DE GRUYTER

Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni,
Gianluigi Rozza, Wil Schilders, Luís Miguel Silveira (Eds.)

MODEL ORDER REDUCTION

VOLUME 2: SNAPSHOT-BASED METHODS AND ALGORITHMS

Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, Wil Schilders, Luís Miguel Silveira (Eds.)
Model Order Reduction

Also of Interest
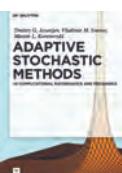
Model Order Reduction. Volume 1: System- and Data-Driven Methods and Algorithms
Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, Wil Schilders, Luís Miguel Silveira (Eds.) 2021
ISBN 978-3-11-050043-1, e-ISBN (PDF) 978-3-11-049896-7, e-ISBN (EPUB) 978-3-11-049771-7

Model Order Reduction. Volume 3: Applications
Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza, Wil Schilders, Luís Miguel Silveira (Eds.), 2021
ISBN 978-3-11-050044-8, e-ISBN (PDF) 978-3-11-049900-1, e-ISBN (EPUB) 978-3-11-049775-5

Adaptive Stochastic Methods. In Computational Mathematics and Mechanics
Dmitry G. Arseniev, Vladimir M. Ivanov, Maxim L. Korenevsky, 2018
ISBN 978-3-11-055364-2, e-ISBN (PDF) 978-3-11-055463-2, e-ISBN (EPUB) 978-3-11-055367-3

Multivariate Algorithms and Information-Based Complexity
Fred J. Hickernell, Peter Kritzer (Eds.), 2020
ISBN 978-3-11-063311-5, e-ISBN (PDF) 978-3-11-063546-1, e-ISBN (EPUB) 978-3-11-063315-3

Dynamics of Solid Structures. Methods using Integrodifferential Relations
Georgy Viktorovich Kostin, Vasily V. Saurin, 2017
ISBN 978-3-11-051623-4, e-ISBN (PDF) 978-3-11-051644-9, e-ISBN (EPUB) 978-3-11-051625-8

Model Order Reduction

Volume 2: Snapshot-Based Methods and Algorithms

Edited by
Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni,
Gianluigi Rozza, Wil Schilders, and Luís Miguel Silveira

DE GRUYTER

Editors

Prof. Dr. Peter Benner
Max Planck Institute for Dynamics of Complex
Technical Systems
Sandtorstr. 1
39106 Magdeburg
Germany
benner@mpi-magdeburg.mpg.de

Prof. Dr. Stefano Grivet-Talocia
Politecnico di Torino
Dipartimento di Elettronica
Corso Duca degli Abruzzi 24
10129 Turin
Italy
stefano.grivet@polito.it

Prof. Alfio Quarteroni
Ecole Polytechnique Fédérale de Lausanne
(EPFL) and Politecnico di Milano
Dipartimento di Matematica
Piazza Leonardo da Vinci 32
20133 Milan
Italy
alfio.quarteroni@polimi.it

Prof. Dr. Gianluigi Rozza
Scuola Internazionale Superiore di Studi
Avanzati - SISSA
Via Bonomea 265
34136 Trieste
Italy
gianluigi.rozza@sissa.it

Prof. Dr. Wil Schilders
Technische Universiteit Eindhoven
Faculteit Mathematik
Postbus 513
5600 MB Eindhoven
The Netherlands
w.h.a.schilders@tue.nl

Prof. Dr. Luís Miguel Silveira
INESC ID Lisboa
IST Técnico Lisboa
Universidade de Lisboa
Rua Alves Redol 9
1000-029 Lisbon
Portugal
lms@inesc-id.pt

Preface to the second volume of *Model Order Reduction*

This second volume of the *Model Order Reduction* handbook project mostly focuses on snapshot-based methods for parameterized partial differential equations. This approach has seen tremendous development in the past two decades, especially in the broad domain of computational mechanics. However, the main ideas were already known long before; see, e. g., the seminal work by J. L. Lumley, "The structure of inhomogeneous turbulent flows," in *Atmospheric Turbulence and Radio Wave Propagation*, 1967, for proper orthogonal decomposition (POD), and the one by A. K. Noor and J. N. Peters, *Reduced basis technique for nonlinear analysis of structures*, AIAA Journal, Vol. 4, 1980, for the reduced basis method.

The most popular mathematical strategy behind snapshot-based methods relies on Galerkin projection on finite-dimensional subspaces generated by snapshot solutions corresponding to a special choice of parameters. Because of that, it is often termed as a projection-based intrusive approach. A suitable offline-online splitting of the computational steps, as well as the use of hyperreduction techniques to be used for the nonlinear (or nonaffine) terms and nonlinear residuals, is key to efficiency.

The first chapter, by G. Rozza et al., introduces all the preliminary notions and basic ideas to start delving into the topic of snapshot-based model order reduction. All the notions will be recast into a deeper perspective in the following chapters.

The second chapter, by Grässle et al., provides an introduction to POD with a focus on (nonlinear) parametric partial differential equations (PDEs) and (nonlinear) time-dependent PDEs, and PDE-constrained optimization with POD surrogate models as application. Several numerical examples are provided to support the theoretical findings.

A second scenario in the methodological development is provided in the third chapter, by Chinesta and Ladevèze, on proper generalized decomposition, a research line significantly grown in the last couple of decades also thanks to real-world applications. Basic concepts used here rely on the separation of variables (time, space, design parameters) and tensorization.

The fourth chapter, by Maday and Patera, focuses on the reduced basis method, including a posteriori error estimation, as well as a primal-dual approach. Several combinations of these approaches have been proposed in the last few years to face problems of increasing complexity.

When facing nonaffine and nonlinear problems, the development of efficient reduction strategies is of paramount importance. These strategies can require either global or local (pointwise) subspace constructions. This issue is thoroughly covered in the fifth chapter, by Farhat et al., where several front-end computational problems in the field of nonlinear structural dynamics, scattering elastoacoustic wave propagation problems, and a parametric PDE-ODE wildfire model problem are presented.

In the sixth chapter, by Buhr et al., localized model order reduction is presented. With this approach, the model order reduction solution is constructed via a suitable coupling of local solutions whose support lies within a subdomain of the global computational domain. Applications are provided for multiscale, linear elasticity, and fluid-flow problems.

Last but not least, the final chapter, by Brunton and Kutz, addresses a snapshot-based nonintrusive data-driven method. In particular, dynamic mode decomposition and its Koopman generalization are used to discover low-rank spatio-temporal patterns of activity, and to provide approximations in terms of linear dynamical systems, which are amenable to simple analysis techniques. These methods can be used in a nonintrusive, equation-free manner for improved computational performance of parametric PDE systems.

Several chapters contain instructive descriptions of algorithms that can serve as templates for implementing the discussed approaches in problem-specific environments.

