example) in some portions of the domain, we aim at reconstructing a global velocity profile on the whole section. The problem can be seen as a problem of surface estimation starting from scattered data, with the peculiarity that the estimated surface should preserve a physiological meaning. Therefore the technique employed for the reconstruction should take into account the shape of the domain and preserve the no-slip condition of the velocity field on the boundary of

the domain. For this reason, the authors in [ADSS11] point out that classical surface estimation methods like thin-plate splines, tensor product splines, etc, are not well suited to tackle the problem at hand and therefore they propose to apply a smoothing technique based on the minimization of a suitable PDE-penalized least-square cost functional. Here we want to provide a flexible geometrical and computational framework enabling the rapid resolution of the problem for different shapes of the carotid sections and for different values of the observations.

Let us describe the problem at hand. We consider a domain $\Omega \subset \mathbb{R}^2$ (the carotid cross section) and we denote by $y : \Omega \to \mathbb{R}$ the unknown surface that we want to reconstruct starting from a given set of areal observations

$$z_i = \frac{1}{|\Omega_{obs,i}|} \int_{\Omega_{obs,i}} y(\mathbf{x})\, d\Omega, \qquad i = 1, \dots, m,$$

being $\{\Omega_{obs,i}\}_{i=1}^m$ nonoverlapping subdomains of observation defined as

$$\Omega_{obs,i} = \{(x_1, x_2) \,|\, (x_1 - x_{i,1})^2 + (x_2 - x_{i,2})^2 \leq r^2\},$$

i.e. small circles surrounding the observation points $\mathbf{x}_i$, see Fig. 6.4. Since the surface we want to estimate represent the horizontal component of the blood flow in the carotid artery, thus satisfying a no-slip condition on $\partial\Omega$, we shall require the variable y to vanish on the boundary, i.e. $y|_{\partial\Omega} = 0$. In [ADSS11, Ram02] the authors propose to minimize the following PDE-penalized least-square cost functional in order to recover the surface $y \in H_0^2(\Omega)$

$$\min_{y \in H_0^2(\Omega)} J(y) = \frac{1}{2} \sum_{i=1}^{m} \int_{\Omega_{obs,i}} |y - z_i|^2 d\Omega + \frac{\sigma}{2} \int_{\Omega} (\Delta y)^2 \, d\Omega. \tag{6.3}$$

Problem (6.3) can be easily cast in the framework of linear-quadratic optimal control problems considered in Sect. 5.7 by defining the control variable $u = -\Delta y$ and expressing explicitly the boundary condition satisfied by the state variable y. We end up with the following equivalent problem: given the domain Ω and a set of observation values $\{z_i\}_{i=1}^m$,

$$\min_{y,u} \mathcal{J}(y, u) = \frac{1}{2} \sum_{i=1}^{m} \int_{\Omega_{obs,i}} |y - z_i|^2 d\Omega + \frac{\sigma}{2} \int_{\Omega} u^2 \, d\Omega, \tag{6.4}$$

where $y = y(u) \in Y$ is the solution of the following Poisson problem

$$\begin{cases} -\Delta y = u & \text{in } \Omega \\ y = 0 & \text{on } \partial\Omega, \end{cases} \tag{6.5}$$

being $\mathcal{U} = L^2(\Omega)$ the control space and $Y = H_0^1(\Omega)$ the space for the state variable. As already mentioned, our goal here is twofold:

- describe different configurations of the section of the carotid artery through a low dimensional shape parametrization, thus yielding a geometrical reduction;
- apply the computational reduction framework of Chap. 5 to provide a rapid resolution of the surface estimation problem.

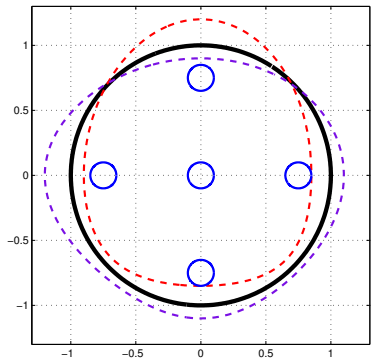


Fig. 6.4 Reference domain $\widetilde{\Omega} = \{(x_1, x_2) \,|\, x_1^2 + x_2^2 \leq 1\}$ in black, examples of deformations of the reference domain in red and violet; fixed observation subdomains in blue.

6.2.1 Geometrical reduction

In order to describe different configurations of the section of the carotid artery by using only a small number of parameters, we introduce a suitable shape parametrization based on the *free-form deformation* (FFD) technique [MQR12b]. In particular, the set of

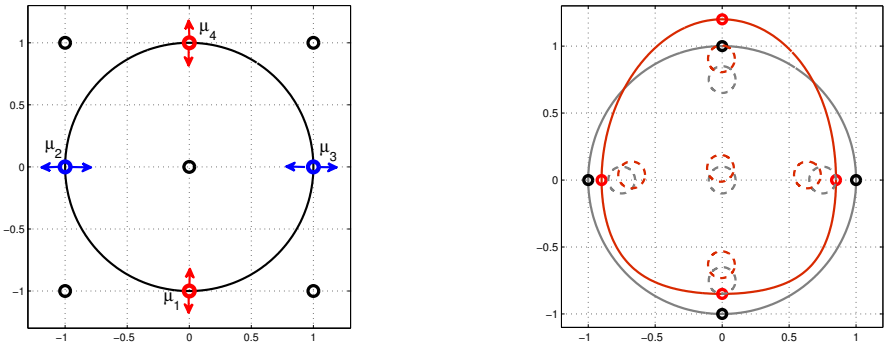


Fig. 6.5 Schematic diagram of the FFD. On the left: reference domain $\widetilde{\Omega}$ and FFD setting; control points depicted in red/blue can be moved in vertical/horizontal direction. On the right: in gray the reference domain Ω, in red a deformed domain $\Omega(\boldsymbol{\mu}_g)$; black control points correspond to the reference shape (i.e. $\boldsymbol{\mu}_g = 0$), while red control points correspond to the choice $\boldsymbol{\mu}_g = [0.15, 0.1, -0.15, 0.2]$. The gray and red dashed circles correspond to the five observation subdomains in the reference and deformed configurations respectively.

admissible shapes is defined as the set of diffeomorphic images of a reference domain $\widetilde{\Omega}$ through a parametrized map $T(\mathbf{x}; \boldsymbol{\mu}_g)$ depending on a set of control points acting as shape design parameters. In our case, the FFD map is built on a 3×3 lattice of control points on the rectangle $[-1, 1] \times [1, 1]$; the active control points and their admissible displacements (see Fig. 6.5) are selected in order to describe reasonable and plausible deformations of the reference circular shape. In the end we use a parametrization with 4 design parameters $\boldsymbol{\mu}_g$ representing the vertical/horizontal displacements of selected control points. These parameters are allowed to vary in the range $[-0.15, 0.15]$, thus yielding the geometrical parameters set

$$\mathcal{D}_g = [-0.15, 0.15]^4.$$

We remark that, since the parametrized map $T(\cdot;\boldsymbol{\mu}_g) : \widetilde{\Omega} \to \Omega(\boldsymbol{\mu}_g)$ is a global geometrical map, the observation subdomains $\Omega_{obs,i}$ are translated and deformed by action of the map itself, see Fig. 6.5. This kind of behavior is clearly a drawback of the use of the FFD technique and it should be avoided in view of a realistic application. In fact, considering the functionality of the clinical device employed to get the measurements, i.e. the eco-doppler, it seems more reasonable to consider fixed observation subdomains. However, the employment of different shape parametrization techniques, such as those based on radial basis functions (see e.g. [MQR12a]), should easily allow to overcome this issue.

We also consider as input parameters the set of observations $\{z_i\}_{i=1}^m$, i.e. we define the parameters $\mu_{obs}^i = z_i$ for $1 \leq i \leq m$ and the observation parameter space

$$\mathcal{D}_{obs} = [-0.25, 0.25]^m.$$

We finally obtain the following parametrized optimal control problem: given $\boldsymbol{\mu} = (\boldsymbol{\mu}_g, \boldsymbol{\mu}_{obs}) \in \mathcal{D} = \mathcal{D}_g \times \mathcal{D}_{obs}$,

$$\min_{y,u} \mathcal{J}(y(\boldsymbol{\mu}), u(\boldsymbol{\mu}); \boldsymbol{\mu}) = \frac{1}{2} \sum_{i=1}^{m} \int_{\Omega_{obs,i}(\boldsymbol{\mu}_g)} |y(\boldsymbol{\mu}) - \mu_{obs}^i|^2 d\Omega + \frac{\sigma}{2} \int_{\Omega(\boldsymbol{\mu}_g)} u(\boldsymbol{\mu})^2 \, d\Omega, \quad (6.6)$$

where $y(\boldsymbol{\mu})$ is the solution of the following state equation

$$\begin{cases} -\Delta y(\boldsymbol{\mu}) = u(\boldsymbol{\mu}) & \text{in } \Omega(\boldsymbol{\mu}_g), \\ y(\boldsymbol{\mu}) = 0 & \text{on } \partial\Omega(\boldsymbol{\mu}_g). \end{cases} \quad (6.7)$$

Upon defining the adjoint space $Q = H_0^1(\Omega(\boldsymbol{\mu}))$, the product space $X = Y \times \mathcal{U}$, and introducing the weak formulation of the state equation, problem (6.6)-(6.7) can be expressed in the form (5.1). Specifically, it belongs to the class of optimization problems constrained by linear elliptic PDEs described in Sect. 5.7.

6.2.2 System approximation

For the high-fidelity approximation of (6.6)-(6.7) we employ $\mathbb{P}_1$ finite element spaces for both the state, control and adjoint variables. The total number of degrees of freedom is $N_h = 43\,461$, obtained using a mesh made of 29 224 triangular elements; in particular, the dimension of the control space is $\dim(\mathcal{U}_h) = 14\ 487$. The resulting optimality system reads (see Sect. 5.2.2): given $\boldsymbol{\mu} \in \mathcal{D}$, find $\mathbf{U}_h = (\mathbf{y}_h, \boldsymbol{u}_h, \mathbf{p}_h)^T \in \mathbb{R}^{N_h}$ such that

$$\mathbf{K}(\boldsymbol{\mu})\mathbf{U}_h = \mathbf{F}(\boldsymbol{\mu}),$$

with

$$\mathbf{K}(\boldsymbol{\mu}) = \begin{pmatrix} \mathbf{M}_{\text{obs}}(\boldsymbol{\mu}) & 0 & \mathbf{A}^T(\boldsymbol{\mu}) \\ 0 & \sigma\mathbf{M}(\boldsymbol{\mu}) & -\mathbf{M}(\boldsymbol{\mu}) \\ \mathbf{A}(\boldsymbol{\mu}) & -\mathbf{M}(\boldsymbol{\mu}) & 0 \end{pmatrix}, \qquad \mathbf{F}(\boldsymbol{\mu}) = \begin{pmatrix} \mathbf{f}_{\text{obs}}(\boldsymbol{\mu}) \\ 0 \\ 0 \end{pmatrix}. \quad (6.8)$$

Here, $\mathbf{A}(\boldsymbol{\mu})$ results from the discretization of the state operator, $\mathbf{M}(\boldsymbol{\mu})$ is the mass matrix, while $\mathbf{M}_{\text{obs}}(\boldsymbol{\mu})$ is the matrix resulting from the discretization of the observation operator. Moreover, we set the penalization constant to $\sigma = 10^{-4}$. Because of the geometrical parametrization, all these matrices – as well as the right-hand side – feature a

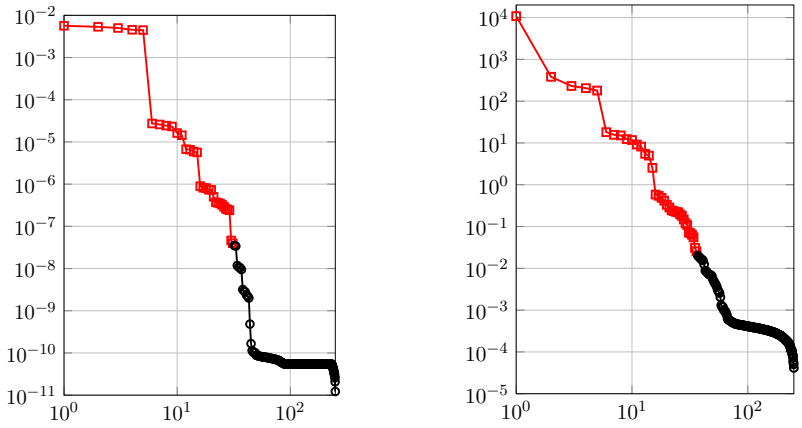


Fig. 6.6 Decay of the singular values of vector (left) and matrix (right) snapshots for the surface reconstruction problem. Red squares correspond to the retained modes, while black circles correspond to the discarded ones.

nonaffine parametric dependence. We thus resort to the system approximation techniques introduced in Sect. 3.3 to restore an approximated affine structure[1]. In particular, we approximate $\mathbf{K}(\boldsymbol{\mu})$ by MDEIM and $\mathbf{F}(\boldsymbol{\mu})$ by DEIM, following the *system approximation then state-space reduction* approach presented in Sect. 3.4.1.

We start by computing a set of 250 matrix and vector snapshots corresponding to 250 parameter samples selected by latin hypercube sampling in $\mathcal{D}$. The eigenvalues of the correlation matrices of matrix and vector snapshots are reported in Fig 6.6. Using a tolerance of 10^{-5}, POD retains the first $M_k = 36$ and $M_f = 31$ modes. We then run MDEIM and DEIM on the resulting matrix and vector bases. Fig. 6.7(a) shows the entries of the matrix $\mathbf{K}(\boldsymbol{\mu})$ selected by MDEIM, while the corresponding reduced mesh is reported in Fig. 6.7(b). Globally, about 1.5% of the original elements are selected.

We end up with the following high-fidelity model with system approximation: given $\boldsymbol{\mu} \in \mathcal{D}$, find $\mathbf{U}_h^m \in \mathbb{R}^{N_h}$ such that

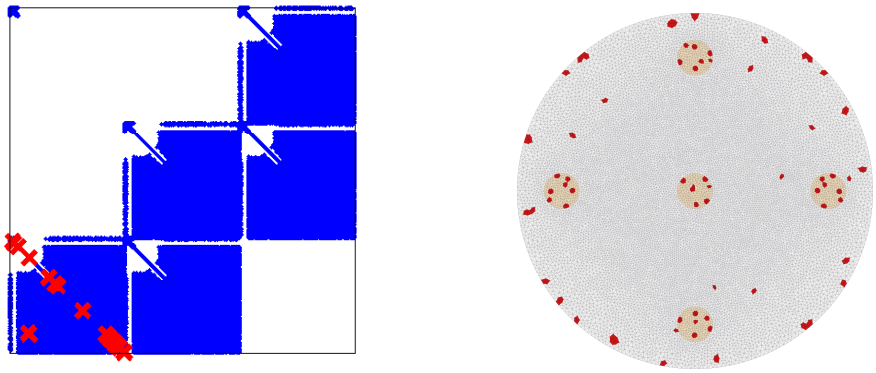
$$\mathbf{K}_m(\boldsymbol{\mu})\mathbf{U}_h^m = \mathbf{F}_m(\boldsymbol{\mu}), \tag{6.9}$$

where $\mathbf{K}_m(\boldsymbol{\mu})$ is the MDEIM approximation of $\mathbf{K}(\boldsymbol{\mu})$ and $\mathbf{F}_m(\boldsymbol{\mu})$ is the DEIM approximation of $\mathbf{F}(\boldsymbol{\mu})$.

6.2.3 Reduced basis approximation

For the construction of the reduced space $\mathcal{X}_N$, rather than employing the greedy algorithm as in Sect. 6.1, we use the POD-based strategy described in Sect. 5.7.2. To this end, we first solve offline the approximated FOM (6.9) to obtain a set of $N_s = 150$ solution snapshots (corresponding to 150 parameter configurations selected by latin hypercube sampling). Then, we perform POD separately on state, adjoint and control snapshots retaining $N = 50$ modes for each variable. After constructing the aggregated spaces as described in Sect 5.7.2, we end up with a reduced problem of dimension $5N$. Finally, we compute all the quantities entering in the evaluation of the error estimate in the online phase.

[1] In [RMN12] an affine structure was instead recovered by means of an EIM-based approach.



(a) sparsity pattern of the matrix $\mathbf{K}(\boldsymbol{\mu})$ (blue) and MDEIM selected entries (red)

(b) reduced elements (red) and observation subdomains (light brown)

Fig. 6.7 Matrix entries selected by MDEIM and corresponding reduced elements.

Table 6.2 Computational details for the surface reconstruction problem (offline time is about 50 minutes using 2 cores). Both offline and online computations are performed on a workstation with a Intel Core i5-2400S processor and 16 GB of RAM.

High-fidelity model		Reduced order model	
Number of FE dofs N_h	43 461	Number of ROM dofs N	$50 \cdot 5$
Number of parameters P	9	Dofs reduction	173:1
Affine (MDEIM) components M_k	36	Reduced mesh assembly	0.019 s
Affine (DEIM) components M_f	31	ROM online solution	0.028 s
FE assembling + solving time	3 s	ROM online estimation	0.25 s

To assess the ROM accuracy, in Fig. 6.8 we report the average errors and estimates over a testing set of 500 samples; on average, the relative system approximation error is below 0.1%. In Fig. 6.9 some representative examples of reconstructed surfaces obtained by solving the ROM are given. The average time for a single resolution of the optimal control problem is around 28 ms, while the online evaluation of the a posteriori error bound, requires around 0.25 s. As a comparison we report also the time required by the FE solver (see also Table 6.2): the resolution of the optimal control problem without any kind of reduction, neither the system approximation nor the space reduction, takes about 3 seconds, since it requires every time to deform the mesh, assemble the FE system and solve the optimality system; exploiting the geometrical reduction and the system approximation also for the finite element solver yields a speedup factor two. Globally, a speedup of about two orders of magnitude is achieved.

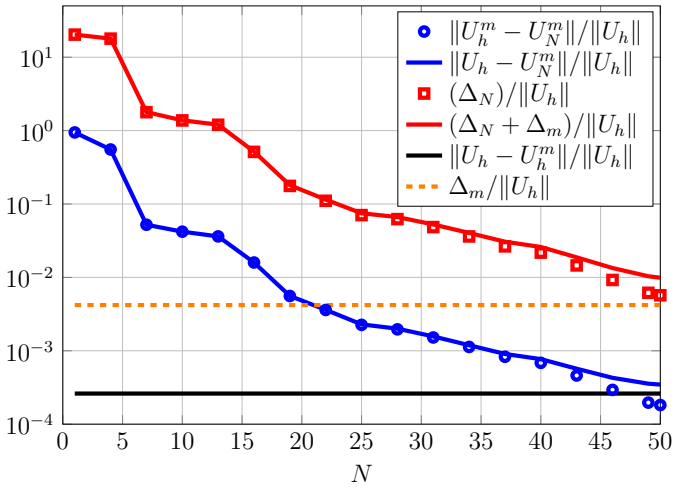


Fig. 6.8 Relative error and estimates averaged over a testing set of 500 random points in $\mathcal{D}$. The notation in the legend refers to the one of Sect. 3.4.

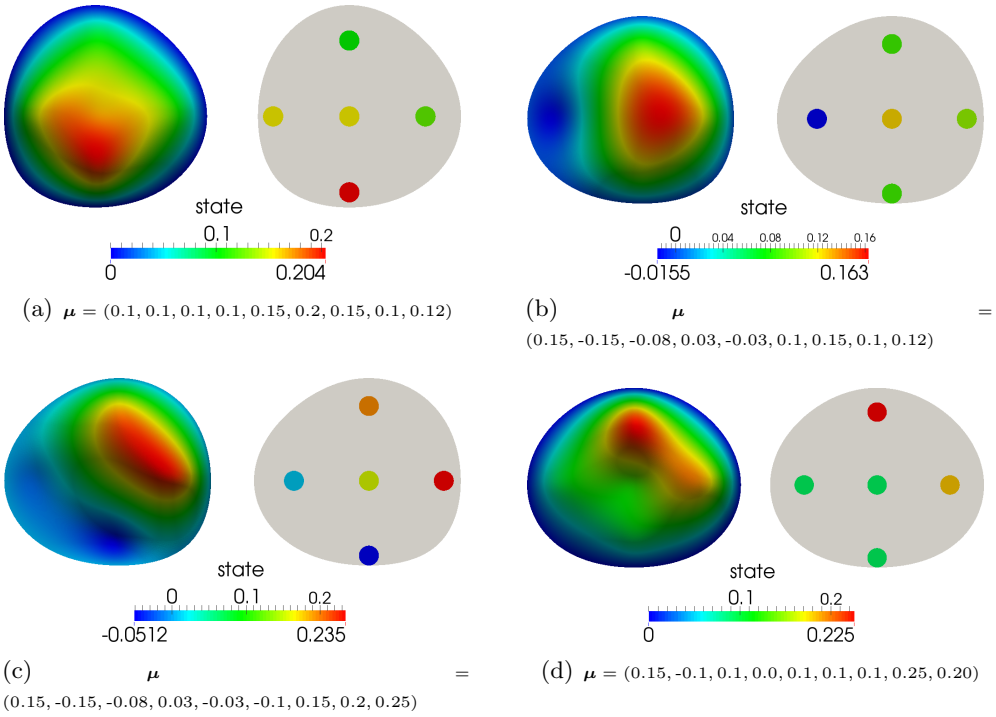


Fig. 6.9 Reconstructed surface for different geometries and observation values, i.e. different values of the parameters $\boldsymbol{\mu}$. In each subfigure we report the observation values on the right and the reconstructed surface on the left.

6.3 Vorticity minimization around a bluff body: the Stokes case

In this section, we deal with a problem of vorticity minimization through suction/injection of fluid on the downstream portion of a bluff body, see e.g. [GB97, Ded07]. In particular, we consider a body embedded in a viscous flow at low Reynolds number governed by the steady incompressible Stokes equations. The goal consists in minimizing the vorticity (and thus the drag, as a result) in the wake of the body, by regulating the flow across a portion of its boundary Γ_C. In particular, we minimize the following cost functional,

$$\mathcal{J}(\boldsymbol{v}, u; \boldsymbol{\mu}) = \frac{1}{2} \int_{\Omega_{\text{obs}}(\mu_1)} |\nabla \times \boldsymbol{v}|^2 \, d\Omega + \frac{1}{2\mu_3} \mathcal{F}_2(u; \boldsymbol{\mu}), \tag{6.10}$$

subject to the steady Stokes equations

$$\begin{aligned} -\nu \Delta \boldsymbol{v} + \nabla \pi &= \boldsymbol{0} && \text{in } \Omega(\mu_1) \\ \operatorname{div} \boldsymbol{v} &= 0 && \text{in } \Omega(\mu_1), \end{aligned} \tag{6.11}$$

together with the following boundary conditions

$$\begin{aligned} \boldsymbol{v} \cdot \boldsymbol{t} = 0, \ \boldsymbol{v} \cdot \boldsymbol{n} = u \quad & \text{on } \Gamma_C(\mu_1) \\ \boldsymbol{v} = \mu_2 \boldsymbol{t} \quad & \text{on } \Gamma_{in} \\ \boldsymbol{v} = \boldsymbol{0} \quad & \text{on } \Gamma_w \\ \boldsymbol{v} \cdot \boldsymbol{n} = \boldsymbol{0}, \ (\nabla \boldsymbol{v}) \boldsymbol{n} \cdot \boldsymbol{t} = \boldsymbol{0} \quad & \text{on } \Gamma_s(\mu_1) \\ -\pi \boldsymbol{n} + \nu (\nabla \boldsymbol{v}) \boldsymbol{n} = \boldsymbol{0} \quad & \text{on } \Gamma_N, \end{aligned} \tag{6.12}$$

where $\boldsymbol{n}$ and $\boldsymbol{t}$ are the outward normal and tangential unit vectors to the boundary. We impose an horizontal constant velocity profile on the inflow boundary Γ_{in}, no-slip conditions on Γ_w, symmetry conditions on Γ_s, no-stress conditions on Γ_N and Dirichlet conditions on the control boundary Γ_C. In particular, we consider suction/injection of fluid through the control boundary only in the normal direction, while we impose a no-slip condition in the tangential one.

We first consider a two-dimensional flow and then a three-dimensional flow, see Figs. 6.10 and 6.14, respectively, for the details about the domain and boundaries. The parameters are given by: the length μ_1 of the control boundary, the magnitude μ_2 of the inflow velocity profile and the inverse of the penalization factor μ_3.

Following the discussion of Sect. 5.9.1, we assume the control variable to belong to $U = H_0^1(\Gamma_C)$, and we treat the Dirichlet control employing a lifting approach. As a result, the penalization term in the cost functional is given by

$$\mathcal{F}_2(u; \boldsymbol{\mu}) = \int_{\Gamma_C(\mu_1)} |\nabla_\Gamma u|^2 \, d\Gamma.$$

6.3.1 Two-dimensional flow

We first deal with the two-dimensional version of the problem, i.e. we take $\Omega(\mu_1) \subset \mathbb{R}^2$ and consider a two-dimensional profile for the body, see Fig. 6.10. The length of the

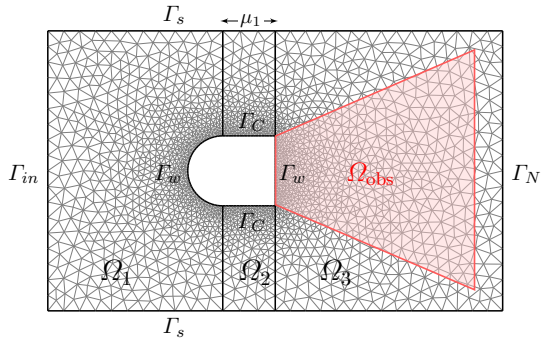


Fig. 6.10 Domain, boundaries, observation region (in red) and computational mesh (less refined than the one actually used in the computations) for the 2D problem.

control boundary can vary in the range $\mu_1 \in [0.1, 0.35]$, the magnitude of the inflow velocity profile $\mu_2 \in [0.5, 5]$, while the penalization constant $\mu_3 \in [1, 10^3]$. To handle the geometric parametrization and provide an affine decomposition of the problem, we divide the domain $\Omega(\mu_1)$ into three subdomains Ω_1, $\Omega_2(\mu_1)$ and $\Omega_3(\mu_1)$, see Fig. 6.10. Provided this decomposition of the domain, we can easily build an affine geometrical mapping such that, by tracing the problem back to the reference domain $\Omega(\bar{\mu}_1)$ with $\bar{\mu}_1 = 0.15$, we obtain a parametrized decomposition with $Q_d = 14$ and $Q_g = 23$ terms.

For the finite element discretization, we use a $\mathbb{P}_2$-$\mathbb{P}_1$ approximation for the velocity and pressure variables, respectively, and a $\mathbb{P}_2$ approximation (obtained as the restriction on Γ_C of the velocity FE space) for the control variable. The total number of degrees of freedom, i.e. the size of the full-order optimality system, is $N_h = 99\,288$, obtained using a mesh made of 11 055 triangular elements.

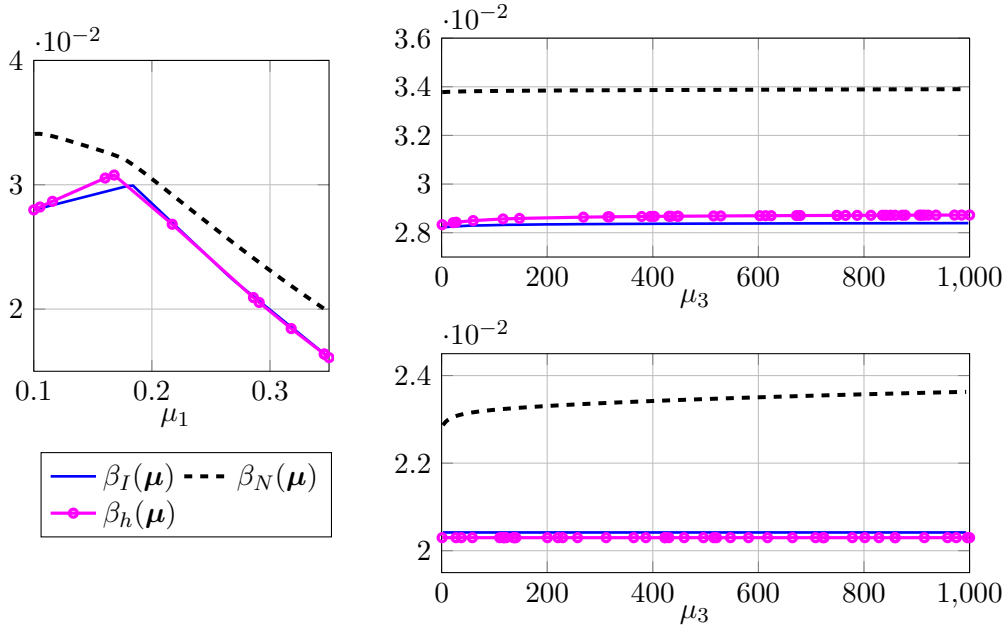


Fig. 6.11 2D flow. Comparison between the FE stability factor $\beta_h(\boldsymbol{\mu})$ (the *dots* represent computed values), the interpolant surrogate $\beta_I(\boldsymbol{\mu})$ and the RB stability factor $\beta_N(\boldsymbol{\mu})$. Left: stability factors as functions of μ_1, $\mu_3 = 100$ fixed. Right: stability factors as functions of μ_3 with $\mu_1 = 0.1$ (top) and $\mu_1 = 0.3$ (bottom).

We employ the heuristic strategy of Sect. 2.5.1 to compute an approximation of the stability factor $\beta_h(\boldsymbol{\mu})$ of the optimality system. However, rather than employing RBF interpolation, we construct $\beta_I(\boldsymbol{\mu})$ by a simpler linear interpolation. Specifically, since the parameter μ_2 does not affect the value of $\hat{\beta}_h(\boldsymbol{\mu})$, we perform a two dimensional interpolation with respect to the parameters μ_1 and μ_3, using an equally spaced grid of 4×12 interpolation points. In Fig 6.11 we report the resulting interpolant $\beta_I(\boldsymbol{\mu})$, which proves to be a sharp approximation of the stability factor $\beta_h(\boldsymbol{\mu})$ (despite not being a lower bound, as can be seen in the figure).

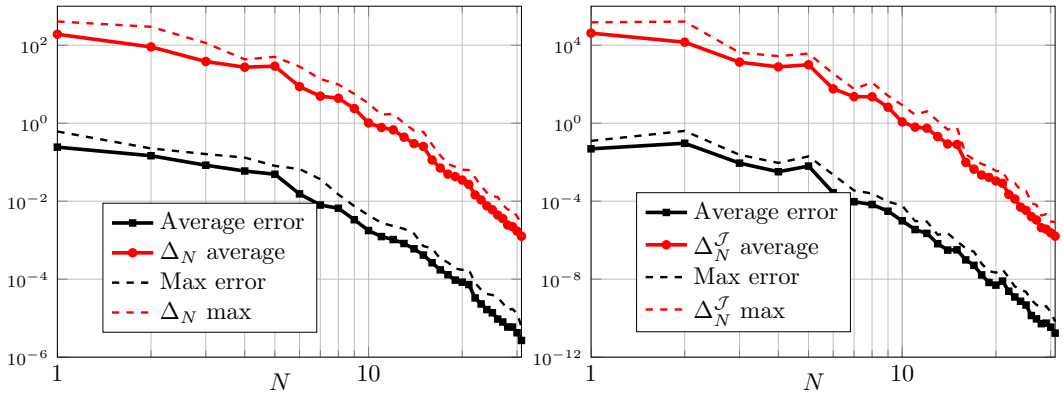


Fig. 6.12 2D flow. Left: average and max computed relative errors and bound $\Delta_N(\boldsymbol{\mu})$ between the full-order FE solution and the RB approximation. Right: average and max relative error and bound $\Delta_N^{\mathcal{J}}(\boldsymbol{\mu})$ between $\mathcal{J}_h(\boldsymbol{\mu})$ and $\mathcal{J}_N(\boldsymbol{\mu})$. Computations have been performed over a test sample set of 300 random points.

The greedy procedure for the construction of the RB spaces selects $N_{max} = 31$ sample points with a fixed tolerance $\varepsilon_{tol} = 5 \cdot 10^{-3}$ so that $\Delta_{N_{max}}(\boldsymbol{\mu}) \leq \varepsilon_{tol}\ \forall \boldsymbol{\mu} \in \Xi_{\text{train}}$. Here $\Xi_{\text{train}} \subset \mathcal{D}$ is a training set of $4 \cdot 10^4$ random points, sufficiently fine so that the number N_{max} of basis functions required to achieve the tolerance ε_{tol} does not vary by adding further points. In Fig. 6.11, the RB stability factor $\beta_N(\boldsymbol{\mu})$ defined in (5.83) is also reported: we can observe that $\beta_N(\boldsymbol{\mu}) \geq \beta_h(\boldsymbol{\mu})$, thus confirming numerically the good stability properties of the RB approximation.

In Fig. 6.12 we compare the error estimate $\Delta_N(\boldsymbol{\mu})$ with the true error between the FE and RB approximations, as well as the error estimate $\Delta_N^{\mathcal{J}}(\boldsymbol{\mu})$ with the error on the cost functional $|\mathcal{J}_h(\boldsymbol{\mu}) - \mathcal{J}_N(\boldsymbol{\mu})|$. In Fig. 6.13 some representative RB solutions are shown; as expected (see e.g. [Ded07] for a comparison), the optimal controls correspond to an aspiration of the flow through the control boundary.

As regards the computational performances, the time spent in the offline computations is about two hours[2], while the solution of the reduced optimality system (of dimension 403×403) requires only 0.03 s (see Table 6.3). Let us remark that only a small fraction (less than 10%) of the time spent performing the greedy procedure is devoted to actually solving the full-order optimization problem, rather most of the time is spent computing

[2] All the full-order optimality systems are solved in one-shot using the sparse direct solver provided by MATLAB. Parallelism is exploited to speed up the assembly of the FE matrices, the evaluation of the stability factor, the calculation of the terms required to compute the dual norm of the residual and the evaluation of the a posteriori error estimate. For the 2D problem, computations have been performed using 8 cores on a node of the SuperB cluster at EPF Lausanne.

the quantities required to evaluate the dual norm of the residual and then evaluating the error estimate over the training set.

Table 6.3 Numerical details for the 2D-flow. The RB spaces have been built by means of the greedy procedure and $N = 31$ samples points have been selected. For both offline and online computations we report the corresponding elapsed times.

Approximation data		Computational performances	
Number of FE dofs N_h	99 288	Number of RB dofs	403
Number of parameters P	3	Dofs reduction	248:1
Error tolerance greedy ε_{tol}	$5 \cdot 10^{-3}$	Stability factor interp. time	415 s
Affine operator components Q_d	14	Offline greedy time	6 095 s
Affine rhs components Q_g	23	RB online solution	0.03 s
FE solution time	≈ 10 s	RB online estimation	0.32 s

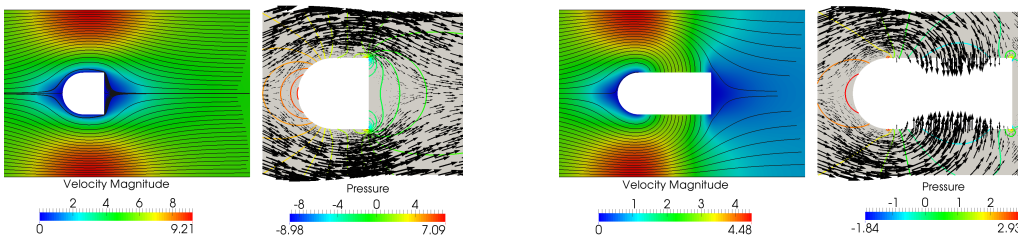


Fig. 6.13 2D flow. RB optimal state and control solutions for $\boldsymbol{\mu} = (0.1, 5, 10)$ (left) and $\boldsymbol{\mu} = (0.35, 2.5, 100)$ (right). For each case, we report the velocity magnitude with the streamlines on the left, the pressure contours and the velocity vectors around the body on the right. For the second configuration, the high value of μ_3 allows a significant flow suction on the control boundary, thus resulting in a low velocity profile at the outflow.