Peter Benner, Stefano Grivet-Talocia, Alfio Quarteroni, Gianluigi Rozza,
Wil Schilders, Luìs Miguel Silveira

Magdeburg, Germany
Torino, Milano, Trieste, Italy
Eindhoven, The Netherlands
Lisbon, Portugal

June 2020

Contents

Gianluigi Rozza, Martin Hess, Giovanni Stabile, Marco Tezzele, and Francesco Ballarin

1 Basic ideas and tools for projection-based model reduction of parametric partial differential equations

Abstract: We provide first the functional analysis background required for reduced-order modeling and present the underlying concepts of reduced basis model reduction. The projection-based model reduction framework under affinity assumptions, offline-online decomposition, and error estimation are introduced. Several tools for geometry parameterizations such as free form deformation, radial basis function interpolation, and inverse distance weighting interpolation are explained. The empirical interpolation method is introduced as a general tool to deal with nonaffine parameter dependency and nonlinear problems. The discrete and matrix versions of the empirical interpolation are considered as well. Active subspace properties are discussed to reduce high-dimensional parameter spaces as a preprocessing step. Several examples illustrate the methodologies.

Keywords: reduced basis method, radial basis function interpolation, shape morphing techniques, empirical interpolation method, active subspaces

MSC 2010: 65D99, 65J05, 65M15

Introduction

Parametric model order reduction (MOR) techniques have been developed in recent decades to deal with increasingly complex computational tasks. The ability to compute how quantities of interest change with respect to parameter variations provides insight and understanding, which is vital in all areas of science and engineering. Model reduction thus allows to deal with optimization or inverse problems of a whole new scale. Each chapter of the handbook gives an in-depth view of a MOR method,

Gianluigi Rozza, Martin Hess, Giovanni Stabile, Marco Tezzele, Francesco Ballarin, Mathematics Area, SISSA mathLab, Trieste, Italy, e-mails: gianluigi.rozza@sissa.it, martin.hess@sissa.it, giovanni.stabile@sissa.it, marco.tezzele@sissa.it, francesco.ballarin@sissa.it

a particular application area, and analytical, numerical, or technical aspects of software frameworks for model reduction.

There exist a large number of MOR techniques used in many areas of science and engineering to improve computational performances and contain costs in a repetitive computational environment, such as many-query and real-time computing [93]. We assume a given parameterized partial differential equation (PDE) as starting point of the model reduction procedure. Typical parameters of interest are material coefficients, corresponding to physical qualities of the media which constitute the domain where the PDE is solved. Also a variable geometry can be of special interest in a task to find the optimal device configuration. Physical states such as the temperature might be considered an input parameter. It is a task of the mathematical modeling to identify the parameters of interest and how they enter the PDE. Once a parameterized model is identified, the MOR techniques described in this and the following chapters can be used either in a "black-box" fashion (nonintrusive way) or by intrusive means, which will be explained in detail, whenever this is necessary.

The particular numerical method to solve a PDE is most often not relevant to the model reduction procedure. We will therefore assume there is a numerical method available, which solves the problem to any required accuracy, and move seamlessly from the continuous form to the discretized form.

This chapter covers briefly the functional analysis framework relevant to many, but not all, MOR methods. Presented is the starting point of PDE-oriented MOR techniques, such as the POD method found in Chapter 2 of this volume, the PGD method found in Chapter 3 of this volume, the reduced basis method found in Chapter 4 of this volume, the hyperreduction technique found in Chapter 5 of this volume, the localized reduced-order modeling (ROM) found in Chapter 6 of this volume, and the data-driven methods found in Chapter 7 of this volume.

In particular, Section 1.1 provides what is needed for the projection-based ROM. Starting from the setting of the classical Lax–Milgram theorem for elliptic PDEs in Sections 1.1.1 and 1.1.2, a numerical discretization is introduced in Section 1.1.2.1. Due to brevity of representation, many concepts of functional analysis and theory of PDEs are only touched upon. Many references to the literature for further reading are given.

Projection-based ROM is presented in Section 1.1.3, with the following topics covered in detail: proper orthogonal decomposition (POD) in Section 1.1.3.1, the greedy algorithm in Section 1.1.3.2, the projection framework in Section 1.1.3.3, affine parameter dependency in Section 1.1.3.4, the offline-online decomposition in Section 1.1.3.6, and basic error estimation in Section 1.1.4.

Section 1.2 introduces efficient techniques for geometric parameterizations, arising from a reference domain approach, such as free form deformation (FFD) in Section 1.2.1, radial basis function (RBF) interpolation in Section 1.2.2, and inverse distance weighting (IDW) in Section 1.2.3.

A widely used method to generate an approximate affine parameter dependency is the *empirical interpolation method* (EIM). The original *EIM* is presented in Section 1.3

as well as the *discrete EIM* in Section 1.3.3 and further options in Section 1.3.4. Several numerical examples show the use of the *EIM* in Section 1.3.5.

Section 1.4 introduces active subspaces as a preprocessing step to reduce the parameter space dimension. Corresponding examples are provided in Section 1.4.3 and also nonlinear dimensionality reduction is briefly discussed in Section 1.4.5.

A brief conclusion and an outlook of the handbook are given in Section 1.5.

1.1 Basic notions and tools

We briefly cover a few main results of linear functional analysis and the analysis of PDEs. This material serves as a reminder of the underlying concepts of model reduction but cannot replace a textbook on these subjects. For a more thorough background, we refer to the literature on functional analysis [30, 110], PDEs [1, 47, 82, 88], and numerical methods [2, 6, 29, 52, 80, 105].

1.1.1 Parameterized partial differential equations

Let $\Omega \subset \mathbb{R}^d$ denote a spatial domain in $d = 1, 2$, or 3 dimensions with boundary $\partial\Omega$. A Dirichlet boundary $\Gamma_D \subset \partial\Omega$ is given, where essential boundary conditions on the field of interest are prescribed. Introduce a Hilbert space $V(\Omega)$ equipped with inner product $(\cdot, \cdot)_V$ and induced norm $\|\cdot\|_V$. A Hilbert space $V(\Omega)$ is a function space, i. e., a function $u \in V(\Omega)$ is seen as a point in the vector space V, as is common in functional analysis. Each $u \in V(\Omega)$ defines a mapping $x \in \Omega \mapsto u(x) \in \mathbb{R}$ or $x \in \Omega \mapsto u(x) \in \mathbb{C}$, depending on whether a real or complex Hilbert space is considered. In many applications, V is a subset of the Sobolev space $H^1(\Omega)$ as $V(\Omega) = \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\}$. Vector-valued Hilbert spaces can be constructed using the Cartesian product of $V(\Omega)$. Given a parameter domain $\mathcal{P} \subset \mathbb{R}^p$, a particular parameter point is denoted by the p-tuple $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_p)$. The set of all linear and continuous forms on V defines the dual space V'; let $L \in \mathcal{L}(V, V')$ denote a linear differential operator.