6.3.2 Three-dimensional flow

We now consider the three-dimensional version of the problem, whose geometry is obtained by extruding the 2D-geometry in the orthogonal direction to the plane (x_1, x_2); we introduce suitable symmetry boundary conditions so that only a quarter of the geometry is meshed, see Fig. 6.14. In this case, we consider a slightly different cost functional, minimizing the viscous energy dissipation rather than the vorticity: the two functionals serve the same purpose, but the latter would lead to an even larger affine decomposition compared to the two-dimensional case.

The parameters and their range of variations are the same as in the previous problem, except that for the geometrical parameter μ_1, which can now vary in a smaller range, $\mu_1 \in [0.1, 0.3]$. We employ a similar decomposition of the geometry into three subdomains to obtain an affine decomposition with $Q_k = 14$ and $Q_g = 9$. In this case, even using a rather coarse mesh, the dimension of the full-order optimality system rapidly grows when using stable $\mathbb{P}_2$-$\mathbb{P}_1$ or $\mathbb{P}_1^b$-$\mathbb{P}_1$ FE spaces for the velocity and pressure variables. Therefore, in order to alleviate the offline computational effort, we use low order $\mathbb{P}_1$-$\mathbb{P}_1$ spaces with Dohrmann-Bochev stabilization [DB04, BDG06] (see also Sect. 5.9.3); the control variable is now discretized with $\mathbb{P}_1$ finite elements. In this way, using a mesh of 91 394 tetrahedral

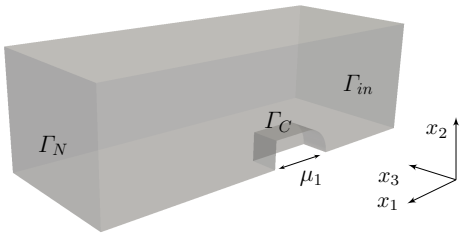


Fig. 6.14 Domain and boundaries for the 3D problem. On the body boundary we impose a no-slip condition except that on the control region; on the top, bottom and lateral boundaries of the domain we impose symmetry conditions.

elements, the total number of degrees of freedom is $N_h = 125\,266$ (for a comparison, using a $\mathbb{P}_1^b$ or $\mathbb{P}_2$ approximation for the velocity we would need $N_h \approx 600\,000$).

As in the previous case, in order to compute the interpolant $\beta_I(\boldsymbol{\mu})$ of the stability factor, we perform a two dimensional linear interpolation with respect to μ_1 and μ_3, using an equally spaced grid of 4×6 interpolation points. The greedy algorithm for the construction of the RB spaces selects $N_{max} = 20$ sample points with a fixed tolerance $\varepsilon_{tol} = 5 \cdot 10^{-3}$ (here Ξ_{train} is a set of $2 \cdot 10^4$ random points). The online evaluation of the errors and the corresponding estimates are reported in Fig. 6.15. In Fig. 6.16 we show some optimal control and state solutions obtained for different values of the parameters; note that in this case the control variable $u = u(x_1, x_3)$ is distributed over the surface $\Gamma_C(\mu_1)$.

Table 6.4 Numerical details for the 3D flow. The RB spaces have been built by means of the greedy procedure and $N = 20$ samples points have been selected. For both offline and online computations we report the corresponding elapsed times.

Approximation data		Computational performances	
Number of FE dofs N_h	125 266	Number of RB dofs	260
Number of parameters P	3	Dofs reduction	481:1
Error tolerance greedy ε_{tol}	$5 \cdot 10^{-3}$	Stability factor interp. time	592 s
Affine operator components Q_d	14	Offline greedy time	4296 s
Affine rhs components Q_g	23	RB online solution	0.026 s
FE solution time	≈ 35 s	RB online estimation	0.3 s

As regards the computational performances, the time spent offline to build the RB spaces is about 1.5 hours[3], while the solution of the reduced optimality system (of dimension 260×260) requires only $0.026\,s$, yet providing a good accuracy (see Table 6.4). As a result, in the online stage, we can perform the optimization in different scenarios, and thus evaluate the dependence of the optimal solution w.r.t the parameters in a very rapid way. For instance, in Fig. 6.17 we report the value of the cost functional $\mathcal{J}_N(\boldsymbol{\mu})$ as a function of μ_1 and μ_3, $\mu_2 = 3$ being fixed; using the ROM, it takes only 20 seconds to solve the optimization problem and to evaluate the cost functional on a grid of 40×15 points in the parameter space.

[3]In this case we have used 12 cores on a node of the SuperB cluster at EPF Lausanne.

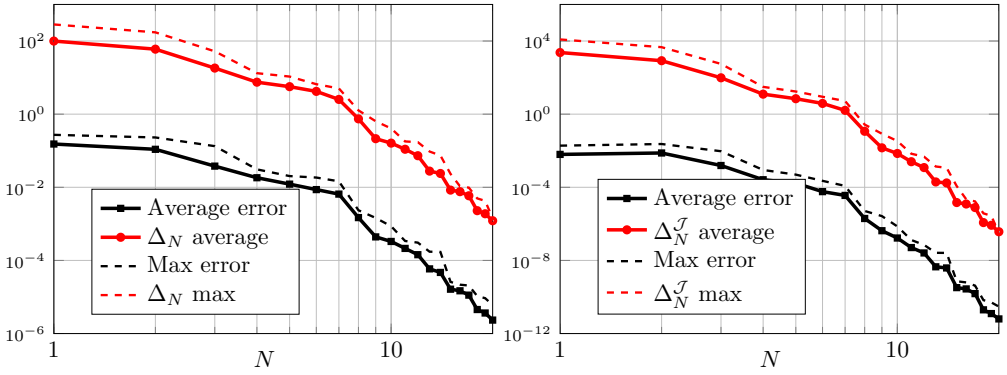


Fig. 6.15 3D flow. Left: average and max computed relative errors and bound $\Delta_N(\boldsymbol{\mu})$ between the full-order FE solution and the RB approximation. Right: average and max relative error and bound $\Delta_N^{\mathcal{J}}(\boldsymbol{\mu})$ between $\mathcal{J}_h(\boldsymbol{\mu})$ and $\mathcal{J}_N(\boldsymbol{\mu})$. Computations have been performed over a test sample set of 200 random points.

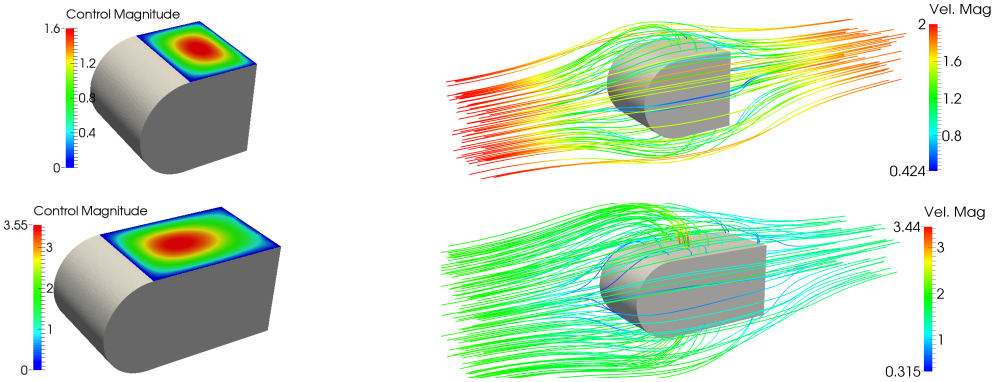


Fig. 6.16 3D flow. RB control and state solutions for $\boldsymbol{\mu} = (0.15, 2, 600)$ (top) and $\boldsymbol{\mu} = (0.3, 2, 1000)$ (bottom); for each case, we report the optimal control $u = u(x_1, x_3)$ on the upper control boundary (left) and the resulting state velocity streamlines around the body (right).

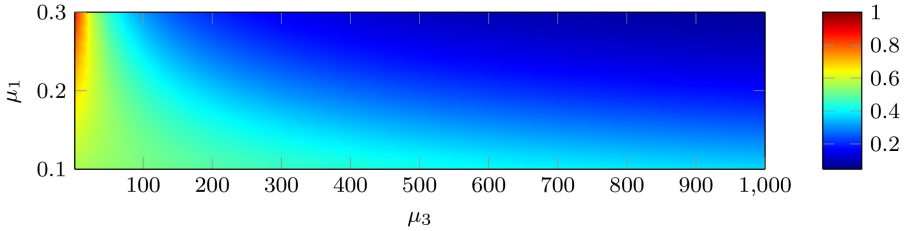


Fig. 6.17 3D flow. We report the value of the cost functional $\mathcal{J}_N(\boldsymbol{\mu})$ w.r.t (μ_1, μ_3), where $\mu_2 = 3$ is fixed; $\mathcal{J}_N(\boldsymbol{\mu})$ here is normalized to its maximum. As expected, $\mathcal{J}_N(\boldsymbol{\mu})$ decreases as the length of the control boundary μ_1 increases and the penalization factor $1/\mu_3$ decreases.

6.4 Application to a bypass graft design problem

We recall from Sect. 4.2 that bypass grafting is a surgical procedure to create an alternate channel for blood flow, bypassing an obstructed or damaged portion of a vessel. However, it is well known (see, e.g., [ODM12]) that arterial bypass grafts tend to fail after some

years due to restenosis formation. Ideally, the design of bypass grafts should aim at minimizing suitable haemodynamics indicators of the restenosis risk, such as the wall shear stress or the vorticity downstream the anastomosis. The optimization process is typically performed with respect to some geometrical design variable like the anastomosis angle or the graft-to-host diameter ratio. As recently proposed in [LMQR13], we rather follow a different approach which is based on the solution of a suitable optimal boundary control problem, for which the control function is the Dirichlet boundary condition representing the flow entering into the artery from the graft on the boundary Γ_C (see Fig. 6.18). Thus, the geometrical properties of the bypass graft are encoded into the velocity profile $\boldsymbol{u}$ imposed at the bypass anastomosis, so that the shape optimization problem is turned into an optimal control one.

We consider an idealized two-dimensional partially occluded artery as in Fig. 6.18. The optimal control problem reads: seek $(\boldsymbol{v}, \pi, \boldsymbol{u})$ such that the cost functional

$$\mathcal{J}(\boldsymbol{v}, \boldsymbol{u}; \boldsymbol{\mu}) = \frac{1}{2} \int_{\Omega_{\text{obs}}(\boldsymbol{\mu})} |\nabla \times \boldsymbol{v}|^2 \, d\Omega + \frac{1}{2\mu_3} \int_{\Gamma_C(\boldsymbol{\mu})} |\nabla_\Gamma \boldsymbol{u}|^2 d\Gamma \tag{6.13}$$

is minimized subject to the steady Navier-Stokes equations (5.89) together with the following boundary conditions:

$$\begin{aligned} -\pi \boldsymbol{n} + \nu(\nabla \boldsymbol{v})\boldsymbol{n} &= \boldsymbol{0} && \text{on } \Gamma_N \\ \boldsymbol{v} &= \boldsymbol{0} && \text{on } \Gamma_w(\boldsymbol{\mu}) \\ \boldsymbol{v} &= \boldsymbol{g}_{\text{res}}(\boldsymbol{\mu}) && \text{on } \Gamma_D \\ \boldsymbol{v} &= \boldsymbol{u} && \text{on } \Gamma_C(\boldsymbol{\mu}). \end{aligned} \tag{6.14}$$

Moreover, in order to obtain a physically meaningful problem, we enforce the total conservation of fluxes by adding the following constraint on the control variable

$$\int_{\Gamma_C(\boldsymbol{\mu})} \boldsymbol{u} \cdot \boldsymbol{n} \, d\Gamma = C(\boldsymbol{\mu}) := C_T - \int_{\Gamma_D} \boldsymbol{g}_{\text{res}}(\mu_2) \, d\Gamma, \tag{6.15}$$

being $C_T = 1$ the physiological flow rate of the host artery.

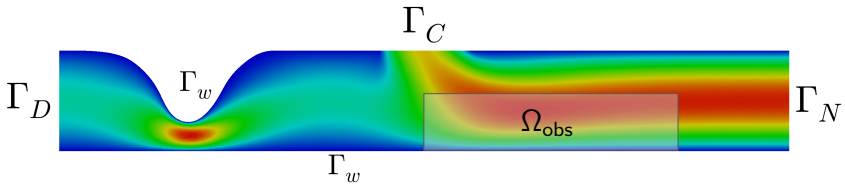


Fig. 6.18 Domain and boundaries for the bypass problem.

We consider the following parameters: the inverse of the kinematic viscosity $\mu_1 = 1/\nu \in [40, 100]$; the percentage $\mu_2 \in [0, 40]$ of residual flow $\boldsymbol{g}_{\text{res}}(\mu_2) = 6\mu_2/100\, y(1-y)$ in the host artery; the penalization parameter $\mu_3 \in [0.05, 10]$ in the cost functional; the length of the control boundary $\mu_4 \in [0.5, 1.2]$ (modeling the graft diameter). To handle the geometric parametrization and provide an affine decomposition of the problem, we divide the domain $\Omega(\mu_4)$ into three subdomains Ω_1, $\Omega_2(\mu_4)$ and $\Omega_3(\mu_4)$, see Figure 6.18. Provided this decomposition of the domain, we can easily build an affine geometrical

mapping such that, by tracing the problem back to the reference domain $\Omega = \Omega(\bar{\mu}_4)$ with $\bar{\mu}_4 = 0.8$, we obtain the affine decomposition (5.37)-(5.38) with $Q_d = 19$ and $Q_g = 28$.

For the finite element approximation, we use a mesh made of 30 926 triangular elements and 15 843 vertices, so that the total number of degrees of freedom is $N_h = 92\,239$; the dimension of the control space is $\dim(\mathcal{U}_h) = 104$.

6.4.1 Strong form of the optimality system

After integration by parts, it can be shown (see [BG05b, GHS91] for similar problems) that system (5.97) corresponds to the weak formulation for the coupled boundary value problem formed by the state equation (5.89)-(6.14), the adjoint equation

$$\begin{cases} -\dfrac{1}{\mu_1}\Delta\boldsymbol{\lambda} + (\nabla\boldsymbol{v})^T\boldsymbol{\lambda} - (\boldsymbol{v}\cdot\nabla)\boldsymbol{\lambda} + \nabla\eta = \chi_{obs}\nabla\times(\nabla\times\boldsymbol{v}) & \text{in } \Omega(\boldsymbol{\mu}) \\ \nabla\cdot\boldsymbol{\lambda} = 0 & \text{in } \Omega(\boldsymbol{\mu}) \\ -\eta\boldsymbol{n} + \nu(\nabla\boldsymbol{\lambda})\boldsymbol{n} + (\boldsymbol{v}\cdot\boldsymbol{n})\boldsymbol{\lambda} = \boldsymbol{0} & \text{on } \Gamma_N \\ \boldsymbol{\lambda} = \boldsymbol{0} & \text{on } \partial\Omega(\boldsymbol{\mu})\setminus\Gamma_N, \end{cases} \tag{6.16}$$

the optimality equation,

$$\begin{cases} -\mu_3\Delta_\Gamma\boldsymbol{u} + \kappa\boldsymbol{n} = \eta\boldsymbol{n} - \dfrac{1}{\mu_1}(\nabla\boldsymbol{\lambda})\boldsymbol{n} & \text{on } \Gamma_C(\boldsymbol{\mu}) \\ \boldsymbol{u} = \boldsymbol{0} & \text{on } \partial\Gamma_C(\boldsymbol{\mu}), \end{cases} \tag{6.17}$$

and the integral constraint (6.15) expressing the conservation of fluxes. In (6.17), we have denoted by Δ_Γ the Laplace-Beltrami operator on $\Gamma_C(\boldsymbol{\mu})$, see e.g. [DZ11].

6.4.2 Assessment of the error estimates

In our first test case we consider as parameter only the inverse of the viscosity $\mu_1 \in [40, 100]$, fixing the others to[4] $\mu_2 = 30$, $\mu_3 = 1$, $\mu_4 = 0.8$. We first compute an approximation of the stability factor by constructing the interpolant surrogate: using the adaptive procedure detailed in [MN15], $\beta_I(\boldsymbol{\mu})$ is built by RBF interpolation of $\beta_h(\boldsymbol{\mu})$ computed in 5 interpolation points. Then, we run the greedy algorithm to construct the ROM, using $\Delta_N(\boldsymbol{\mu})$ as error estimate. Through this procedure, we select $N_{\max} = 12$ sample points with a fixed tolerance $\varepsilon_{tol} = 10^{-5}$ so that $\Delta_{N_{\max}}(\boldsymbol{\mu}) \leq \varepsilon_{tol}\ \forall\boldsymbol{\mu} \in \Xi_{\text{train}}$, being Ξ_{train} a training set of 250 random points.

In Fig. 6.19 we compare, for $N = 1, \dots, N_{\max}$, the error bound $\Delta_N(\boldsymbol{\mu})$ with the true error between the full and reduced-order solutions: the estimate correctly reproduces the convergence of the error, however it shows a large effectivity $\eta_N(\boldsymbol{\mu})$ of order 10^3. Moreover, the proximity indicator $\tau_N(\boldsymbol{\mu})$ is smaller than one only for (on average) $N \geq 9$, so that the error bound $\Delta_N(\boldsymbol{\mu})$ is not available even when the true error is considerably small (for instance below 10^{-2}). In Fig. 6.19 we also report a pool of 240 test points $(\Delta_N^L(\boldsymbol{\mu}), \|E_N(\boldsymbol{\mu})\|)$ for different parameter values, while Fig. 6.20 shows the distributions of the error $\|E_N(\boldsymbol{\mu})\|_{\mathcal{X}}$ and the distribution of the linear estimator $\Delta_N^L(\boldsymbol{\mu})$ as functions of μ_1 for different values of N. It is rather evident the strong correlation of the error

[4] In this case the affine decomposition reduces to $Q_g = 6$ and $Q_d = 3$ terms.

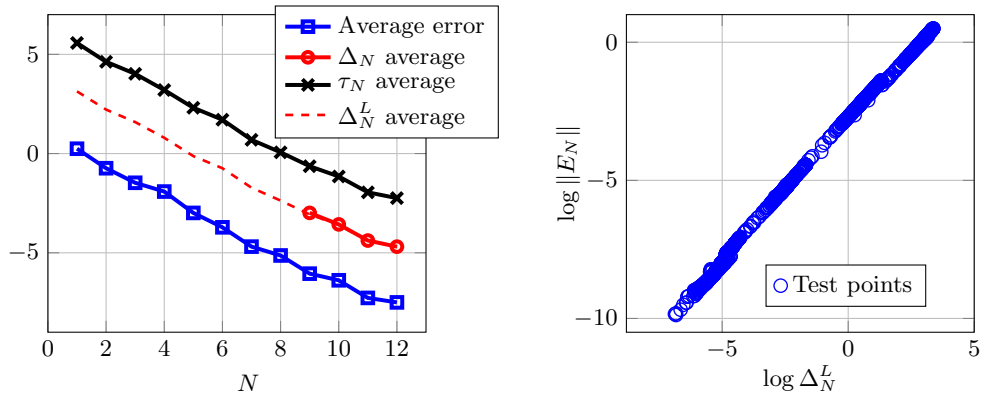


Fig. 6.19 Bypass problem, 1 parameter case. On the left: average absolute error and estimate over a testing set of 80 random points in the parameter space (vertical axis in log scale). On the right: plot of the error $\|E_N(\boldsymbol{\mu})\|_{\mathcal{X}}$ versus the linear estimator $\Delta_N^L(\boldsymbol{\mu})$ (computed for $N = 1, 4, 8$ and 80 random parameter values).

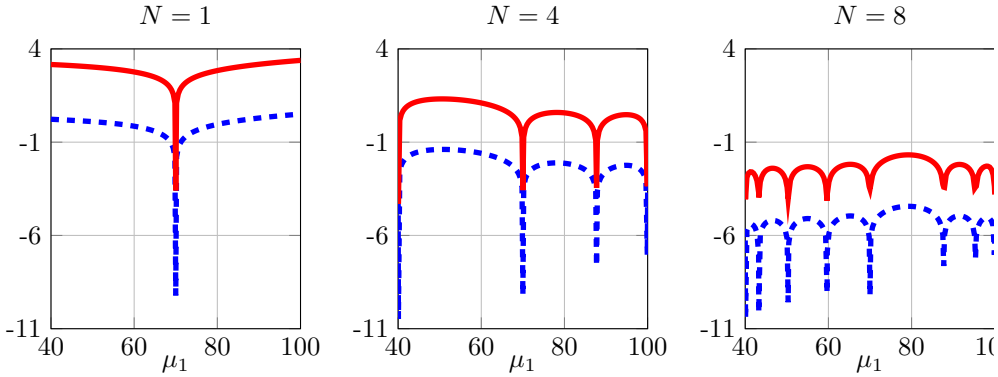


Fig. 6.20 Bypass problem, 1 parameter case. Error $\|E_N(\boldsymbol{\mu})\|_{\mathcal{X}}$ (dashed blue line) and linear estimator $\Delta_N^L(\boldsymbol{\mu})$ (solid red line) for $N = 1, 4, 8$ as functions of μ_1.

with $\Delta_N^L(\boldsymbol{\mu})$, which can therefore be used as a reliable error indicator in combination with the regression model described in Sect. 5.6.

Thus, considering the same setup, we now employ Algorithm 5.1 to build the reduced spaces. Thanks to the improved sharpness of the error indicator, the procedure selects only $N_{\max} = 7$ sample points to achieve a tolerance $\varepsilon_{tol} = 10^{-5}$ on the maximum error over Ξ_{train}. From a computational standpoint, we gain both in the offline phase, where 5 iterations of the greedy algorithm are avoided, and in the online phase, where a smaller system has to be solved. As a result, the solution of the reduced optimization problem takes about 0.03 s, while the high-fidelity one requires on average 90 seconds to be solved[5].

In Fig. 6.21 we report the error behavior versus N, as well as its distribution in the parameter space for $N = 7$. The final regression model obtained at $N = 7$ is reported in Fig. 6.22 (left), where a set of 560 test points (generated from the convergence

[5] All the full-order linear systems are solved in one-shot using the sparse direct solver provided by MATLAB. Offline computations are performed on a node (with two Intel Xeon E5-2660 processors and 64 GB of RAM) of the SuperB cluster at EPFL. Online computations are performed on a workstation with a Intel Core i5-2400S processor and 16 GB of RAM.

analysis of Fig. 6.21, left) is also shown. We observe that the regression model slightly underestimates the true error in the untrained region corresponding to $\Delta_N^L \in [10^{-4}, 10^{-1}]$. This is mainly due to the *reproductive* training points $\{(\Delta_n^L(\boldsymbol{\mu}^n), \|E_n(\boldsymbol{\mu}^n)\|)\}_{n=1}^7$, which distort the regression model at some extent. For this reason, we also report in Fig. 6.22 the regression model built upon the following set of $M_r = N(N-1)/2$ *predictive* training points

$$\mathcal{T}_N = \left\{ \left(\Delta_n^L(\boldsymbol{\mu}^i),\, \|E_n(\boldsymbol{\mu}^i)\|_{\mathcal{X}} \right), \quad n = 1, \dots, i-1, \quad i = 1, \dots, N \right\}. \tag{6.18}$$

Even though in this case it never underestimates the true error, the regression model is so poor in the untrained region $\Delta_N^L \lesssim 10^{-4}$ that it could possibly lead to the selection of snapshots that are already included during the greedy algorithm. Therefore, we recommend the use of the training set (5.57).

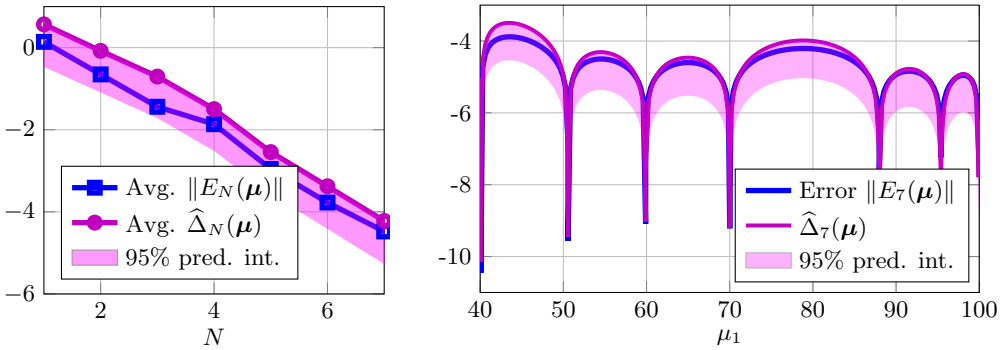


Fig. 6.21 Bypass problem, 1 parameter case. On the left: average (over a testing set of 80 random points in the parameter space) absolute error $\|E_N(\boldsymbol{\mu})\|$ and error indicator $\widehat{\Delta}_N(\boldsymbol{\mu})$ for $N = 1, \dots, 7$ (vertical axis in log scale). On the right: error and error indicator for $N = 7$ as functions of μ_1.

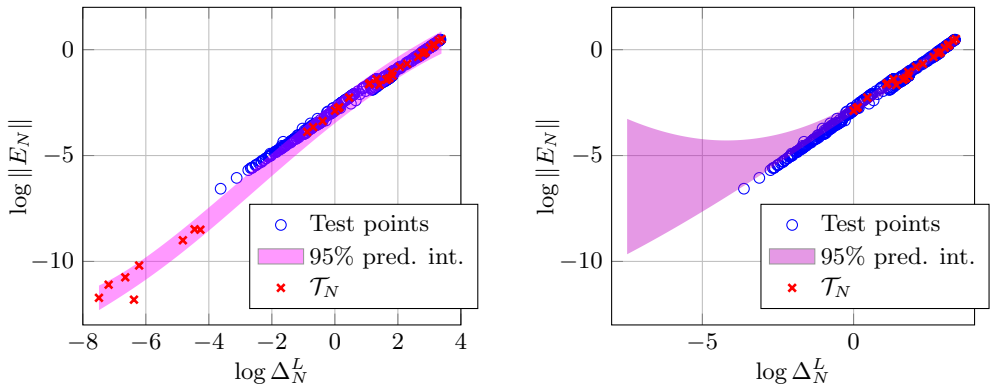
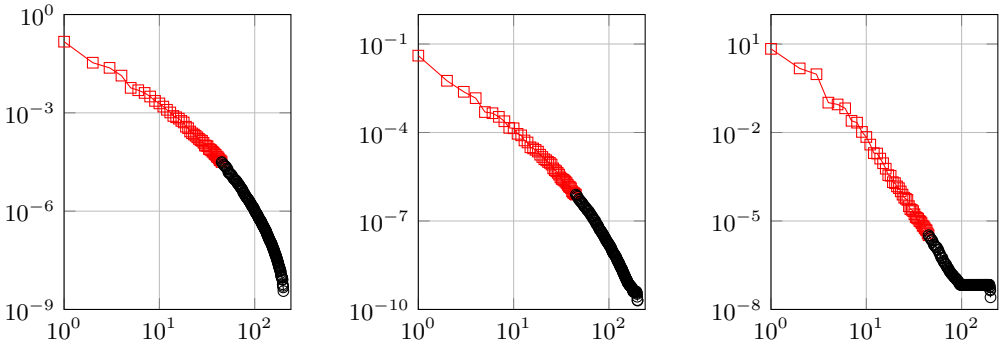


Fig. 6.22 Bypass problem, 1 parameter case. Comparison between the 95% prediction intervals obtained using $\mathcal{T}_N$ as defined in (5.57) (on the left) and the one defined in (6.18) (on the right). In both cases, the set of test points is the one generated from the the convergence analysis of Fig. 6.21 (left plot).

6.4.3 Parameter space exploration

We now let all the four parameters free to vary in $\mathcal{D}$. Due to the extent of the parameter space and the high number of terms in the affine decomposition, in this case we adopt a POD-based approach to build the ROM. We start by constructing the interpolant $\beta_I(\boldsymbol{\mu})$ of the stability factor by computing $\beta_h(\boldsymbol{\mu})$ in 40 interpolation points. Then, we solve the full-order model in correspondence of $N_s = 200$ parameter values selected by LHS sampling. We build the reduced spaces following the procedure detailed in Section 5.9.4 retaining $N = 45$ POD modes; the singular values of the snapshots matrix are shown in Fig. 6.23. Once the ROM is built, we compute the ingredients required for the evaluation of the dual norm of the residual. Finally, we generate the regression model (5.56) using as training points a random subset of $\mathcal{T}_N$ (as defined in (5.58)) of dimension 200. This latter and the resulting regression model are shown in Fig. 6.24, where we also report the convergence with respect to N of $\|E_N(\boldsymbol{\mu})\|_{\mathcal{X}}$ and $\widehat{\Delta}_N(\boldsymbol{\mu})$. In Fig. 6.25 we show the optimal state velocity obtained by solving the reduced optimization problem for different values of the parameters.



(a) state velocity (b) state pressure (c) control

Fig. 6.23 Bypass problem, 4 parameters case. Decay of the singular values (denoted by σ) of the state velocity, state pressure and control snapshot matrices. Red squares correspond to the retained modes, while black circles correspond to the discarded ones.

As regards the computational aspects, we notice a significant degradation of the online performances with respect to the previous case: the solution of the reduced optimization problem now takes on average 0.9 seconds. This is mainly due to the cost of assembling the reduced Jacobian matrix at each Newton iteration. Indeed, the assembly of $d\mathbf{G}_N(\cdot;\boldsymbol{\mu})$ requires to perform $5NQ_c$ additions of dense, square matrices of dimension $9N + 1$, yielding a computational complexity of $\mathcal{O}(Q_c N^3)$. As a result, at each Newton iteration, most of the time is spent assembling the reduced operator, and only a small fraction solving the linear system. A first remedy to reduce this computational bottleneck would be to exploit a suitable parallel implementation of the operator assembly. On top of this, a more intrusive approach would rely on the use of local reduced bases (see e.g. [EPR10, AF11]) in order to lower the N^3 leading term in the operations count.

As a matter of fact however, even relying on an inefficient implementation, the reduced model provides a speedup of at least one order of magnitude, which is expected to further increase as the size and complexity of the underlying full-order model increase.

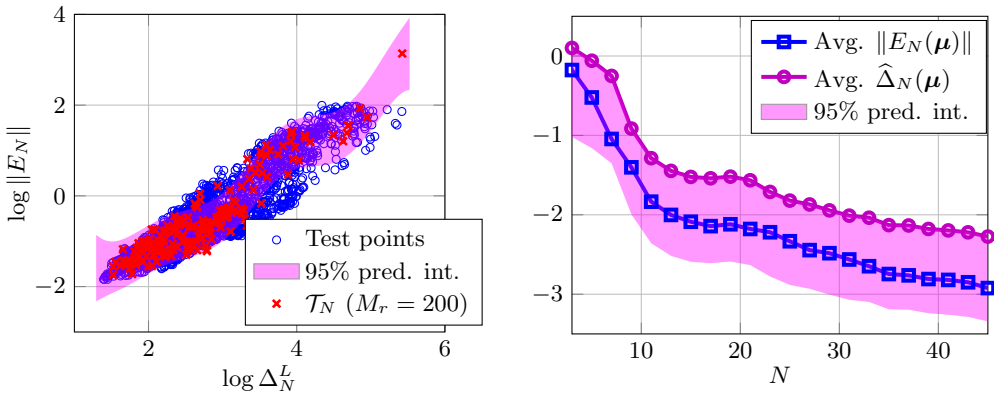


Fig. 6.24 Bypass problem, 4 parameters case. On the left: training set $\mathcal{T}_N$, 95% predictive intervals and test points. On the right: convergence of the relative error and error indicator averaged on a test sample of 150 random parameter values.

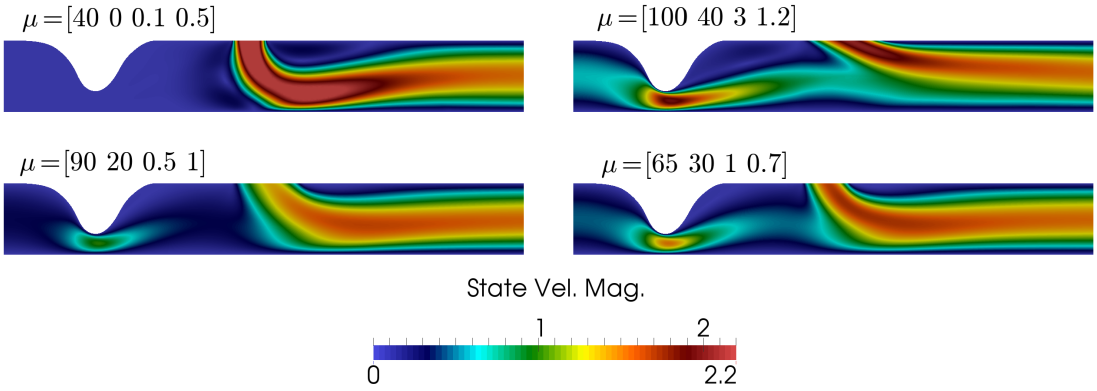


Fig. 6.25 Bypass problem. Optimal state velocity obtained by solving the ROM for different values of $\boldsymbol{\mu}$.

6.5 Vorticity minimization around a bluff body: the Navier-Stokes case

We consider again the (three-dimensional) vorticity minimization problem described in Sect. 6.3.2. This time, however, the fluid is modeled by the Navier-Stokes – rather than the Stokes – equations. The goal consists in minimizing the viscous energy dissipation in the wake of the body, by regulating the flow across a portion of its boundary Γ_C. Specifically, we minimize the following cost functional,

$$\mathcal{J}(\boldsymbol{v}, u; \boldsymbol{\mu}) = \frac{1}{2} \int_{\Omega_{\text{obs}}(\mu_1)} |\nabla \boldsymbol{v}|^2 \, d\Omega + \frac{1}{2\mu_3} \int_{\Gamma_C(\mu_1)} |\nabla_\Gamma u|^2 \, d\Gamma, \tag{6.19}$$

subject to the steady Navier-Stokes equations (5.89) together with the boundary conditions (6.12). In particular, we impose an horizontal constant velocity profile on the inflow boundary Γ_{in}, no-slip conditions on Γ_w, symmetry conditions on Γ_s, no-stress conditions on Γ_N and Dirichlet conditions on the control boundary Γ_C. See Fig. 6.26 for the details of the geometry.

The parameters are given by: the length $\mu_1 \in [0.1, 0.3]$ of the control boundary, the

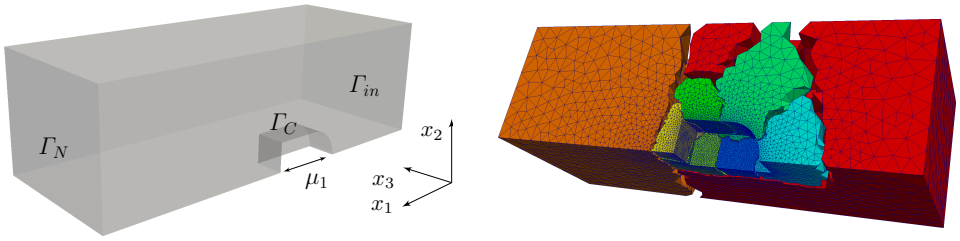


Fig. 6.26 On the left: domain and boundaries for problem (6.19). On the body boundary we impose a no-slip condition except that on the control region; on the top, bottom and lateral boundaries of the domain we impose symmetry conditions. On the right: computational mesh and its decomposition into 8 subdomains used for the preconditioner (6.20).

magnitude $\mu_2 \in [0.5, 3]$ of the inflow velocity profile and the inverse of the penalization factor $\mu_3 \in [1, 800]$. The kinematic viscosity ν is kept fixed at 0.03. We employ a decomposition of the geometry into three subdomains to obtain an affine decomposition with $Q_g = 27$ and $Q_d = 18$.