A field variable $u \in V : \Omega \to \mathbb{R}$ is defined implicitly as the solution to a parameterized linear PDE through the operator $L : V \times \mathcal{P} \to V'$ with $L(\cdot; \boldsymbol{\mu}) \in \mathcal{L}(V, V')$ and load vector $f_L(\boldsymbol{\mu}) \in V'$ for each fixed $\boldsymbol{\mu}$, as

$$L(u; \boldsymbol{\mu}) = f_L(\boldsymbol{\mu}). \tag{1.1}$$

As in the case of function spaces, operators between function spaces form vector spaces themselves, such as $L(\cdot; \boldsymbol{\mu}) \in \mathcal{L}(V, V')$, with $\mathcal{L}(V, V')$ being the space of operators mapping from the vector space V to V'.

Typical examples of scalar-valued linear PDEs are the Poisson equation, the heat equation, and the wave equation, while typical examples of vector-valued linear PDEs

are the Maxwell equations and the Stokes equations. The nonlinear case will be addressed in various chapters as well: Examples of nonlinear PDEs include the Navier–Stokes system and the equations describing nonlinear elasticity.

1.1.2 Parameterized variational formulation

The variational form or weak form of a parameterized linear PDE in the continuous setting is given as

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall v \in V, \tag{1.2}$$

with bilinear form $a : V \times V \times \mathcal{P} \rightarrow \mathbb{R}$ and linear form $f : V \times \mathcal{P} \rightarrow \mathbb{R}$. In many application scenarios, a particular output of interest is sought, given by the linear form $l : V \times \mathcal{P} \rightarrow \mathbb{R}$ as

$$s(\boldsymbol{\mu}) = l(u(\boldsymbol{\mu}); \boldsymbol{\mu}). \tag{1.3}$$

In the case that $a(\cdot, \cdot; \boldsymbol{\mu})$ is symmetric and $l = f$, the problem is called compliant.

For each $\boldsymbol{\mu} \in \mathcal{P}$ assume coercivity and continuity of the bilinear form $a(\cdot, \cdot; \boldsymbol{\mu})$, i. e.,

$$a(w, w; \boldsymbol{\mu}) \geq \alpha(\boldsymbol{\mu}) \|w\|_V^2, \tag{1.4}$$

$$a(w, v; \boldsymbol{\mu}) \leq \gamma(\boldsymbol{\mu}) \|w\|_V \|v\|_V, \tag{1.5}$$

and continuity of the linear form $f(\cdot; \boldsymbol{\mu})$,

$$f(w; \boldsymbol{\mu}) \leq \delta(\boldsymbol{\mu}) \|w\|_V, \tag{1.6}$$

with parameter-independent bounds, which satisfy $0 < \alpha \leq \alpha(\boldsymbol{\mu})$, $\gamma(\boldsymbol{\mu}) \leq \gamma < \infty$, and $\delta(\boldsymbol{\mu}) \leq \delta < \infty$. To do actual computations, the bilinear form is discretized into a linear equation. The coercivity property means that the matrix discretizing the bilinear form will be positive definite.

For fixed parameter the well-posedness of (1.2) is then established by the Lax–Milgram theorem.

Theorem 1.1 (Lax–Milgram theorem). *Let $a : V \times V \rightarrow \mathbb{R}$ be a continuous and coercive bilinear form over a Hilbert space V and $f \in V'$ a continuous linear form. Then the variational problem*

$$a(u, v) = f(v) \quad \forall v \in V \tag{1.7}$$

has a unique solution $u \in V$ and we have

$$\|u\|_V \leq \frac{1}{\alpha} \|f\|_V, \tag{1.8}$$

with the coercivity constant $\alpha > 0$ of the bilinear form.

Thus, in the parametric setting, the $\boldsymbol{\mu}$-dependence also carries over to the coercivity constant as $\alpha = \alpha(\boldsymbol{\mu})$.

The function space in which the field variable resides is called the ansatz space, while the second function space is called the test space, i.e., where a test function v resides. If the test space is distinct from the ansatz space, then the bilinear form is defined over $a : V \times W \times \mathcal{P} \rightarrow \mathbb{R}$ for V and W Hilbert spaces. With $f \in W'$ and for fixed $\boldsymbol{\mu}$, the well-posedness is then established through the Banach–Nečas–Babuška theorem.

Theorem 1.2 (Banach–Nečas–Babuška theorem). *Let V and W denote Hilbert spaces, let $a : V \times W \rightarrow \mathbb{R}$ be a continuous bilinear form, and $f \in W'$. Then the variational problem*

$$a(u, v) = f(v) \quad \forall v \in W \tag{1.9}$$

has a unique solution if and only if
(i) *the inf-sup condition holds, i.e.,*

$$\exists \beta > 0 \, , \, s.\,t., \, \beta \leq \inf_{v \in V \setminus \{0\}} \sup_{w \in W \setminus \{0\}} \frac{a(v, w)}{\|v\|_V \|w\|_W},$$

(ii) $\forall w \in W$:

$$\{a(v, w) = 0 \ \forall v \in V\} \implies w = 0.$$

1.1.2.1 Discretized parameterized variational formulation

The method of weighted residuals is used to cast (1.1) into a discrete variational formulation. Given the linear PDE $L(u; \boldsymbol{\mu}) = f_L(\boldsymbol{\mu})$, consider a discrete, i.e., finite-dimensional, approximation $u_h \in V_h \subset V$ to u as

$$u_h(\boldsymbol{\mu}) = \sum_{i=1}^{N_h} u_h^{(i)} \varphi^i. \tag{1.10}$$

The dimension of V_h is N_h and the set of ansatz functions $\varphi^i(\mathbf{x}) : \Omega \rightarrow \mathbb{R}$ belong to V. The $u_h^{(i)}$ are scalar coefficients such that the vector $\mathbf{u}_h = (u_h^{(1)}, \ldots, u_h^{(N_h)})^T \in \mathbb{R}^{N_h}$ is the coordinate representation of u_h in the basis $\{\varphi^i\}$ of V_h. A conforming discretization is considered, i.e., $V_h \subset V$ holds.

Plugging (1.10) into (1.1) yields the discrete residual $R(u_h(\boldsymbol{\mu})) = L(u_h(\boldsymbol{\mu}); \boldsymbol{\mu}) - f_L(\boldsymbol{\mu}) \in V'$. To compute the scalar coefficients $u_h^{(i)}$, Galerkin orthogonality is invoked, as

$$0 = (\varphi_j, R)_{(V, V')}, \quad j = 1 \ldots N_h, \tag{1.11}$$

where $(\cdot, \cdot)_{(V, V')}$ is the duality pairing between V and V'.

In short, Galerkin orthogonality means that the test space is orthogonal to the residual. In Ritz–Galerkin methods, the residual is tested against the same set of functions as the ansatz functions. If test space and trial space are different, one speaks of a Petrov–Galerkin method. Numerous discretization methods can be understood in terms of the method of weighted residuals. They are distinguished by the particular choice of trial and test space.

The well-posedness of the discrete setting follows the presentation of the continuous setting, by casting the equations and properties over V_h instead of V.

The weak form in the discrete setting is given as

$$a(u_h(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) = f(v_h; \boldsymbol{\mu}) \quad \forall v_h \in V_h, \tag{1.12}$$

with bilinear form $a : V_h \times V_h \times \mathcal{P} \to \mathbb{R}$ and linear form $f : V_h \times \mathcal{P} \to \mathbb{R}$. The discrete bilinear form is then derived from (1.11) through the integration-by-parts formula and Green's theorem.

Correspondingly, the discrete coercivity constant $\alpha_h(\mu)$ and the discrete continuity constant $\gamma_h(\mu)$ are defined as

$$\alpha_h(\boldsymbol{\mu}) = \min_{w_h \in V_h} \frac{a(w_h, w_h; \boldsymbol{\mu})}{\|w_h\|_{V_h}^2}, \tag{1.13}$$

$$\gamma_h(\boldsymbol{\mu}) = \max_{w_h \in V_h} \max_{v_h \in V_h} \frac{a(w_h, v_h; \boldsymbol{\mu})}{\|w_h\|_{V_h} \|v_h\|_{V_h}}. \tag{1.14}$$

The well-posedness of (1.2) is then analogously established by the Lax–Milgram theorem and the Banach–Nečas–Babuška theorem. Cea's lemma is a fundamental result about the approximation quality that can be achieved.

Lemma 1.3 (Cea's lemma). *Let $a : V \times V \to \mathbb{R}$ be a continuous and coercive bilinear form over a Hilbert space V and $f \in V'$ a continuous linear form. Given a conforming finite-dimensional subspace $V_h \subset V$, the continuity constant γ, and coercivity constant α of $a(\cdot, \cdot)$, for the solution u_h to*

$$a(u_h, v_h) = f(v_h) \quad \forall v_h \in V_h, \tag{1.15}$$

we have

$$\|u - u_h\|_V \le \frac{\gamma}{\alpha} \inf_{v_h \in V_h} \|u - v_h\|_V. \tag{1.16}$$

The stiffness matrix $\mathbb{A}_h \in \mathbb{R}^{N_h \times N_h}$ assembles the bilinear form entrywise as $(\mathbb{A}_h)_{ij} = a(\varphi^j, \varphi^i)$. The load vector $\mathbf{f}_h \in \mathbb{R}^{N_h}$ is assembled entrywise as $(\mathbf{f}_h)_i = f(\varphi^i)$ and the solution vector is denoted $\mathbf{u}_h$ with coefficients $u_h^{(j)}$.

Then solving (1.12) amounts to solving the linear system

$$\mathbb{A}_h \mathbf{u}_h = \mathbf{f}_h. \tag{1.17}$$

The most common discretization method is the finite element method [13], besides the finite difference [97], discontinuous Galerkin [4], finite volume [48], and spectral element methods [19].

1.1.3 Model reduction basic concepts

A wide variety of ROM methods exist today, thanks to large research efforts in the last decades. Reduced basis MOR is a projection-based MOR method and also shares many features with other MOR methods, so that the topics mentioned here will occur throughout the handbook. Two common algorithms for the generation of a projection space, POD and the greedy algorithm, are presented first.

1.1.3.1 Proper orthogonal decomposition

Assume a sampled set of high-fidelity solutions $\{u_h(\boldsymbol{\mu}_i), i = 1, \ldots, N_{\max}\}$, i. e., solutions to (1.12) or (1.17), respectively. The discrete solution vectors are stored columnwise in a snapshot matrix $\mathbb{S} \in \mathbb{R}^{N_h \times N_{\max}}$. POD compresses the data stored in $\mathbb{S}$ by computing an orthogonal matrix $\mathbb{V}$, which is a best approximation in the least-squares sense to $\mathbb{S}$. In particular, the POD solution of size N is the solution to

$$\min_{\mathbb{V} \in \mathbb{R}^{N_h \times N}} \left\| \mathbb{S} - \mathbb{V}\mathbb{V}^T \mathbb{S} \right\|_F, \tag{1.18}$$

$$\text{subject to } \mathbb{V}^T \mathbb{V} = \mathbb{I}_{N \times N}, \tag{1.19}$$

with $\| \cdot \|_F$ being the Frobenius norm and $\mathbb{I}_{N \times N}$ being the identity matrix.

There exists a solution to (1.18)–(1.19) according to the Eckardt–Young–Mirsky theorem [43], which can be computed with singular value decomposition (SVD) as

$$\mathbb{S} = \mathbb{U}\Sigma\mathbb{Z}, \tag{1.20}$$

with orthogonal matrix $\mathbb{U} \in \mathbb{R}^{N_h \times N_h}$, rectangular diagonal matrix $\Sigma \in \mathbb{R}^{N_h \times N_{\max}}$, and orthogonal matrix $\mathbb{Z} \in \mathbb{R}^{N_{\max} \times N_{\max}}$. The solution $\mathbb{V}$ is composed of the first N column vectors of $\mathbb{U}$. They are also called the *POD modes*. The diagonal entries $\{\sigma_i, i = 1, \ldots, \min(N_h, N_{\max})\}$ of Σ are nonnegative and are called *singular values*. We have

$$\min_{\mathbb{V} \in \mathbb{R}^{N_h \times N}} \left\| \mathbb{S} - \mathbb{V}\mathbb{V}^T \mathbb{S} \right\|_F = \sum_{i=N+1}^{\min(N_h, N_{\max})} \sigma_i. \tag{1.21}$$

Thus, the neglected singular values give an indication of the approximate truncation error. In practise, a high tolerance threshold like 99 % or 99.99 % is chosen and N is determined so that the sum of the first N *singular values* reaches this percentage of the sum of all *singular values*. In many applications, an exponential singular value decay can be observed, which allows to reach the tolerance with a few POD modes.

1.1.3.2 Greedy algorithm

The greedy algorithm also computes an orthogonal matrix $\mathbb{V} \in \mathbb{R}^{N_h \times N}$ to serve as a projection operator, just as in the POD case. The greedy algorithm is an iterative procedure, which enriches the snapshot space according to where an error indicator or error estimator Δ attains its maximum. Starting from a field solution at a given initial parameter value, the parameter location is sought, whose field solution is worst approximated with the initial solution. This solution is then computed and appended to the projection matrix to obtain a two-dimensional projection space. The greedy typically searches for new snapshot solutions within a discrete surrogate P of the parameter space $\mathcal{P}$. The process is repeated until a given tolerance on the error estimator is fulfilled. The error estimator is residual-based and estimates the error between a reduced-order solve for a projection space $\mathbb{V}$ and the high-fidelity solution (Section 1.1.4). The greedy algorithm is stated in pseudo-code in Algorithm 1.1.