6.5.1 High-fidelity solver

Using a mesh made of 92 280 tetrahedral elements and 17 478 vertices, the total number of degrees of freedom is $N_h = 140\,441$; in particular, the dimension of the control space is $\dim(\mathcal{U}_h) = 617$. For any given $\boldsymbol{\mu}$, we solve the resulting nonlinear system employing the full-space Newton-Krylov-Schwarz solver proposed in [PBC06], i.e. we use the GMRES method with a two-level Additive Schwarz (AS) preconditioner to solve the linear system arising at each Newton step (5.19). To build the AS preconditioner for the Hessian matrix $\mathbf{K} = d\mathbf{G}(\cdot;\cdot)$, we first partition the domain Ω into overlapping subdomains Ω_j^δ, $j = 1 \dots, J$, featuring an overlap of size $\delta = h$ (see Fig. 6.26 on the right). We then build suitable restriction matrices $\mathrm{R}_j \in \mathbb{R}^{N_{h,j} \times N_h}$ so that the local matrices $\mathbf{K}_j = \mathrm{R}_j \mathbf{K} \mathrm{R}_j^T \in \mathbb{R}^{N_{h,j} \times N_{h,j}}$ correspond to the restriction of $\mathbf{K}$ to the subdomains Ω_m^δ. Finally we build a suitable coarse correction matrix $\mathbf{K}_0 = \mathrm{R}_0 \mathbf{K} \mathrm{R}_0^T \in \mathbb{R}^{n_v J \times n_v J}$ (n_v being the number of variables), whose restriction matrix $\mathrm{R}_0 \in \mathbb{R}^{n_v J \times N_h}$ is obtained by aggregation [Sal04]. The AS preconditioner is then defined as

$$\mathrm{P}_{AS}^{-1} = \sum_{j=0}^{M} \mathrm{R}_j^T \mathbf{K}_j^{-1} \mathrm{R}_j, \tag{6.20}$$

$\mathbf{K}_j^{-1}$ being the inverse of $\mathbf{K}_j$, here computed by means of an exact LU factorization; each local preconditioner is then applied in parallel.

Using 8 subdomains, the GMRES method converges on average in 50 iterations up to a tolerance of 10^{-8} on the relative norm of the residual. The outer Newton loop takes on average 7 iterations to reach a tolerance of 10^{-7} on the relative norm of the increment. Overall, solving the full-order problem for a given $\boldsymbol{\mu} \in \mathcal{D}$ takes about 5 minutes[6].

[6]Here, offline computations are performed on a node (with two Intel Xeon E5-2660 processors and 64 GB of RAM) of the SuperB cluster at EPFL. Online computations are performed on a workstation with a Intel Core i5-2400S processor and 16 GB of RAM.

6.5.2 Reduced-order approximation

We adopt a POD-based approach to build the ROM: we solve the full-order model in correspondence of $N_s = 100$ parameter values selected by LHS sampling and then retain $N = 30$ POD modes. Once the ROM is built, we compute the ingredients required for the evaluation of the dual norm of the residual. Finally, we build the regression model (5.56) using as training points a random subset of $\mathcal{T}_N$ (as defined in (5.58)) of dimension 200. This latter and the resulting regression model are shown in Fig. 6.27, where we also report the convergence of $\|E_N(\boldsymbol{\mu})\|_{\mathcal{X}}$ and $\widehat{\Delta}_N(\boldsymbol{\mu})$ with respect to N.

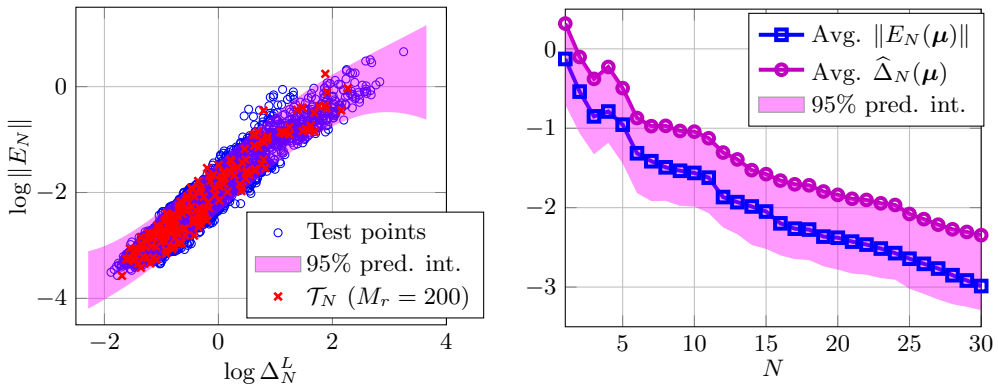


Fig. 6.27 Bluff body problem. On the left: training set $\mathcal{T}_N$, 95% predictive intervals and test points. On the right: convergence of the relative error and error indicator averaged on a test sample of 100 random parameter values.

Figure 6.28 shows the streamlines of the optimal state velocity around the body obtained by solving the reduced optimization problem for $\boldsymbol{\mu} = (0.15, 3, 1)$ and $\boldsymbol{\mu} = (0.15, 3, 800)$. The benefits of the optimization in reducing the vorticity are clearly visible, as the small vortices occurring for $\mu_3 = 1$ (which yields an almost uncontrolled velocity field) disappear for $\mu_3 = 800$. Moreover, solving the reduced problem takes only 0.5 seconds, leading to a speedup of about 600.

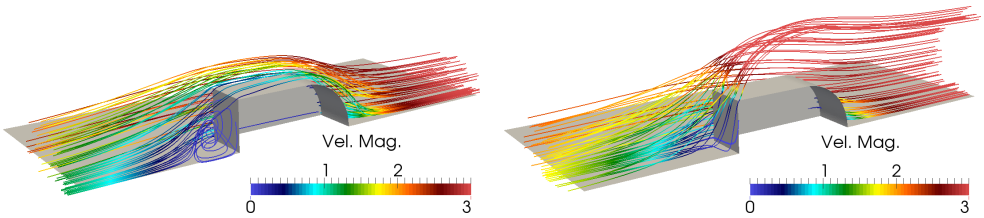


Fig. 6.28 Streamlines of the optimal state velocity around the bluff body obtained by solving the ROM for $\boldsymbol{\mu} = (0.15, 3, 1)$ (left) and $\boldsymbol{\mu} = (0.15, 3, 800)$ (right).

7 Conclusions

In this thesis, we have developed, analyzed and applied several model order reduction techniques for the simulation and optimization of parametrized PDEs.

In this context, a model order reduction approach successfully exploits three factors: the (offline) construction of an "optimal" reduced space, the use of fast and efficient algorithms for the (online) solution of the reduced problem, and the availability of a reliable method for judging the quality of the resulting reduced solution. The first and last factors determine the accuracy and reliability of the resulting ROM, while the second one directly affects its online efficiency. However, achieving this latter, which represents the main objective of model order reduction, requires to carefully take into account all these interdependent aspects.

From a computational standpoint, other key factors have to be considered in designing suitable reduction techniques. For instance, offline efficiency deserves particular attention, especially in case of limited computational resources or time constraints. In this respect, during offline computations, special care has to be devoted to properly balance the time spent computing the high-fidelity snapshots and that spent generating the ROM. Both are highly affected by the choice of the algorithms employed for the reduced space construction and the (possible) use of a posteriori error estimation. In particular, reliability can have a major impact on offline computational efficiency. Versatility and intrusiveness also play an important role in the development of reduction techniques. Indeed, projection-based model order reduction methods can be regarded as intrusive methods, since they usually require both accessing the high-fidelity model data-structures and ad-hoc implementations for the problem at hand. Sometimes, they even require to reformulate the discretization of the high-fidelity model or to modify its implementation. Therefore, the possibility to reuse high-fidelity legacy codes, or to apply the same technique to different types of problems independently of the underlying implementation, is particularly valuable.

The aim of this work was to develop suitable reduction strategies for the simulation of coupled fluid and mass transfer problems and the optimization of parametrized PDEs featuring high-dimensional control variables. To this end, we have developed some new reduction techniques and suitably adapted and combined existing ones, following the

general guidelines outlined above. The main methodological contributions of this thesis are summarized below.

- *Heuristic strategies for the approximation of stability factors*: after providing a suitable theoretical framework, we have first proposed a linearized version of the successive constraint method for quadratically nonlinear PDEs. Unfortunately, this method does not satisfy any of the general needs outlined above. Indeed, besides requiring an extensive offline computational effort, it fails to provide sufficiently sharp approximations of the stability factors, thus affecting the overall efficiency of the reduction process. Moreover, it is highly intrusive and specifically tailored on the class of problems at hand. For these reasons, we proposed an alternative approach based on adaptive radial basis interpolation, wherein accurate approximations of the stability factors can be obtained with a moderate computational effort. Moreover, this strategy is independent of the high-fidelity problem formulation, as it directly seeks an approximation of the stability factor. As such, it is completely non-intrusive with respect to the high-fidelity model implementation and suitable for linear and nonlinear, affine and nonaffine problems.

- *Hyper-reduction of parametrized systems by MDEIM*: we have developed a general framework to embed the MDEIM in the context of model reduction of parametric systems arising from the discretization of PDEs. Special emphasis was placed on the underlying offline-online computational strategy, based on a non-intrusive and efficient implementation relying on the reduced mesh concept. We demonstrated how the system approximation resulting from MDEIM can be combined with a state approximation resulting from a reduced-basis greedy approach as well as POD. Moreover, we derived a posteriori error estimates in the case of elliptic and parabolic systems, highlighting the contributions to the error from both the system and the state approximations.

- *A hyper-reduction strategy for the time-dependent Navier-Stokes equations*: we further extended the framework above by developing a new hyper-reduction strategy for the time-dependent Navier-Stokes equations. This strategy is tailored to the underlying high-fidelity approximation, which employs equal-order SUPG stabilized finite elements for the space discretization, a BDF time discretization and a semi-implicit treatment of the convective term. The reduced-order model was then generated by a Galerkin projection of the resulting fully-discrete problem onto a POD basis. A hybrid approach for the treatment of the nonlinear operators was proposed: we combined an exact quadratic expansion to reconstruct the convective term with an MDEIM approximation of the nonlinear SUPG terms.

- *A model order reduction framework for parametrized optimization problems*: the above-mentioned techniques have been exploited at last to develop a general model order reduction framework for parametrized optimization problems constrained by linear and nonlinear stationary PDEs. By adopting a full space formulation, the ROM was generated through a Galerkin projection of the high-fidelity optimality system onto a low-dimensional space. It was demonstrated how a simultaneous reduction of the state, control and adjoint spaces can be achieved by employing a greedy approach as well as POD. In both cases, the availability of a tight error

estimate is crucial to bound the offline computational costs and effectively quantify the accuracy of the reduced approximation in the online phase. To this end, we first derived a rigorous – yet too pessimistic in practice – error bound, which was then used to generate a much tighter error indicator. With this aim, we took advantage of the ROMES method proposing suitable strategies to embed this technique into greedy and POD-based offline basis construction algorithms.

All these methods have been instrumental to allow a very efficient treatment of several benchmark cases and other more complex problems relevant for a wide range of engineering and physical applications.

Besides the numerical tests dealing with the Navier-Stokes equations in Chap. 2, we have successfully applied our heuristic strategy for the approximation of stability factors to a nonaffinely parametrized Helmholtz problem, as well as to several PDE-constrained optimization problems. We note that the robustness of the method could be further improved by introducing more reliable error indicators, for instance based on cross-validation techniques [Bis06], to better quantify the interpolation accuracy.

The effectiveness of MDEIM on the complexity reduction of parametric problems was demonstrated for different scenarios. First, it was shown that MDEIM can be used for the fast solution of parametric optimization problems by applying it to the robust shape optimization of an acoustic horn. Then, the MDEIM was employed for the reduction of a coupled fluid-heat transfer problem past a cylinder. The latter served as a testbed for the applications presented in Chap. 4, dealing with coupled blood flow and mass transfer in the cardiovascular system. Given the complexity of this coupled problem – featuring time-dependency, nonlinearities and a nonaffine parametric dependence – the (still preliminary) results we obtained are very promising in terms of accuracy and efficiency. Moreover, as all these methods were designed with particular attention to non-intrusiveness and versatility, we are confident that they could be successfully applied to other challenging problems also in combination with different high-fidelity approximation techniques. In this respect, we would like to mention the successful application [Rin15] of this framework to a Kirchoff-Love shell model discretized by means of NURBS-based isogeometric analysis [HCB05]. Furthermore, the combination of these system approximation techniques with greedy-like training algorithms and adaptivity certainly deserves further investigation.

Finally, our reduction framework was applied to several linear and nonlinear, affine and nonaffine, high-dimensional optimization problems arising in fluid mechanics. Thanks to the monolithic approach that we have proposed, we were able to exploit many of the techniques previously discussed. Significant speedups of at least two order of magnitudes were obtained, yielding the possibility to solve the optimization problem in a small fraction of the time required by the solution of the underlying high-fidelity approximation. In the nonlinear case, the offline and online efficiency of the reduction process was significantly improved by the use of the ROMES method, where the rigorous yet over-conservative error bound was replaced by a much tighter error indicator. Our numerical results showed that the latter well serves for both adaptivity during the basis construction and online reliability. Following the lines of [PDTA14], this approach could thus be suitably exploited also to estimate the accuracy of the ROMs considered in Chap. 4, where the development and effective evaluation of rigorous error bounds is far more challenging and, often, out of reach.

Bibliography

[ABG+06] V. Akcelik, G. Biros, O. Ghattas, J. Hill, D. Keyes, and B. Waanders. Parallel algorithms for PDE-constrained optimization. In M.A. Heroux, P. Raghavan, and H.D. Simon, editors, *Parallel Processing for Scientific Computing*. SIAM, Philadelphia, 2006.

[ABH04] F. Abraham, M. Behr, and M. Heinkenschloss. The effect of stabilization in finite element methods for the optimal boundary control of the Oseen equations. *Finite Elem. Anal. Des.*, 41(3):229–251, 2004.

[ABK01] J.A. Atwell, J.T. Borggaard, and B.B. King. Reduced order controllers for Burgers' equation with a nonlinear observer. *Int. J. Appl. Math. Comput. Sci.*, 11(6):1311–1330, 2001.

[ACCF09] D. Amsallem, J. Cortial, K. Carlberg, and C. Farhat. A method for interpolating on manifolds structural dynamics reduced-order models. *Int. J. Numer. Methods Engrg.*, 80(9):1241–1258, 2009.

[ADGF13] D. Amsallem, S. Deolalikar, F. Gurrola, and C. Farhat. Model predictive control under coupled fluid-structure constraints using a database of reduced-order models on a tablet. *AIAA Paper 2013-2588, 21st AIAA Computational Fluid Dynamics Conference, San Diego*, pages 1–12, 2013.

[ADSS11] L. Azzimonti, M. Domanin, L. M. Sangalli, and P. Secchi. Surface estimation via spatial spline models with PDE penalization. In *Proceedings of S.Co.2011 Conference*, Padova, 2011.

[ADVN09] C. Audouze, F. De Vuyst, and P.B. Nair. Reduced-order modeling of parameterized PDEs using time-space parameter principal component analysis. *Int. J. Numer. Methods Engrg.*, 80(10):1025–1057, 2009.

[AF11] D. Amsallem and C. Farhat. An online method for interpolating linear parametric reduced-order models. *SIAM J. Sci. Comput.*, 33(5):2169–2198, 2011.

[AFS00] E. Arian, M. Fahl, and E.W. Sachs. Trust-region proper orthogonal decomposition for flow control. Technical Report Tech. report ICASE 2000-25, Institute for Computer Applications in Science and Engineering, Langley, VA, 2000.

[AHH+12] H. Antil, M. Heinkenschloss, R. W. Hoppe, C. Linsenmann, and A. Wixforth. Reduced order modeling based shape optimization of surface acoustic wave driven microfluidic biochips. *Math. Comput. Simul.*, 82(10):1986 – 2003, 2012.

[AHS14] H. Antil, M. Heinkenschloss, and D. C. Sorensen. Application of the discrete empirical interpolation method to reduced order modeling of nonlinear and parametric systems. In A. Quarteroni and G. Rozza, editors, *Reduced Order Methods for Modeling and Computational Reduction*, volume 9 of *Modeling, Simulation and Applications*, pages 101–136. Springer, Switzerland, 2014.

[ANR09] I. Akhtar, A. H. Nayfeh, and C. J. Ribbens. On the stability and extension of reduced-order Galerkin models in incompressible flows. *Theor. Comput. Fluid Dyn.*, 23(3):213–237, 2009.

[ANSS14] L. Azzimonti, F. Nobile, L.M. Sangalli, and S. Secchi. Mixed finite elements for spatial regression with PDE penalization. *SIAM/ASA J. Uncert. Quant.*, 2(1):305–335, 2014.

[Ant05] A.C. Antoulas. *Approximation of Large-Scale Dynamical Systems*. SIAM, Philadelphia, 2005.

[APB+08] L. Antiga, M. Piccinelli, L. Botti, B. Ene-Iordache, A. Remuzzi, and D. Steinman. An image-based modeling framework for patient-specific computational hemodynamics. *Med. Biol. Eng. Comput.*, 46(11):1097–1112, 2008.

[AT12] Aneurisk-Team. AneuriskWeb project website, URL: http://ecm2.mathcs.emory.edu/aneuriskweb. Web Site, 2012.

[AV14] A. Alla and S. Volkwein. Asymptotic stability of POD based model predictive control for a semilinear parabolic pde. *Adv. Comput. Math.*, pages 1–30, 2014.

[AWWB08] P. Astrid, S. Weiland, K. Willcox, and T. Backx. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Trans. Autom. Control*, 53(10):2237–2251, 2008.

[AZCF14] D. Amsallem, M. J. Zahr, Y. Choi, and C. Farhat. Design optimization using hyper-reduced-order models. *Struct. Multidisc. Optim.*, 51(4):919–940, 2014.

[Bal15] F. Ballarin. *Reduced-Order Models for Patient-Specific Haemodynamics of Coronary Artery Bypass Grafts*. PhD thesis, Politecnico di Milano, Milano, 2015.

[BB01] R. Becker and M. Braack. A finite element pressure gradient stabilization for the Stokes equations based on local projections. *Calcolo*, 38(4):173–199, 2001.

[BBD04] G. Biswas, M. Breuer, and F. Durst. Backward-facing step flows for various expansion ratios at low and moderate Reynolds numbers. *J. Fluids Eng.*, 126(3):362–374, 2004.

[BBF13] D. Boffi, F. Brezzi, and M. Fortin. *Mixed Finite Elements and Applications*. Springer-Verlag, Berlin-Heidelberg, 2013.

[BBI09] M. Bergmann, C.H. Bruneau, and A. Iollo. Enablers for robust POD models. *J. Comp. Phys.*, 228(2):516–538, 2009.

[BC08] M. Bergmann and L. Cordier. Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reduced-order models. *J. Comput. Phys.*, 227(16):7813–7840, 2008.

[BC10] A.T. Barker and X.C. Cai. Scalable parallel methods for monolithic coupling in fluid-structure interaction with application to blood flow modeling. *J. Comput. Phys.*, 229(3):642–659, 2010.

[BCC+07] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Comput. Methods Appl. Mech. Engrg.*, 197(1-4):173 – 201, 2007.