Algorithm 1.1: The greedy algorithm.

Input: discrete surrogate P of parameter space $\mathcal{P}$, approximation tolerance `tol`,
   initial parameter $\boldsymbol{\mu}_1$

Output: projection matrix $\mathbb{V}$

  $N = 1$

  $\mathbb{V}_1 = \dfrac{\mathbf{u}_h(\boldsymbol{\mu}_1)}{\|\mathbf{u}_h(\boldsymbol{\mu}_1)\|}$

  while $\max_{\boldsymbol{\mu} \in P} \Delta(\boldsymbol{\mu}) > $ `tol` **do**

    $N = N + 1$

    $\boldsymbol{\mu}_N = \underset{\boldsymbol{\mu} \in P}{\arg\max}\, \Delta(\boldsymbol{\mu})$

    solve (1.17) at $\boldsymbol{\mu}_N$ for $\mathbf{u}_h(\boldsymbol{\mu}_N)$

    orthonormalize $\mathbf{u}_h(\boldsymbol{\mu}_N)$ with respect to $\mathbb{V}_{N-1}$ to obtain ζ_N

    append ζ_N to $\mathbb{V}_{N-1}$ to obtain $\mathbb{V}_N$

  end while

  set $\mathbb{V} = \mathbb{V}_N$

1.1.3.3 Reduced-order system

Starting from the discrete high-fidelity formulation (1.12), another Galerkin projection is invoked to arrive at the reduced-order formulation. Assume a projection space V_N is then determined through either a POD or the greedy sampling, with $\mathbb{V} \in \mathbb{R}^{N_h \times N}$ denoting a discrete basis of V_N. Thus $V_N \subset V_h$ and $\dim V_N = N$.

    The reduced-order variational formulation is to determine $u_N(\boldsymbol{\mu}) \in V_N$, such that

$$a\big(u_N(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}\big) = f(v_N; \boldsymbol{\mu}) \quad \forall v_N \in V_N. \tag{1.22}$$

Equation (1.17) is then projected onto the reduced-order space as

$$\mathbb{V}^T \mathbb{A}_h \mathbb{V} \mathbf{u}_N = \mathbb{V}^T \mathbf{f}_h. \tag{1.23}$$

The reduced system matrix $\mathbb{A}_N = \mathbb{V}^T \mathbb{A}_h \mathbb{V}$ is then a dense matrix of small size $N \times N$ as depicted in (1.24):

$$[\mathbb{A}_N] = \begin{bmatrix} \mathbb{V} \end{bmatrix}^T \begin{bmatrix} a(\varphi^1, \varphi^1) & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & a(\varphi^{N_h}, \varphi^{N_h}) \end{bmatrix} \begin{bmatrix} \mathbb{V} \end{bmatrix}. \tag{1.24}$$

The high-order solution is then approximated as

$$\mathbf{u}_h \approx \mathbb{V} \mathbf{u}_N. \tag{1.25}$$

1.1.3.4 Affine parameter dependency

Many MOR algorithms rely on an affine parameter dependency, because the affine parameter dependency provides the computational efficiency of the model reduction. Thus, it is a significant advancement from the 2000s [85] over the first use of ROMs [3, 76].

An affine parameter dependency means that the bilinear form can be expanded as

$$a(\cdot, \cdot; \boldsymbol{\mu}) = \sum_{i=1}^{Q_a} \Theta_a^i(\boldsymbol{\mu}) a_i(\cdot, \cdot), \tag{1.26}$$

and affine expansions hold as

$$f(\cdot; \boldsymbol{\mu}) = \sum_{i=1}^{Q_f} \Theta_f^i(\boldsymbol{\mu}) f_i(\cdot), \tag{1.27}$$

$$l(\cdot; \boldsymbol{\mu}) = \sum_{i=1}^{Q_l} \Theta_l^i(\boldsymbol{\mu}) l_i(\cdot), \tag{1.28}$$

with scalar-valued functions $\Theta_a^i : \mathcal{P} \to \mathbb{R}, \Theta_f^i : \mathcal{P} \to \mathbb{R}$, and $\Theta_l^i : \mathcal{P} \to \mathbb{R}$.

Correspondingly the linear system (1.17) can be expanded as

$$\left(\sum_{i=1}^{Q_a} \Theta_a^i(\boldsymbol{\mu}) \mathbb{A}_i \right) \mathbf{u}_h = \sum_{i=1}^{Q_f} \Theta_f^i(\boldsymbol{\mu}) \mathbf{f}_i, \tag{1.29}$$

as well as the reduced-order form (1.23)

$$\mathbb{V}^T \left(\sum_{i=1}^{Q_a} \Theta_a^i(\boldsymbol{\mu}) \mathbb{A}_i \right) \mathbb{V} \mathbf{u}_N = \mathbb{V}^T \sum_{i=1}^{Q_f} \Theta_f^i(\boldsymbol{\mu}) \mathbf{f}_i, \tag{1.30}$$

$$\left(\sum_{i=1}^{Q_a} \Theta_a^i(\boldsymbol{\mu}) \mathbb{V}^T \mathbb{A}_i \mathbb{V} \right) \mathbf{u}_N = \sum_{i=1}^{Q_f} \Theta_f^i(\boldsymbol{\mu}) \mathbb{V}^T \mathbf{f}_i. \tag{1.31}$$

MOR relies on an affine parameter dependency, such that all computations depending on the high-order model size can be moved into a parameter-independent offline phase, while having a fast input-output evaluation online. If the problem is not affine, an affine representation can be approximated using a technique such as the *EIM* (Section 1.3).

1.1.3.5 Affine shape parameterizations: an example

Consider heat conduction in a square domain $\Omega(x, y) = [0, 1]^2$. On the left side $x = 0$, inhomogeneous Neumann conditions, i. e., a nonzero heat flux, are imposed and on the right side $x = 1$, homogeneous Dirichlet conditions, i. e., zero temperature, are imposed. On the top and bottom sides, homogeneous Neumann conditions, i. e., a zero heat flux, are imposed. Consider two different media with different conductivities σ_1 and σ_2 occupying the subdomains $\Omega_1(\mu) = [0, \mu] \times [0, 1]$ and $\Omega_2(\mu) = [\mu, 1] \times [0, 1]$, for $\mu \in \mathcal{P} = (0, 1)$, as shown in Figure 1.1. For the sake of clarity, in the rest of this section we identify the one-dimensional parameter vector $\boldsymbol{\mu}$ with its (only) component μ, thus dropping the bold notation from the symbol.
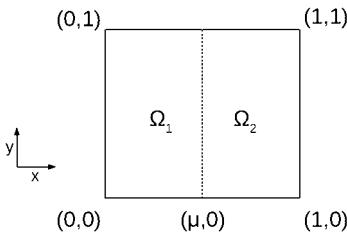
Figure 1.1: The computational domain is subdivided into two domains $\Omega = \Omega_1 \cup \Omega_2$, depending on the parameter μ. Shown here for $\mu = 0.5$.