[BCI13] J. Baiges, R. Codina, and S. Idelsohn. Explicit reduced-order models for the stabilized finite element approximation of the incompressible Navier–Stokes equations. *Int. J. Numer. Methods Fluids*, 72(12):1219–1243, 2013.

[BCI15] J. Baiges, R. Codina, and S. Idelsohn. Reduced-order subscales for POD models. *Comput. Methods Appl. Mech. Engrg.*, 291(0):173–196, 2015.

[BCM99] R.K. Beatson, J.B. Cherrie, and C.T. Mouat. Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration. *Adv. Comput. Math.*, 11(2-3):253–270, 1999.

[BCP96] K. Eleda Brenan, S.L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. SIAM, Philadelphia, 1996.

[BCTH07] Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. Yzβ discontinuity capturing for advection-dominated processes with application to arterial drug delivery. *Int. J. Numer. Methods Fluids*, 54(6-8):593–608, 2007.

[BDG06] P.B. Bochev, C.R. Dohrmann, and M.D. Gunzburger. Stabilization of low-order mixed finite elements for the Stokes equations. *SIAM J. Numer. Anal.*, 44(1):82–101, 2006.

[BF08] E. Burman and M.A. Fernández. Galerkin finite element methods with symmetric pressure stabilization for the transient Stokes equations: stability and convergence analysis. *SIAM J. Numer. Anal.*, 47(1):409–439, 2008.

[BG82] D.G. Buerk and T.K. Goldstick. Arterial wall oxygen consumption rate varies spatially. *Am. J. Physiol. Heart Circ. Physiol.*, 243(6):H948–H958, 1982.

[BG04] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numer.*, 13:147–269, 2004.

[BG05a] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver. *SIAM J. Sci. Comput.*, 27(2):687–713, 2005.

[BG05b] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part II: The Lagrange-Newton solver and its application to optimal control of steady vscous flows. *SIAM J. Sci. Comput.*, 27(2):714–739, 2005.

[BGL05] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numer.*, 14(1):1–137, 2005.

[BGL06] J. Burkardt, M. Gunzburger, and H.C. Lee. POD and CVT-based reduced-order modeling of Navier-Stokes flows. *Comput. Meth. Appl. Mech. Engrg.*, 196(1-3):337–355, 2006.

[BGW15] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *To appear in SIAM Review*, 2015.

[BHZ+10] Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, X. Liang, T. Kvamsdal, R. Brekken, and J.G. Isaksen. A fully-coupled fluid-structure interaction simulation of cerebral aneurysms. *Comput. Mech.*, 46(1):3–16, 2010.

[Bis06] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, New York, 2006.

[BK05] A. Borzì and K. Kunisch. A multigrid scheme for elliptic constrained optimal control problems. *Comput. Optim. Appl.*, 31:309–333, 2005.

[BKR00] R. Becker, H. Kapp, and R. Rannacher. Adaptive finite element methods for optimal control of partial differential equations: Basic concept. *SIAM J. Control Optim.*, 39:113–132, 2000.

[BME03] P. Benner, V. Mehrmann, and D.C. Sorensen (Eds.). *Dimension Reduction of Large-Scale Systems*. Lecture Notes in Computational Science and Engineering. Springer, Heildeberg, 2003.

[BMNP04] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Paris. Sér. I Math.*, 339(9):667 – 672, 2004.

[BMS14] M. Bebendorf, Y. Maday, and B. Stamm. Comparison of some reduced representation approximations. In A. Quarteroni and G. Rozza, editors, *Reduced Order Methods for Modeling and Computational Reduction*, volume 9 of *Modeling, Simulation and Applications (MS&A)*, pages 67–100. Springer International Publishing, Switzerland, 2014.

[BNB03] E. Bangtsson, D. Noreland, and M. Berggren. Shape optimization of an acoustic horn. *Comput. Methods Appl. Mech. Engrg.*, 192(11–12):1533 – 1571, 2003.

[BP84] F. Brezzi and J. Pitkäranta. On the stabilization of finite element approximations of the Stokes equations. In W. Hackbusch, editor, *Efficient Solutions of Elliptic Systems*, volume 10 of *Notes on Numerical Fluid Mechanics*, pages 11–19. Vieweg+Teubner Verlag, 1984.

[Bra09] M. Braack. Optimal control in fluid mechanics by finite elements with symmetric stabilization. *SIAM J. Control Optim.*, 48(2):672–687, 2009.

[Bre74] F. Brezzi. On the existence, uniqueness, and approximation of saddle point problems arising from Lagrangian multipliers. *R.A.I.R.O., Anal. Numér.*, 2:129–151, 1974.

[BRR80] F. Brezzi, J. Rappaz, and P.A. Raviart. Finite dimensional approximation of nonlinear problems. Part I: Branches of nonsingular solutions. *Numer. Math.*, 36:1–25, 1980.

[BS09] A. Borzì and V. Schulz. Multigrid methods for PDE optimization. *SIAM Review*, 51(2):361–395, 2009.

[BS11] A. Borzì and V. Schulz. *Computational Optimization of Systems Governed by Partial Differential Equations*. SIAM, Philadelphia, 2011.

[BSN06] J.L. Brisman, J.K. Song, and D.W. Newell. Cerebral aneurysms. *N. Engl. J. Med.*, 355(9):928–939, 2006.

[BSV14] P. Benner, E. Sachs, and S. Volkwein. Model order reduction for PDE constrained optimization. In G. Leugering, P. Benner, S. Engell, A. Griewank, H. Harbrecht, M. Hinze, R. Rannacher, and S. Ulbrich, editors, *Trends in PDE Constrained Optimization*, volume 165 of *International Series of Numerical Mathematics*, pages 303–326. Springer International Publishing, 2014.

[BTDW03] T. Bui-Thanh, M. Damodaran, and K. Willcox. Proper orthogonal decomposition extensions for parametric applications in transonic aerodynamics. In *Proceedings of the 15th AIAA Computational Fluid Dynamics Conference (AIAA Paper 2003-4213)*, 2003.

[BTDW04] T. Bui-Thanh, M. Damodaran, and K. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA J.*, 42(8):1505–1516, 2004.

[BTWG08] T. Bui-Thanh, K. Willcox, and O. Ghattas. Parametric reduced-order models for probabilistic analysis of unsteady aerodynamics applications. *AIAA J.*, 46(10):2520–2529, 2008.

[Buh03] M. D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge University Press, Cambridge, 2003.

[BWP+15] P.J. Blanco, S.M. Watanabe, M.A. Passos, P. Lemos, and R.A. Feijoo. An anatomically detailed arterial network model for one-dimensional computational hemodynamics. *IEEE Trans. Biomed. Eng*, 62(2):736–753, 2015.

[Car11] K. Carlberg. *Model Reduction of Nonlinear Mechanical Systems via Optimal Projection and Tensor Approximation*. PhD thesis, Stanford University, Stanford, 2011.

[CBBH08] V.M. Calo, N.F. Brasher, Y. Bazilevs, and T.J.R. Hughes. Multiphysics model for blood flow and drug transport with application to patient-specific coronary artery flow. *Comput. Mech.*, 43(1):161–177, 2008.

[CBMF11] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. *Int. J. Numer. Methods Engrg.*, 86(2):155–181, 2011.

[CBS99] E.A. Christensen, M. Brøns, and J.N. Sørensen. Evaluation of proper orthogonal decomposition–based decomposition techniques applied to parameter-dependent nonturbulent flows. *SIAM J. Sci. Comput.*, 21:1419–1434, 1999.

[CC08] G. Coppola and C. Caro. Oxygen mass transfer in a model three-dimensional artery. *J. R. Soc. Interface*, 5(26):1067–1075, 2008.

[CC09] G. Coppola and C. Caro. Arterial geometry, flow pattern, wall shear and mass transport: potential physiological significance. *J. R. Soc. Interface*, 6(35):519–528, 2009.

[CCC+13] M. Caputo, C. Chiastra, C. Cianciolo, E. Cutrì, G. Dubini, J. Gunn, B. Keller, F. Migliavacca, and P. Zunino. Simulation of oxygen transfer in stented arteries and correlation with in-stent restenosis. *Int. J. Numer. Meth. Biomed. Eng.*, 29(12):1373–1387, 2013.

[CFCA13] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *J. Comput. Phys.*, 242(0):623 – 647, 2013.

[CH12] S.S. Collis and M. Heinkenschloss. Analysis of the streamline upwind/Petrov Galerkin method applied to the solution of optimal control problems. tech. Report CAAM TR02-01, Rice University, 2012.

[CHMR08] Y. Chen, J. Hesthaven, Y. Maday, and J. Rodriguez. A monotonic evaluation of lower bounds for inf-sup stability constants in the frame of reduced basis approximations. *C. R. Acad. Sci. Paris, Ser. I*, 346:1295–1300, 2008.

[CHMR09] Y. Chen, J. Hesthaven, Y. Maday, and J. Rodriguez. Improved successive constraint method based a posteriori error estimate for reduced basis approximation of 2D Maxwell's problem. *ESAIM Math. Modelling Numer. Anal.*, 43:1099–1116, 2009.

[CIJS14] A. Caiazzo, T. Iliescu, V. John, and S. Schyschlowa. A numerical investigation of velocity-pressure reduced order models for incompressible flows. *J. Comput. Phys.*, 259:598–616, 2014.

[CMW15] T. Cui, Y. Marzouk, and K. Willcox. Data-driven model reduction for the Bayesian solution of inverse problems. *Int. J. Numer. Methods Engrg.*, 102(5):966–990, 2015.

[CN12] M. Chevreuil and A. Nouy. Model order reduction based on proper generalized decomposition for the propagation of uncertainties in structural dynamics. *Int. J. Numer. Meth. Engng*, 89(2):241–268, 2012.

[Coc07] W. G. Cochran. *Sampling techniques*. John Wiley & Sons, Chichester, 2007.

[Col14] C. Colciago. *Reduced Order Fluid-Structure Interaction Models for Haemodynamics Applications*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Lausanne, 2014.

[CR97] G. Caloz and J. Rappaz. Numerical analysis for nonlinear and bifurcation problems. In P.G. Ciarlet and J.L. Lions, editors, *Handbook of Numerical Analysis, Vol. V*, Techniques of Scientific Computing (Part 2), pages 487–637. Elsevier Science B.V., 1997.

[CS10] S. Chaturantabut and D.C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.*, 32(5):2737–2764, 2010.

[cSN15] R. Ştefănescu, A. Sandu, and I.M. Navon. POD/DEIM reduced-order strategies for efficient four dimensional variational data assimilation. *J. Comput. Phys.*, 295(295):569–595, 2015.

[CST00] K.A. Cliffe, A. Spence, and S.J. Tavener. The numerical analysis of bifurcation problems with application to fluid mechanics. *Acta Numer.*, 9(00):39–131, 2000.

[CTB12] K. Carlberg, R. Tuminaro, and P. Boggs. Efficient structure-preserving model reduction for nonlinear mechanical systems with application to structural dynamics. *AIAA Paper 2012-1969, 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, Hawaii*, 2012.

[CTB15] K. Carlberg, R. Tuminaro, and P. Boggs. Preserving Lagrangian structure in nonlinear model reduction with application to structural dynamics. *SIAM J. Sci. Comput.*, 37(2):B153–B184, 2015.

[CTU09] C. Canuto, T. Tonn, and K. Urban. A posteriori error analysis of the reduced basis method for non-affine parameterized nonlinear PDEs. *SIAM J. Numer. Anal.*, 47(3):2001–2022, 2009.

[DB04] C.R. Dohrmann and P.B. Bochev. A stabilized finite element method for the Stokes problem based on polynomial pressure projections. *Int. J. Numer. Methods Fluids*, 46(2):183–201, 2004.

[DC15] M. Drohmann and K. Carlberg. The ROMES method for statistical modeling of reduced-order-model error. *SIAM/ASA J. Uncert. Quant.*, 3(1):116–145, 2015.

[Ded07] L. Dedè. Optimal flow control for Navier-Stokes equations: Drag minimization. *Int. J. Numer. Meth. Fluids*, 55(4):347 – 366, 2007.

[Ded10] L. Dedè. Reduced basis method and a posteriori error estimation for parametrized linear-quadratic optimal control problems. *SIAM J. Sci. Comput.*, 32:997–1019, 2010.

[Ded12] R.J. Dedden. Model order reduction using the discrete empirical interpolation method. Master's thesis, TU Delft, 2012.

[Dep08] S. Deparis. Reduced basis error bound computation of parameter-dependent Navier-Stokes equations by the natural norm approach. *SIAM J. Num. Anal.*, 46(4):2039–2067, 2008.

[DFQ14] S. Deparis, D. Forti, and A. Quarteroni. A rescaled localized radial basis function interpolation on non-cartesian and nonconforming grids. *SIAM J. Sci. Comput.*, 36(6):A2745–A2762, 2014.

[DH13] M. Dihlmann and B. Haasdonk. Certified nonlinear parameter optimization with reduced basis surrogate models. *Proc. Appl. Math. Mech*, 13(1):3–6, 2013.

[DHO12] M. Drohmann, B. Haasdonk, and M. Ohlberger. Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM J. Sci. Comput.*, 34(2):A937–A969, 2012.

[DHV98] J.E. Dennis, M. Heinkenschloss, and L.N. Vicente. Trust-region interior-point SQP algorithms for a class of nonlinear programming problems. *SIAM J. Control Optim.*, 36(5):1750–1794, 1998.

[DI94] M. Desai and K. Ito. Optimal controls of Navier-Stokes equations. *SIAM J. Control Optim.*, 32(5):1428–1446, 1994.

[Dri97] D. Drikakis. Bifurcation phenomena in incompressible sudden expansion flows. *Phys. Fluids*, 9(1):76–87, 1997.

[DVW10] J. Degroote, J. Vierendeels, and K. Willcox. Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis. *Int. J. Numer. Methods Fluids*, 63(2):207–230, 2010.

[DZ11] M.C. Delfour and J.-P. Zolésio. *Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization*. SIAM, Philadelphia, 2011.

[EPR10] J. Eftang, A. Patera, and E. Rønquist. An *hp* certified reduced basis method for parametrized elliptic partial differential equations. *SIAM J. Sci. Comput.*, 32(6):3170–3200, 2010.

[ES95] R. Everson and L. Sirovich. Karhunen–Loeve procedure for gappy data. *J. Opt. Soc. Am. A*, 12(8):1657–1664, 1995.

[ESW04] H.C. Elman, D.J. Silvester, and A.J. Wathen. *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*. Oxford University Press, New York, 2004.

[FD15] D. Forti and L. Dede. Semi-implicit BDF time discretization of the Navier-Stokes equations with VMS-LES modeling in a high performance computing framework. *Comput. Fluids*, 117(0):168–182, 2015.

[FR09] C. Fisher and J.S. Rossmann. Effect of non-Newtonian behavior on hemodynamics of cerebral aneurysms. *J. Biomech. eng.*, 131(9):091004, 2009.

[FZ07] B. Fornberg and J. Zuev. The Runge phenomenon and spatially variable shape parameters in RBF interpolation. *Comput. Math. Appl.*, 54(3):379–398, 2007.

[GA04] S. Gugercin and A.C. Antoulas. A survey of model reduction by balanced truncation and some new results. *Int. J. Control*, 77(8):748–766, 2004.

[GB97] O. Ghattas and J.H. Bark. Optimal control of two- and three-dimensional incompressible Navier-Stokes flows. *J. Comput. Phys.*, 136(2):231–244, 1997.

[GB04] M. D. Gunzburger and P. B. Bochev. Finite element methods for optimization and control problems for the Stokes equations. *Comp. Math. Appl.*, 48:1035–1057, 2004.

[GB09] M. D. Gunzburger and P. B. Bochev. *Least-Squares Finite Element Methods*. Springer-Verlag, New York, 2009.

[GFWG10] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas. Nonlinear model reduction for uncertainty quantification in large-scale inverse problems. *Int. J. Numer. Methods Engrg.*, 81(12):1581–1608, 2010.

[GG98] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numer. algorithms*, 18(3-4):209–232, 1998.

[GHS91] M. D. Gunzburger, L. S. Hou, and Th. P. Svobodny. Analysis and finite element approximation of optimal control problems for the stationary Navier-Stokes equations with Dirichlet controls. *ESAIM Math. Modelling Numer. Anal.*, 25(6):711–748, 1991.

[GHS93] M.D. Gunzburger, L. Hou, and T.P. Svobodny. Optimal control and optimization of viscous, incompressible flows. In Gunzburger. M.D. and R.A. Nicolaides, editors, *Incompressible Computational Fluid Dynamics*, pages 109–150. Cambridge University Press, 1993.

[GK11] M.A. Grepl and M. Kärcher. Reduced basis a posteriori error bounds for parametrized linear-quadratic elliptic optimal control problems. *C. R. Math. Acad. Sci. Paris*, 349(15-16):873 – 877, 2011.

[GLLS97] M. Giles, M. Larson, M. Levenstam, and E. Suli. Adaptive error control for finite element approximations of the lift and drag coefficients in viscous flow. Technical report, Oxford University Computing Laboratory, 1997.

[GMNP07] M. Grepl, Y. Maday, N.C. Nguyen, and A.T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM Math. Modelling Numer. Anal.*, 41(3):575–605, 2007.

[GNV+07] M. Grepl, N.C. Nguyen, K. Veroy, A.T. Patera, and G.R. Liu. Certified rapid solution of partial differential equations for real-time parameter estimation and optimization. In L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. Van Bloemen Waanders, editors, *Real-time PDE-Constrained Optimization*, pages 197–215. SIAM, Philadelphia, 2007.

[GP05] M. Grepl and A.T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM Math. Modelling Numer. Anal.*, 39(1):157–181, 2005.

[GR86] V. Girault and P.-A. Raviart. *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms.* Springer-Verlag, Berlin and New York, 1986.

[GR09] C. Geuzaine and J.-F. Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *Int. J. Numer. Methods Engrg.*, 79(11):1309–1331, 2009.

[Gre12] M. Grepl. Certified reduced basis methods for nonaffine linear time-varying and nonlinear parabolic partial differential equations. *Math. Mod. and Meth. in Appl. Sc.*, 22(3):1150015, 2012.

[GSV06] P. Gervasio, F. Saleri, and A. Veneziani. Algebraic fractional-step schemes with spectral methods for the incompressible Navier–Stokes equations. *J. Comput. Phys.*, 214(1):347–365, 2006.