Choosing $\overline{\mu} = 0.5$ as the reference configuration, there exist affine transformations from the reference domain to the actual domain. We have

$$T_1 : \Omega_1(\overline{\mu}) \to \Omega_1(\mu) : (\overline{x}, \overline{y}) \mapsto (2\mu\overline{x}, \overline{y}), \tag{1.32}$$

$$T_2 : \Omega_2(\overline{\mu}) \to \Omega_2(\mu) : (\overline{x}, \overline{y}) \mapsto \left((2 - 2\mu)\overline{x}, \overline{y}\right) + (2\mu - 1, 0). \tag{1.33}$$

In general, an affine transformation of a subdomain can be expressed as

$$T_k : \Omega_k(\overline{\mu}) \to \Omega_k(\mu) : \mathbf{x} \mapsto G_k(\mu)\mathbf{x} + D_k(\mu), \tag{1.34}$$

with $\mathbf{x} \in \mathbb{R}^d$, $G_k \in \mathbb{R}^{d \times d}$ and $D_k \in \mathbb{R}^d$ in $d = 2, 3$ spatial dimensions.

Thus, the bilinear form

$$a(u, v; \mu) = \int_{\Omega_1(\mu)} \sigma_1 \nabla u \cdot \nabla v d\mathbf{x} + \int_{\Omega_2(\mu)} \sigma_2 \nabla u \cdot \nabla v \, d\mathbf{x} \tag{1.35}$$

can be mapped to the reference domain with the inverse affine transformation

$$T_k^{-1} : \Omega_k(\mu) \to \Omega_k(\overline{\mu}) : \mathbf{x} \mapsto G_k^{-1}(\mu)\mathbf{x} - G_k^{-1}(\mu)D_k(\mu), \tag{1.36}$$

and integration by substitution as

$$a(u, v; \mu) = \int_{\Omega_1(\overline{\mu})} \sigma_1(\nabla u G_1^{-1}(\mu)) \cdot (G_1^{-T}(\mu)\nabla v) \det(G_1(\mu)) \, d\mathbf{x} \tag{1.37}$$

$$+ \int_{\Omega_2(\overline{\mu})} \sigma_2(\nabla u G_2^{-1}(\mu)) \cdot (G_2^{-T}(\mu)\nabla v) \det(G_2(\mu)) \, d\mathbf{x}, \tag{1.38}$$

which establishes the affine parameter dependency (1.26) by computing $\Theta_a^i(\mu)$ from the coefficients of G_1 and G_2 [85, 84, 28]. That is,

$$\int_{\Omega_1(\overline{\mu})} \sigma_1(\nabla u G_1^{-1}(\mu)) \cdot (G_1^{-T}(\mu)\nabla v) \det(G_1(\mu)) \, d\mathbf{x} \tag{1.39}$$

$$= \int_{\Omega_1(\overline{\mu})} \sigma_1((2\mu)^{-1}\partial_x u, \partial_y u) \cdot ((2\mu)^{-1}\partial_x v, \partial_y v) 2\mu \, d\mathbf{x} \tag{1.40}$$

$$= (2\mu)^{-1} \int_{\Omega_1(\overline{\mu})} \sigma_1(\partial_x u)(\partial_x v) \, d\mathbf{x} + 2\mu \int_{\Omega_1(\overline{\mu})} \sigma_1(\partial_y u)(\partial_y v) \, d\mathbf{x}, \tag{1.41}$$

and

$$\int_{\Omega_2(\overline{\mu})} \sigma_2(\nabla u G_2^{-1}(\mu)) \cdot (G_2^{-T}(\mu)\nabla v) \det(G_2(\mu)) \, d\mathbf{x} \tag{1.42}$$

$$= \int_{\Omega_2(\overline{\mu})} \sigma_2((2 - 2\mu)^{-1}\partial_x u, \partial_y u) \cdot ((2 - 2\mu)^{-1}\partial_x v, \partial_y v)(2 - 2\mu) \, d\mathbf{x} \tag{1.43}$$

$$= (2 - 2\mu)^{-1} \int_{\Omega_2(\overline{\mu})} \sigma_2(\partial_x u)(\partial_x v) \, d\mathbf{x} + (2 - 2\mu) \int_{\Omega_2(\overline{\mu})} \sigma_2(\partial_y u)(\partial_y v) \, d\mathbf{x}, \tag{1.44}$$

which establishes the affine form (1.26) with $Q_a = 4$, and

$$\Theta_a^1(\mu) = (2\mu)^{-1}, \tag{1.45}$$

$$\Theta_a^2(\mu) = 2\mu, \tag{1.46}$$

$$\Theta_a^3(\mu) = (2 - 2\mu)^{-1}, \tag{1.47}$$

$$\Theta_a^4(\mu) = 2 - 2\mu, \tag{1.48}$$

and

$$a_1(\cdot,\cdot) = \int\limits_{\Omega_1(\overline{\mu})} \sigma_1(\partial_x u)(\partial_x v)\, d\mathbf{x}, \tag{1.49}$$

$$a_2(\cdot,\cdot) = \int\limits_{\Omega_1(\overline{\mu})} \sigma_1(\partial_y u)(\partial_y v)\, d\mathbf{x}, \tag{1.50}$$

$$a_3(\cdot,\cdot) = \int\limits_{\Omega_2(\overline{\mu})} \sigma_2(\partial_x u)(\partial_x v)\, d\mathbf{x}, \tag{1.51}$$

$$a_4(\cdot,\cdot) = \int\limits_{\Omega_2(\overline{\mu})} \sigma_2(\partial_y u)(\partial_y v)\, d\mathbf{x}. \tag{1.52}$$

The second and fourth terms can be further simplified to a term depending on 2μ and a μ-independent term, but in this case it still leaves $Q_a = 4$ terms. In some cases the number of affine terms can be automatically reduced further using symbolic computations.