[GT02] J. R. Gilbert and S.-H. Teng. MESHPART, a Matlab Mesh Partitioning and Graph Separator Toolbox, 2002.

[GU14] J. Ghiglieri and S. Ulbrich. Optimal flow control based on POD and MPC and an application to the cancellation of Tollmien-Schlichting waves. *Optim. Methods Softw.*, 29(5):1042–1074, 2014.

[Gun03] M.D. Gunzburger. *Perspectives in Flow Control and Optimization*. SIAM, Philadelphia, 2003.

[GV12] A. Gerner and K. Veroy. Certified reduced basis methods for parametrized saddle point problems. *SIAM J. Sci. Comput.*, 34(5):A2812–A2836, 2012.

[GVL13] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, fourth edition, 2013.

[HCB05] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Meth. Appl. Mech. Engrg.*, 194(39):4135–4195, 2005.

[HDO11] B. Haasdonk, M. Dihlmann, and M. Ohlberger. A training set and multiple bases generation approach for parametrized model reduction based on adaptive grids in parameter space. *Math. Comput. Model. Dynam. Syst.*, 17(4):423–442, 2011.

[Hei98] M. Heinkenschloss. Formulation and analysis of a sequential quadratic programming method for the optimal dirichlet boundary control of Navier-Stokes flow. In *Optimal Control: Theory, Algorithms, and Applications*, pages 178–203. Kluwer Academic Publishers B.V., 1998.

[HG27] T. H. Hildebrandt and Lawrence M. Graves. Implicit functions and their differentials in general analysis. *Trans. Amer. Math. Soc.*, 29(1):127–153, 1927.

[HGT10] E.M.T. Hendrix and B. G.-Tóth. *Introduction to Nonlinear and Global Optimization*. Springer, New York, 2010.

[HHB+12] S.S. Hossain, S.F.A. Hossainy, Y. Bazilevs, V.M. Calo, and T.J.R. Hughes. Mathematical modeling of coupled drug and drug-encapsulated nanoparticle transport in patient-specific coronary artery walls. *Comput. Mech.*, 49(2):213–242, 2012.

[HHD14] S.S. Hossain, T.J.R. Hughes, and P. Decuzzi. Vascular deposition patterns for nanoparticles in an inflamed patient-specific arterial tree. *Biomech. Model. Mechanobiol.*, 13(3):585–597, 2014.

[HKC+10] D.B.P. Huynh, D.J. Knezevic, Y. Chen, J.S. Hesthaven, and A.T. Patera. A natural-norm successive constraint method for inf-sup lower bounds. *Comput. Meth. Appl. Mech. Engrg.*, 199(29–32):1963–1975, 2010.

[HLB96] P. Holmes, J. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, Cambridge, 1996.

[HMP13] H. Herrero, Y. Maday, and F. Pla. RB (reduced basis) for RB (Rayleigh-Bénard). *Comput. Meth. Appl. Mech. Engrg.*, 261–262:132–141, 2013.

[HNPR12] D.B.P. Huynh, N.C. Nguyen, A.T. Patera, and G. Rozza. Rapid reliable solution of the parametrized partial differential equations of continuum mechanics and transport, 2008-2012. URL: http://augustine.mit.edu.

[HO08] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM Math. Model. Numer. Anal.*, 42:277–302, 2008.

[HO09] B. Haasdonk and M. Ohlberger. Efficient reduced models for parametrized dynamical systems by offline/online decomposition. In *Proc. MATHMOD 2009, 6th Vienna International Conference on Mathematical Modelling*, 2009.

[HPUU09] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE Constraints*. Springer, Netherlands, 2009.

[HR99] L. S. Hou and S. S. Ravindran. Numerical approximation of optimal flow control problems by a penalty method: Error estimates and numerical results. *SIAM J. Sci. Comput.*, 20(5):1753–1777, 1999.

[HRSP07] D.B.P Huynh, G. Rozza, S. Sen, and A.T. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability costants. *C. R. Acad. Sci. Paris. Sér. I Math.*, 345:473–478, 2007.

[HSF04] T.J.R. Hughes, G. Scovazzi, and L.P. Franca. *Multiscale and stabilized methods.* Wiley Online Library, 2004.

[HSZ14] J.S. Hesthaven, B. Stamm, and S. Zhang. Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods. *ESAIM Math. Modelling Numer. Anal.*, 48:259–283, 2014.

[HV05] M. Hinze and S. Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. In P. Benner, D.C. Sorensen, and V. Mehrmann, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational Science and Engineering*, pages 261–306. Springer Berlin Heidelberg, 2005.

[HV08] M. Hinze and S. Volkwein. Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition. *Comput. Optim. Appl.*, 39:319–345, 2008.

[IK08] K. Ito and K. Kunisch. *Lagrange Multiplier Approach to Variational Problems and Applications.* SIAM, Philadelphia, 2008.

[IR98a] K. Ito and S. Ravindran. A reduced basis method for control problems governed by PDEs. In F. Kappel W. Desch and K. Kunisch, editors, *Control and Estimation of Distributed Parameter System*, pages 153–168. Birkhäuser, Basel, 1998.

[IR98b] K. Ito and S.S. Ravindran. A reduced order method for simulation and control of fluid flows. *J. Comput. Phys.*, 143(2):403–425, 1998.

[IW14] T. Iliescu and Z. Wang. Variational multiscale proper orthogonal decomposition: Navier-stokes equations. *Numer. Meth. Part. D. E.*, 30(2):641–663, 2014.

[KG14a] M. Kärcher and M.A. Grepl. A certified reduced basis method for parametrized elliptic optimal control problems. *ESAIM Control Optim. Calc. Var.*, 20(2):416–441, 2014.

[KG14b] M. Kärcher and M.A. Grepl. A posteriori error estimation for reduced order solutions of parametrized parabolic optimal control problems. *ESAIM: Math. Model. Numer. Anal.*, 48(6):1615–1638, 2014.

[KK09] G. Karypis and V. Kumar. MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0. URL: http://www.cs.umn.edu/~metis, 2009.

[KP99] G. Karner and K. Perktold. Numerical modeling of mass transport in the arterial wall. In V.K. Goal, R.L. Spilker, G.A. Ateshian, and L.J. Solawsky, editors, *Proceedings of the 1999 Bioengineering Conference, BED*, volume 42, page 739–740, New York, 1999.

[KSZ13] W. Krendl, V. Simoncini, and W. Zulehner. Stability estimates and structural spectral properties of saddle point problems. *Numer. Math.*, 124(1), 2013.

[KTV13] Kammann, E., Tröltzsch, F., and Volkwein, S. A posteriori error estimation for semilinear parabolic optimal control problems with application to model reduction by POD. *ESAIM Math. Model. Numer. Anal.*, 47(2):555–581, 2013.

[KV03] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.*, 40(2):492–515, 2003.

[KV07] M. Kahlbacher and S. Volkwein. Galerkin proper orthogonal decomposition methods for parameter dependent elliptic systems. *Discuss. Math., Differ. Incl. Control Optim.*, 27:95–117, 2007.

[KV12] M. Kahlbacher and S. Volkwein. POD a-posteriori error based inexact SQP method for bilinear elliptic optimal control problems. *ESAIM: Math. Model. Numer. Anal.*, 46(02):491–511, 2012.

[KVX04] K. Kunisch, S. Volkwein, and L. Xie. HJB-POD-based feedback design for the optimal control of evolution problems. *SIAM J. Appl. Dyn. Syst.*, 3(4):701–722, 2004.

[KWB12] F. Kasolis, E. Wadbro, and M. Berggren. Fixed-mesh curvature-parameterized shape optimization of an acoustic horn. *Struct. Multidiscip. Optim.*, 46(5), 2012.

[Lev95] M.J. Lever. Mass transport through the walls of arteries and veins. In M.Y. Jaffrin and C.G. Caro, editors, *Biological Flows*, pages 177–197. Springer, New York, 1995.

[LMQR13] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. Boundary control and shape optimization for the robust design of bypass anastomoses under uncertainty. *ESAIM Math. Model. Numer. Anal.*, 47:1107–1131, 2013.

[LMR12] T. Lassila, A. Manzoni, and G. Rozza. On the approximation of stability factors for general parametrized partial differential equations with a two-level affine decomposition. *ESAIM Math. Modelling Numer. Anal.*, 46:1555–1576, 2012.

[Loh10] S. L. Lohr. *Sampling: Design and Analysis*. Cengage Learning, Boston, second edition, 2010.

[LR10] T. Lassila and G. Rozza. Parametric free-form shape design with PDE models and reduced basis method. *Comput. Methods Appl. Mech. Engrg.*, 199(23–24):1583–1592, 2010.

[LR11] T. Lassila and G. Rozza. Model reduction of semiaffinely parametrized partial differential equations by two-level affine approximation. *C.R. Math. Acad. Sci. Paris, Series I*, 349(1–2):61–66, 2011.

[Lum67] J. Lumley. The structure of inhomogeneous turbulent flows. In *Atmospheric Turbulence and Radio Wave Propagation*, pages 166–178. Moscow: Nauka, 1967.

[LWG10] C. Lieberman, K. Willcox, and O. Ghattas. Parameter and state model reduction for large-scale statistical inverse problems. *SIAM J. Sci. Comput.*, 32(5):2523–2542, 2010.

[MA10] T. J. Mackman and C. B. Allen. Investigation of an adaptive sampling method for data interpolation using radial basis functions. *Int. J. Numer. Methods Engrg.*, 83(7):915–938, 2010.

[Man12] A. Manzoni. *Reduced models for optimal control, shape optimization and inverse problems in haemodynamics*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Lausanne, 2012.

[Man14] A. Manzoni. An efficient computational framework for reduced basis approximation and a posteriori error estimation of parametrized Navier-Stokes flows. *ESAIM: Math. Model. Numer. Anal.*, 48(4):1199–1226, 2014.

[Mat] Matlab®. The MathWorks. URL: http://www.mathworks.com.

[MCG+12] E. Marchandise, P. Crosetto, C. Geuzaine, J.-F. Remacle, and E. Sauvage. Quality open source mesh generation for cardiovascular flow simulations. In D. Ambrosi, A. Quarteroni, and G. Rozza, editors, *Modeling of Physiological Flows*, volume 5 of *MS&A – Modeling, Simulation and Applications*, pages 395–414. Springer Milan, 2012.

[ME97] J.A. Moore and C.R. Ethier. Oxygen mass transfer calculations in large arteries. *J. Biomech. Eng.*, 119(4):469–475, 1997.

[MN15] A. Manzoni and F. Negri. Heuristic strategies for the approximation of stability factors in quadratically nonlinear parametrized PDEs. *Adv. Comput. Math.*, 2015. DOI: 10.1007/s10444-015-9413-4.

[MNPP09] Y. Maday, N.C. Nguyen, A.T. Patera, and G.S.H. Pau. A general multipurpose interpolation procedure: the magic points. *Commun. Pure Appl. Anal.*, 8(1), 2009.

[Moo81] B.C. Moore. Principal component analysis in linear systems: controllability, observability and model reduction. *IEEE Trans. Autom. Control*, AC-26(1):17–32, 1981.

[MPL14] A. Manzoni, S. Pagani, and T. Lassila. Accurate solution of Bayesian inverse uncertainty quantification problems using model and error reduction methods. Mathicse Report Nr. 47.2014 (submitted), 2014.

[MPPY15] Y. Maday, A.T. Patera, J.D. Penn, and M. Yano. A parametrized-background data-weak approach to variational data assimilation: formulation, analysis, and application to acoustics. *Int. J. Numer. Methods Engrg.*, 102:933–965, 2015.

[MQR12a] A. Manzoni, A. Quarteroni, and G. Rozza. Model reduction techniques for fast blood flow simulation in parametrized geometries. *Int. J. Numer. Meth. Biomed. Engng.*, 28(6-7):604–625, 2012.

[MQR12b] A. Manzoni, A. Quarteroni, and G. Rozza. Shape optimization for viscous flows by reduced basis method and free-form deformation. *Int. J. Numer. Methods Fluids*, 70(5):646–670, 2012.

[MRV13] S. May, R. Rannacher, and B. Vexler. Error analysis for a finite element approximation of elliptic Dirichlet boundary control problems. *SIAM J. Control Optim.*, 51(3):2585–2611, 2013.

[Neg11] F. Negri. Reduced basis method for parametrized optimal control problems governed by PDEs. Master's thesis, Politecnico di Milano, Milano, 2011.

[Neg15] F. Negri. A model order reduction framework for parametrized nonlinear PDE-constrained optimization. Technical report, EPFL, 2015. Submitted, available online as MATHICSE report 11.2015.

[Nic82] R. Nicolaides. Existence, uniqueness and approximation for generalized saddle point problems. *SIAM J. Numer. Anal.*, 19(5):349–357, 1982.

[NMA15] F. Negri, A. Manzoni, and D. Amsallem. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *J. Comp. Phys.*, 2015. To appear, DOI: 10.1016/j.jcp.2015.09.046.

[NMR15] F. Negri, A. Manzoni, and G. Rozza. Reduced basis approximation of parametrized optimal flow control problems for the Stokes equations. *Comput. Math. Appl.*, 69(4):319 – 336, 2015.

[NP08] N.C. Nguyen and J. Peraire. An efficient reduced-order modeling approach for non-linear parametrized partial differential equations. *Int. J. Numer. Meth. Engrg.*, 76(1):27–55, 2008.

[NPM05] B.R. Noack, P. Papas, and P.A. Monkewitz. The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. *J. Fluid Mech.*, 523:339–365, 2005.

[NPP08] N.C. Nguyen, A.T. Patera, and J. Peraire. A 'best points' interpolation method for efficient approximation of parametrized functions. *Int. J. Numer. Methods Engrg.*, 73(4):521–543, 2008.

[NRMQ13] F. Negri, G. Rozza, A. Manzoni, and A. Quarteroni. Reduced basis method for parametrized elliptic optimal control problems. *SIAM J. Sci. Comput.*, 35(5):A2316–A2340, 2013.

[NW06] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, New York, 2006.

[ODM12] A.A. Owida, H. Do, and Y.S Morsi. Numerical analysis of coronary artery bypass grafts: An over view. *Comput. Meth. Prog. Bio.*, 108(2):689 – 705, 2012.

[OKP08] U. Olgac, V. Kurtcuoglu, and D. Poulikakos. Computational modeling of coupled blood-wall mass transport of LDL: effects of local wall shear stress. *Am. J. Physiol. Heart Circ. Physiol.*, 294(2):H909–H919, 2008.

[Olg12] U. Olgac. Patient-specific modeling of low-density lipoprotein transport in coronary arteries. In U. Demirci, A. Khademhosseini, R. Langer, and J. Blander, editors, *Microfluidic Technologies for Human Health*. World Scientific, 2012.

[OPS14] G. Of, T.X. Phan, and O. Steinbach. An energy space finite element approach for elliptic dirichlet boundary control problems. *Numer. Math.*, 129(4):723–748, 2014.

[PBC06] E.E. Prudencio, R Byrd, and X.-C. Cai. Parallel full space SQP Lagrange–Newton–Krylov–Schwarz algorithms for PDE-constrained optimization problems. *SIAM J. Sci. Comput.*, 27(4):1305–1328, 2006.

[PDTA14] A. Paul-Dubois-Taine and D. Amsallem. An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models. *Int. J. Numer. Methods Engrg.*, pages 1–31, 2014.

[PHV15] F. Pla, H. Herrero, and J.M. Vega. A flexible symmetry-preserving Galerkin/POD reduced order model applied to a convective instability problem. *Comput. Fluids*, 2015.

[Pin08] R. Pinnau. Model reduction via proper orthogonal decomposition. In W.H.A. Schilder, H. van der Vorst, and J. Rommes, editors, *Model Order Reduction: Theory, Research Aspects and Applications*, pages 96–109. Springer, 2008.

[PLP+02] K. Perktold, A. Leuprecht, M. Prosi, T. Berk, M. Czerny, W. Trubel, and H. Schima. Fluid dynamics, wall mechanics, and oxygen transfer in peripheral bypass anastomoses. *Ann. Biomed. Eng.*, 30(4):447–460, 2002.

[PP03] K. Perktold and M. Prosi. Computational models of arterial flow and mass transport. In G. Pedrizzetti and K. Perktold, editors, *Cardiovascular Fluid Mechanics*, volume 446 of *International Centre for Mechanical Sciences*, pages 73–136. Springer Vienna, 2003.

[PRV+02] C. Prud'homme, D.V. Rovas, K. Veroy, L. Machiels, Y. Maday, A.T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *J. Fluid Eng.*, 124(1):70–80, 2002.

[PVS+09] M. Piccinelli, A. Veneziani, D. Steinman, A. Remuzzi, and L. Antiga. A framework for geometric analysis of vascular structures: application to cerebral aneurysms. *IEEE Trans. Med. Imag*, 28(8):1141–1155, 2009.

[PZPQ05] M. Prosi, P. Zunino, K. Perktold, and A. Quarteroni. Mathematical and numerical models for transfer of low-density lipoproteins through the arterial walls: a new methodology for the model set up with applications to the study of disturbed lumenal flow. *J. Biomech.*, 38(4):903–917, 2005.

[QMN16] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*. Springer International Publishing, Switzerland, 2016.

[QR07] A. Quarteroni and G. Rozza. Numerical solution of parametrized Navier-Stokes equations by reduced basis methods. *Numer. Meth. Part. D. E.*, 23(4):923–948, 2007.

[QR14] A. Quarteroni and G. Rozza, editors. *Reduced Order Methods for Modeling and Computational Reduction*, volume 9 of *Modeling, Simulation and Applications (MS&A)*. Springer International Publishing, Switzerland, 2014.

[QRQ07] A. Quarteroni, G. Rozza, and A. Quaini. Reduced basis methods for optimal control of advection-diffusion problem. In W. Fitzgibbon, R. Hoppe, J. Periaux, , O. Pironneau, and Y. Vassilevski, editors, *Advances in Numerical Mathematics*, pages 193–216, 2007.

[QSS07] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer, New York, second edition, 2007.

[QV94] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer-Verlag, Berlin-Heidelberg, 1994.

[QVZ01] A. Quarteroni, A. Veneziani, and P. Zunino. Mathematical and numerical modeling of solute dynamics in blood flow and arterial walls. *SIAM J. Numer. Anal.*, 39(5):1488–1511, 2001.