1.1.3.6 Offline-online decomposition

The offline-online decomposition enables the computational speedup of the ROM approach in many-query scenarios. It is also known as the offline-online paradigm, which assumes that a computation-intensive offline phase can be performed on a supercomputer, which generates all quantities depending on the large discretization size N_h. Once completed, a reduced-order solve, i. e., an online solve for a new parameter of interest, can be performed with computational cost independent of the large discretization size N_h. The online phase can thus be performed even on mobile and embedded devices (Figure 1.2). If a supercomputer is not available, this can be relaxed, however. There exist heuristic algorithms to make also the offline phase feasible on a common workstation, such that a typical scenario would be that the offline phase runs overnight and a reduced model is available the next morning.
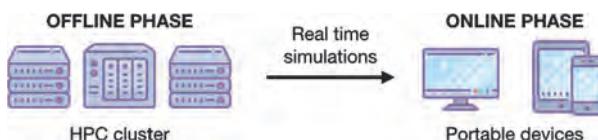

Figure 1.2: Offline-online paradigm. The complex high-fidelity simulations are carried out in high performance clusters (HPCs) for given preselected parameters. The solution snapshots can be stored and the ROM trained. Then in the offline phase the ROM provides approximated solutions at new untried parameters in real-time on simple portable devices.

Noting that the terms $\mathbb{V}^T \mathbb{A}_i \mathbb{V}$ and $\mathbb{V}^T \mathbf{f}_i$ in (1.31) are parameter-independent, they can be precomputed, prior to any ROM parameter sweep. This will store small-sized dense matrices of dimension $N \times N$. Once a reduced-order solution $\mathbf{u}_N$ is desired for a given parameter $\boldsymbol{\mu}$, the sum given in (1.31) is formed and solved for $\mathbf{u}_N$. Since this is the same as solving (1.23), the reduced-order approximation is then available as $\mathbf{u}_h \approx \mathbb{V}\mathbf{u}_N$; see (1.25).

1.1.4 Error bounds

In this section we develop effective and reliable a posteriori error estimators for the field variable or an output of interest. The use of such error bounds drives the construction of the reduced basis during the offline stage, thanks to the so-called greedy algorithm. Moreover, during the online stage, such bounds provide a certified accuracy of the proposed ROM.

Following [85], we introduce residual-based a posteriori error estimation for the elliptic case. From (1.12) and (1.22) it follows that the error $e(\boldsymbol{\mu}) = u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})$ satisfies

$$a\big(e(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}\big) = r(v_h; \boldsymbol{\mu}) \quad \forall v_h \in V_h, \tag{1.53}$$

where the residual $r(\cdot; \boldsymbol{\mu}) \in V_h'$ is defined as

$$r(v_h; \boldsymbol{\mu}) = f(v_h; \boldsymbol{\mu}) - a\big(u_N(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}\big) \quad \forall v_h \in V_h. \tag{1.54}$$

The following theorem further characterizes the relation between error and residual:

Theorem 1.4. *Under compliance assumptions, the following inequalities hold:*

$$\big\|e(\boldsymbol{\mu})\big\|_{\boldsymbol{\mu}} = \big\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\big\|_{\boldsymbol{\mu}} \leq \Delta_{en}(\boldsymbol{\mu}) = \frac{\|r(\cdot; \boldsymbol{\mu})\|_{V_h'}}{\sqrt{\alpha_h(\boldsymbol{\mu})}}, \tag{1.55}$$

$$0 \leq s_h(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu}) \leq \Delta_s(\boldsymbol{\mu}) = \frac{\|r(\cdot; \boldsymbol{\mu})\|_{V_h'}^2}{\alpha_h(\boldsymbol{\mu})}, \tag{1.56}$$

where $\|v\|_{\boldsymbol{\mu}}^2 = a(v, v; \boldsymbol{\mu})$ defines an equivalent norm to $\|v\|_{V_h}$.

Proof. The norm $\|\cdot\|_{\boldsymbol{\mu}}$ defines an equivalent norm thanks to symmetry, continuity, and coercivity of $a(\cdot, \cdot; \boldsymbol{\mu})$.

Since $e(\boldsymbol{\mu}) \in V_h$, from (1.53) with $v_h = e(\boldsymbol{\mu})$ it follows that

$$\big\|e(\boldsymbol{\mu})\big\|_{\boldsymbol{\mu}}^2 = a\big(e(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu}\big) = r\big(e(\boldsymbol{\mu}); \boldsymbol{\mu}\big) \leq \big\|r(\cdot; \boldsymbol{\mu})\big\|_{V_h'} \big\|e(\boldsymbol{\mu})\big\|_{V_h},$$

the last inequality being due to the definition of the norm in V_h'. Furthermore, due to coercivity, we have

$$\big\|e(\boldsymbol{\mu})\big\|_{\boldsymbol{\mu}}^2 = a\big(e(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu}\big) \geq \alpha(\boldsymbol{\mu}) \big\|e(\boldsymbol{\mu})\big\|_{V_h}^2.$$

Combining these two results yields (1.55).

Furthermore, since $l = f$ are linear forms,

$$s_h(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu}) = l(e(\boldsymbol{\mu}); \boldsymbol{\mu}) = f(e(\boldsymbol{\mu}); \boldsymbol{\mu}) = a(u_h(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu}). \qquad (1.57)$$

From (1.53) with $v_h := v_N \in V_N$ and (1.22) it follows that

$$a(e(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = r(v_N(\boldsymbol{\mu}); \boldsymbol{\mu}) = 0.$$

This holds in particular for $v_N = u_N(\boldsymbol{\mu})$. Moreover, due to symmetry,

$$a(u_N(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu}) = 0$$

as well. Thus, $a(u_h(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu}) = a(e(\boldsymbol{\mu}), e(\boldsymbol{\mu}); \boldsymbol{\mu})$ in (1.57), and we conclude that

$$s_h(\boldsymbol{\mu}) - s_N(\boldsymbol{\mu}) = \|e(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}^2. \qquad (1.58)$$

The upper bound in (1.56) is then a consequence of (1.55), while the lower bound trivially holds as the right-hand side of (1.58) is a nonnegative quantity. □

Offline-online decomposition is usually solicited for the a posteriori error bounds introduced by the previous theorem, for the sake of a fast computation of the right-hand side of (1.55)–(1.56). This requires the efficient evaluation of both the numerator (dual norm of the residual) and the denominator (parameterized coercivity constant). The Riesz representation theorem is employed to define the unique $\hat{r}(\boldsymbol{\mu}) \in V_h$ such that

$$(\hat{r}(\boldsymbol{\mu}), v_h)_{V_h} = r(v_h; \boldsymbol{\mu}), \quad \forall v_h \in V_h. \qquad (1.59)$$