[QVZ02] A. Quarteroni, A. Veneziani, and P. Zunino. A domain decomposition method for advection-diffusion processes with application to blood solutes. *SIAM J. Sci. Comput.*, 23(6):1959–1980, 2002.

[Ram02] T. Ramsay. Spline smoothing over difficult regions. *J. Roy. Statist. Soc. Ser. B*, 64(2):307–319, 2002.

[Rav00] S.S. Ravindran. A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. *Int. J. Numer. Meth. Fluids*, 34:425–448, 2000.

[RBAB+08] V.L. Rayz, L. Boussel, G. Acevedo-Bolton, A.J. Martin, W.L. Young, M.T. Lawton, R. Higashida, and D. Saloner. Numerical simulations of flow in cerebral aneurysms: comparison of cfd results and in vivo mri measurements. *J. Biomech. Eng.*, 130(5):051011, 2008.

[RDW10] T. Rees, H. S. Dollar, and A. J. Wathen. Optimal solvers for PDE-constrained optimization. *SIAM J. Sci. Comput.*, 32:271–298, 2010.

[RG12] M. Rasty and M.A. Grepl. Efficient reduced basis solution of quadratically nonlinear diffusion equations. In *Proc. MATHMOD 2012, 7th Vienna International Conference on Mathematical Modelling*, 2012.

[RHM13] G. Rozza, D.B.P. Huynh, and A. Manzoni. Reduced basis approximation and a posteriori error estimation for Stokes flows in parametrized geometries: roles of the inf-sup stability constants. *Numer. Math.*, 125(1):115–152, 2013.

[RHP08] G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Arch. Comput. Methods Eng.*, 15:229–275, 2008.

[Rin15] M. Rinaldi. Reduced Basis Method for Isogeometric Analysis: Application to Structural Problems. Master's thesis, Politecnico di Milano, Milano, 2015.

[RMN12] G. Rozza, A. Manzoni, and F. Negri. Reduced strategies for PDE-constrained optimization problems in haemodynamics. In *Proceedings of the 6th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012), Vienna, Austria*, 2012.

[Rov03] D.V. Rovas. *Reduced-Basis Output Bound Methods for Parametrized Partial Differential Equations*. PhD thesis, Massachusetts Institute of Technology, Cambridge, 2003.

[RP96a] G. Rappitsch and K. Perktold. Computer simulation of convective diffusion processes in large arteries. *J. Biomech.*, 29(2):207–215, 1996.

[RP96b] G. Rappitsch and K. Perktold. Pulsatile albumin transport in large arteries: a numerical simulation study. *J. Biomech. Eng.*, 118(4):511–519, 1996.

[RPP97] G. Rappitsch, K. Perktold, and E. Pernkopf. Numerical modelling of shear-dependent mass transfer in large arteries. *Int. J. Numer. Methods Fluids*, 25(7):847–857, 1997.

[RSW10] T. Rees, M. Stoll, and A.J. Wathen. All-at-once preconditioning in PDE-constrained optimization. *Kybernetika*, 46(2):341–360, 2010.

[RV07] G. Rozza and K. Veroy. On the stability of the reduced basis method for Stokes equations in parametrized domains. *Comput. Methods Appl. Mech. Engrg.*, 196(7):1244 – 1260, 2007.

[RW06] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, 2006.

[RW11] T. Rees and A. J. Wathen. Preconditioning iterative methods for the optimal control of the Stokes equation. *SIAM J. Sci. Comput.*, 33(5):2903–2926, 2011.

[Ryc09] D. Ryckelynck. Hyper-reduction of mechanical models involving internal variables. *Int. J. Numer. Methods Engrg.*, 77(1):75–89, 2009.

[SA02] D.C. Sorensen and A.C. Antoulas. The Sylvester equation and approximate balanced reduction. *Linear Algebra Appl.*, 351:671–700, 2002.

[Sal04] M. Sala. Analysis of two-level domain decomposition preconditioners based on aggregation. *ESAIM: Math. Model. Numer. Anal.*, 38(05):765–780, 2004.

[Sch97] W.I. Schievink. Intracranial aneurysms. *N. Engl. J. Med.*, 336(1):28–40, 1997.

[SE02] D.K. Stangeby and C.R. Ethier. Computational analysis of coupled blood-wall arterial LDL transport. *J. Biomech. Eng.*, 124(1):1–8, 2002.

[Sir87] L. Sirovich. Turbulence and the dynamics of coherent structures, part I: Coherent structures. *Quart. Appl. Math.*, 45(3):561–571, 1987.

[SLW+14] N. Sun, J.H. Leung, N.B. Wood, A.D. Hughes, S.A. Thom, N.J. Cheshire, and X.Y. Xu. Computational analysis of oxygen transport in a patient-specific model of abdominal aortic aneurysm with intraluminal thrombus. *Brit. J. Radiol.*, 2014.

[SOT+04] M. Shojima, M. Oshima, K. Takagi, R. Torii, M. Hayakawa, K. Katada, A. Morita, and T. Kirino. Magnitude and role of wall shear stress on cerebral aneurysm computational fluid dynamic study of 20 middle cerebral artery aneurysms. *Stroke*, 35(11):2500–2505, 2004.

[SS90] G.W. Stewart and J. Sun. *Matrix Perturbation Theory*. Academic Press, New York, 1990.

[SS14] R. Stefanescu and A. Sandu. Efficient approximation of sparse jacobians for time-implicit reduced order models. Technical report, Virginia Polytechnic Institute and State University, CSTR-19/2015, 2014.

[STD+96] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher. Benchmark computations of laminar flow around a cylinder. In E.H. Hirschel, editor, *Flow Simulation with High-Performance Computers II*, volume 48 of *Notes on Numerical Fluid Mechanics (NNFM)*, pages 547–566. Vieweg+Teubner Verlag, Wiesbaden, 1996.

[STL00] S.M. Santilli, A.S. Tretinyak, and E.S. Lee. Transarterial wall oxygen gradients at the deployment site of an intra-arterial stent in the rabbit. *Am. J. Physiol. Heart Circ. Physiol.*, 279(4):H1518–H1525, 2000.

[SvdVR08] W.H.A. Schilders, H. A. van der Vorst, and J. Rommes, editors. *Model Order Reduction: Theory, Research Aspects and Applications*, volume 13 of *Mathematics in Industry*. Springer, Berlin Heidelberg, 2008.

[SVH+06] S. Sen, K. Veroy, D.B.P. Huynh, S. Deparis, N.C. Nguyen, and A.T. Patera. "Natural norm" a posteriori error estimators for reduced basis approximations. *J. Comp. Phys.*, 217(1):37–62, 2006.

[SZ07] J. Schöberl and W. Zulehner. Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems. *SIAM J. Matrix Anal. Appl.*, 29:752–773, 2007.

[TALL15] A. Tallet, C. Allery, C. Leblond, and E. Liberge. A minimum residual projection to build coupled velocity–pressure POD–ROM for incompressible Navier-Stokes equations. *Commun. Nonlinear. Sci. Numer. Simul.*, 22(1-3):909–932, 2015.

[Tem01] R. Temam. *Navier-Stokes Equations: Theory and Numerical Analysis*. AMS Chelsea Publishing, Providence, 2001.

[TMV+03] S. Tateshima, Y. Murayama, J.P. Villablanca, T. Morino, K. Nomura, K. Tanishita, and F. Viñuela. In vitro measurement of fluid-induced wall shear stress in unruptured cerebral aneurysms harboring blebs. *Stroke*, 34(1):187–192, 2003.

[Ton12] T. Tonn. *Reduced-Basis Method (RBM) for Non-Affine Elliptic Parametrized PDEs (Motivated by Optimization in Hydromechanics)*. PhD thesis, Universitat Ulm, Ulm, 2012.

[TR13] P. Tiso and D.J. Rixen. Discrete empirical interpolation method for finite element structural dynamics. In G. Kerschen, D. Adams, and A. Carrella, editors, *Topics in Nonlinear Dynamics, Volume 1*, volume 35 of *Conference Proceedings of the Society for Experimental Mechanics Series*, pages 203–212. Springer, New York, 2013.

[Tri14] P. Tricerri. *Mathematical and Numerical Modeling of Healthy and Unhealthy Cerebral Arterial Tissues*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Lausanne, 2014.

[Trö10] F. Tröltzsch. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, volume 112 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, 2010.

[TUV10] T. Tonn, K. Urban, and S. Volkwein. Optimal control of parameter-dependent convection-diffusion problems around rigid bodies. *SIAM J. Sci. Comput.*, 32(3):1237–1260, 2010.

[TUV11] T. Tonn, K. Urban, and S. Volkwein. Comparison of the reduced-basis and POD a-posteriori estimators for an elliptic linear-quadratic optimal control problem. *Math. Comput. Model. Dyn. Syst.*, 17(1):355–369, 2011.

[TV09] F. Tröltzsch and S. Volkwein. POD a-posteriori error estimates for linear-quadratic optimal control problems. *Comput. Optim. Appl.*, 44:83–115, 2009.

[UB08] R. Udawalpola and M. Berggren. Optimization of an acoustic horn with respect to efficiency and directivity. *Int. J. Numer. Methods Engrg.*, 73(11):1571–1606, 2008.

[Ver13] R. Verfürth. *A posteriori Error Estimation Techniques for Finite Element Methods.* Oxford University Press, Oxford, 2013.

[Vex07] B. Vexler. Finite element approximation of elliptic Dirichlet optimal control problems. *Numer. Funct. Anal. Optim.*, 28(7-8):957–973, 2007.

[VGFA06] A.A. Valencia, A.M. Guzmán, E.A. Finol, and C.H. Amon. Blood flow dynamics in saccular aneurysm models of the basilar artery. *J. Biomech. eng.*, 128(4):516–526, 2006.

[VMR+08] A. Valencia, H. Morales, R. Rivera, E. Bravo, and M. Galvez. Blood flow dynamics in patient-specific cerebral aneurysm models: the relationship between wall shear stress and aneurysm area index. *Med. Eng. Phys.*, 30(3):329–340, 2008.

[Vol00] S. Volkwein. Mesh-independence for an augmented Lagrangian-SQP method in Hilbert spaces. *SIAM J. Control Optim.*, 38(3):767–785, 2000.

[Vol11] S. Volkwein. Model reduction using proper orthogonal decomposition, 2011. Lecture Notes, University of Konstanz.

[VP05] K. Veroy and A.T. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. *Int. J. Numer. Meth. Fluids*, 47:773–788, 2005.

[VPP03] K. Veroy, C. Prud'homme, and A. T. Patera. Reduced-basis approximation of the viscous Burgers equation: Rigorous *a posteriori* error bounds. *C. R. Acad. Sci. Paris, Série I*, 337(9):619–624, 2003.

[VPRP03] K. Veroy, C. Prud'homme, D.V. Rovas, and A.T. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *Proceedings of the 16th AIAA computational fluid dynamics conference*, volume 3847, 2003.

[WABI12] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. *Comput. Meth. Appl. Mech. Engrg.*, 237–240:10–26, 2012.

[WC14] Y. Wu and X.C. Cai. A fully implicit domain decomposition based ALE framework for three-dimensional fluid-structure interaction with application in blood flow computation. *J. Comput. Phys.*, 258:524–537, 2014.

[WH15] Y. Wu and U. Hetmaniuk. Adaptive training of local reduced bases for unsteady incompressible Navier–Stokes flows. *Int. J. Numer. Methods Engr.*, 2015.

[Wil06] K. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Comput. Fluids*, 35(2):208–226, 2006.

[WLBI10] J. Weller, E. Lombardi, M. Bergmann, and A. Iollo. Numerical methods for low-order modeling of fluid flows based on POD. *Int. J. Numer. Methods Fluids*, 63(2):249–268, 2010.

[WSH14] D. Wirtz, D.C. Sorensen, and B. Haasdonk. A posteriori error estimation for DEIM reduced nonlinear dynamical systems. *SIAM J. Sci. Comput.*, 36(2):A311–A338, 2014.

[XFB+14] 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 the residual DEIM method. *J. Comp. Phys.*, 263:1–18, 2014.

[XZ03] J. Xu and L. Zikatanov. Some observations on Babuška and Brezzi theories. *Numer. Math.*, 94:195–202, 2003.

[Yan14] M. Yano. A space-time Petrov-Galerkin certified reduced basis method: Application to the Boussinesq equations. *SIAM J. Sci. Comput.*, 36(1):A232–A266, 2014.

[YCC+14] L. Yoshihara, M. Coroneo, A. Comerford, G. Bauer, T. Klöppel, and W.A. Wall. A combined fluid–structure interaction and multi–field scalar transport model for simulating mass transport in biomechanics. *Int. J. Numer. Methods Engng.*, 100(4):277–299, 2014.

[YPU14] M. Yano, A.T. Patera, and K. Urban. A space-time hp-interpolation-based certified reduced basis method for Burgers' equation. *Math. Mod. Meth. Appl. S.*, 24(09):1903–1935, 2014.

[Zei85] E. Zeidler. *Nonlinear Functional Analysis and its Applications*, volume I: Fixed-Point Theorems. Springer-Verlag, New York, 1985.

[ZF15] M. J. Zahr and C. Farhat. Progressive construction of a parametric reduced-order model for PDE-constrained optimization. *Int. J. Numer. Methods Engng.*, 102(5):1111–1135, 2015.

[Zul11] W. Zulehner. Nonstandard norms and robust estimates for saddle point problems. *SIAM J. Matrix Anal. Appl.*, 32(2):536–560, 2011.

Curriculum Vitae

Personal details

Birth	December 15, 1987
Place of birth	Lodi, Italy
Address (work)	Av. Piccard, Station 8, CH-1015 Lausanne, Switzerland.
Phone (work)	+41 21 69 30352
Mail	federico.negri@epfl.ch

Work experience

Doctoral assistant — May 2012 - October 2015
Chair of Modeling and Scientific Computing, EPFL, Lausanne, Switzerland

Education

Ph.D. in Mathematics — May 2012 - October 2015
Ecole Polytechnique Fédérale de Lausanne

Laboratory:	Chair of Modeling and Scientific Computing
Thesis title:	Efficient reduction techniques for the simulation and optimization of parametrized systems: analysis and applications
Advisor:	Prof. A. Quarteroni
Co-Advisor:	Prof. G. Rozza

Master degree in Mathematical Engineering — Sept. 2009 - Dec. 2011
Politecnico di Milano

Thesis title:	Reduced basis method for parametrized optimal control problems governed by PDEs
Advisor:	Prof. A. Quarteroni
Co-advisor:	Prof. G. Rozza
Grade:	110/110 cum laude

Bachelor degree in Mathematical Engineering Sept. 2006 - Sept. 2009
Politecnico di Milano
Thesis title: Singularly perturbed quasilinear ordinary differential equations
Advisor: Prof. P. Biscari
Grade: 110/110 cum laude

Scientific publications

G. Rozza, A. Manzoni, and F. Negri. *Reduced strategies for PDE-constrained optimization problems in haemodynamics*. In *Proceedings of the 6th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012), Vienna, Austria*, 2012.

F. Negri, G. Rozza, A. Manzoni, and A. Quarteroni. *Reduced basis method for parametrized elliptic optimal control problems*. SIAM J. Sci. Comput. 35(5), A2316–A2340, 2013.

F. Negri, A. Manzoni, and G. Rozza. *Reduced basis approximation of parametrized optimal flow control problems for the Stokes equations*. Comput. Math. Appl. 69(4), 319-336, 2015.

A. Manzoni and F. Negri. *Heuristic strategies for the approximation of stability factors in quadratically nonlinear parametrized PDEs*. Adv. Comput. Math., 2015. DOI: 10.1007/s10444-015-9413-4.

F. Negri, A. Manzoni, and D. Amsallem. *Efficient model reduction of parametrized systems by matrix discrete empirical interpolation method*. J. Comput. Phys., accepted for publication, 2015. DOI: 10.1016/j.jcp.2015.09.046

F. Negri. *A model order reduction framework for parametrized nonlinear PDE-constrained optimization*. Submitted, 2015. Available online as MATHICSE report 11.2015.

A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations. An Introduction.* Springer International Publishing, Switzerland, 2016.

Conferences and workshops

University of Konstanz, Invited Seminar, April 15-19, 2012.

CECAM Workshop on *Reduced Basis, POD and Reduced Order Methods for model and computational reduction: towards real-time computing and visualization?*, Poster session, EPFL, Lausanne, Switzerland, May 14-16, 2012.

SIMAI 2012, Contributed talk, Politecnico di Torino, Italy, June 25-28, 2012.

Swiss Numerics Colloquium 2013, Poster session, EPFL, Switzerland, April 5, 2013.

MPF2013 - V International Symposium on Modelling of Physiological Flows, Contributed

talk, Chia Laguna, Sardinia, Italy, June 11-14, 2013.

ENUMATH2013 - European Numerical Mathematics and Advanced Applications, Contributed talk, EPFL, Lausanne, Switzerland, August 26-30, 2013.

Domain Decomposition Methods for Optimization with PDE Constraints, Invited talk in a minisymposium, Ascona, Switzerland, September 1-6, 2013.

IFAC Workshop on Control of Systems Modeled by Partial Differential Equations, Invited talk in a minisymposium, Henri Poincaré Institute, Paris, France, September 25-27, 2013.

SIAM CSE 2015, Invited talk in a minisymposium, March 13-18, 2015, Salt Lake City, Utah, U.S.A.

Teaching activities

Teaching assistant of the bachelor/master course *Introduction to the finite element method* held by Prof. A. Quarteroni in the autumn semesters 2012-2013, 2013-2014 and 2014-2015.

Teaching assistant of the bachelor course *Analyse numérique* held by Dr. R. Ruiz-Bayer in the spring semester 2012-2013 and by Prof. A. Quarteroni in the spring semesters 2013-2014, 2014-2015.

Co-advisor of the master thesis *Reduced basis method for isogeometric analysis: application to structural problems* by M. Rinaldi (exchange student from Mathematical Engineering, Politecnico di Milano), 2014.

Co-supervisor of the following semester projects:

- *Optimal control of parabolic equations: an inverse problem* by A. Imboden (bachelor student in Mathematics), spring semester 2012-2013;
- *Numerical simulation of drug delivery in blood flows* by J. Droxler (master student in Mathematics), autumn semester 2013-2014;
- *Numerical approximation of the Navier-Stokes-Forchheimer equations and application to a passive flow control problem* by L. Hausammann (master student in Physics), autumn semester 2014-2015.