Under affine separability assumptions (1.26)–(1.28), we have

$$r(v_h; \boldsymbol{\mu}) = \sum_{i=1}^{Q_f} \Theta_f^i(\boldsymbol{\mu}) f_i(v_h) - \sum_{n=1}^{N} \mathbf{u}_{Nn} \sum_{i=1}^{Q_a} \Theta_a^i(\boldsymbol{\mu}) a_i(\zeta^n, v_h), \quad \forall v_h \in V_h,$$

so that an affine expansion with $Q_f + N Q_a$ terms is obtained for $r(\cdot; \boldsymbol{\mu})$. Riesz representation is then invoked for

$$
\begin{aligned}
& r_1(v_h; \boldsymbol{\mu}) = f_1(v_h), && \ldots, \quad r_{Q_f}(v_h; \boldsymbol{\mu}) = f_{Q_f}(v_h), \\
& r_{Q_f+1}(v_h; \boldsymbol{\mu}) = a_1(\zeta^1, v_h), && \ldots, \quad r_{Q_f+Q_a}(v_h; \boldsymbol{\mu}) = a_{Q_a}(\zeta^1, v_h), \\
& \ldots && \\
& r_{Q_f+(N-1)Q_a+1}(v_h; \boldsymbol{\mu}) = a_1(\zeta^N, v_h), && \ldots, \quad r_{Q_f+NQ_a}(v_h; \boldsymbol{\mu}) = a_{Q_a}(\zeta^N, v_h)
\end{aligned}
$$

during the offline stage, storing the corresponding solutions to (1.59).

As concerns the evaluation of the denominator of (1.55)–(1.56), exact evaluation of $\alpha(\boldsymbol{\mu})$ is seldom employed. Instead, an offline-online decomposable lower bound is

sought. Early proposals on the topic are available in [107, 78, 106, 85, 18]. In 2007, the *successive constraint method* (SCM) was devised in [57] based on successive linear programming approximations, and subsequently extended in [26, 27, 103, 111]. Alternative methodologies based on interpolation techniques have also appeared in recent years in [54, 71, 59].

A posteriori error estimation can be derived for more general problems as well (including noncoercive linear, nonlinear, or time-dependent problems), through application of the Brezzi–Rappaz–Raviart theory. We refer to [106, 41, 109, 70, 81] for a few representative cases. To this end, extensions of SCM are discussed in [27, 55, 58, 25].

1.2 Geometrical parameterization for shapes and domains

In this section we discuss problems characterized by a geometrical parameterization. In particular, a reference domain approach is discussed, relying on a map that deforms the reference domain into the parameterized one. Indeed, while affine shape parameterization (see Section 1.1.3.5 for an example, and [85] for more details) naturally abides by the offline-online separability assumption, it often results in very limited deformation of the reference domain, or strong assumptions on the underlying shape.

Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, be the reference domain. Let $\mathcal{M}$ be a parametric shape morphing function, that is,

$$\mathcal{M}(\boldsymbol{x}; \boldsymbol{\mu}) : \mathbb{R}^d \to \mathbb{R}^d, \tag{1.60}$$

which maps the reference domain Ω into the deformed domain $\Omega(\boldsymbol{\mu})$ as $\Omega(\boldsymbol{\mu}) = \mathcal{M}(\Omega; \boldsymbol{\mu})$, where $\boldsymbol{\mu} \in \mathcal{P}$ represents the vector of the geometrical parameters. This map will change accordingly to the chosen shape morphing technique. The case of Section 1.1.3.5 is representative of an affine map $\mathcal{M}(\cdot; \boldsymbol{\mu})$. Instead, in the following we address more general (not necessarily affine) techniques such as FFD, RBF interpolation, and IDW interpolation.

From a practical point of view, we recommend the Python package called PyGeM – Python Geometrical Morphing [79], which allows an easy integration with the majority of industrial CAD files and the most common mesh files.

1.2.1 Free form deformation

Free form deformation (FFD) is a widely used parameterization and morphing technique both in academia and in industry.

For the original formulation see [94]. More recent works use FFD coupled with reduced basis methods for shape optimization and design of systems modeled by elliptic PDEs (see [63], [86], and [96]), in naval engineering for the optimization of the bulbous bow shape of cruise ships (in [37]), in the context of sailing boats in [65], and in automotive engineering in [91].

FFD can be used for both global and local deformations and it is completely independent of the geometry to morph. It acts through the displacement of a lattice of points, called FFD control points, constructed around the domain of interest. In particular it consists in three different steps, as depicted in Figure 1.3. First the physical domain Ω is mapped to $\hat{\Omega}$, the reference one, through the affine map $\boldsymbol{\psi}$. Then the lattice of control points is constructed, and the displacements of these points by the map $\hat{T}$ is what we call geometrical parameters $\boldsymbol{\mu}$. The deformation is propagated to the entire embedded body usually by using Bernstein polynomials. Finally through the inverse map $\boldsymbol{\psi}^{-1}$ we return back to the parametric physical space $\Omega(\boldsymbol{\mu})$.
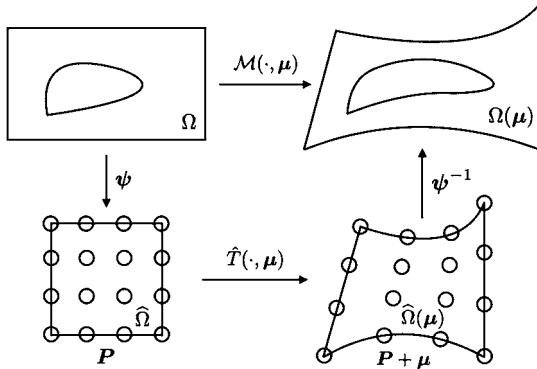
Figure 1.3: Scheme of the three maps composing the FFD map $\mathcal{M}$. In particular ψ maps the physical space to the reference one, then $\hat{T}$ deforms the entire geometry according to the displacements of the lattice control points, and finally ψ^{-1} maps back the reference domain to the physical one.

So, recalling equation (1.60), we have the explicit map $\mathcal{M}$ for the FFD, that is, the composition of the three maps presented, i. e.,

$$\mathcal{M}(\boldsymbol{x}, \boldsymbol{\mu}) = (\boldsymbol{\psi}^{-1} \circ \hat{T} \circ \boldsymbol{\psi})(\boldsymbol{x}, \boldsymbol{\mu}) = \tag{1.61}$$

$$= \boldsymbol{\psi}^{-1}\left(\sum_{l=0}^{L} \sum_{m=0}^{M} \sum_{n=0}^{N} b_{lmn}(\boldsymbol{\psi}(\boldsymbol{x})) \boldsymbol{P}^{0}_{lmn}(\boldsymbol{\mu}_{lmn}) \right) \quad \forall \boldsymbol{x} \in \Omega, \tag{1.62}$$

where b_{lmn} are Bernstein polynomials of degree l, m, n in each direction, respectively, and $\boldsymbol{P}^{0}_{lmn}(\boldsymbol{\mu}_{lmn}) = \boldsymbol{P}_{lmn} + \boldsymbol{\mu}_{lmn}$, with $\boldsymbol{P}_{lmn}$ representing the coordinates of the control point identified by the three indices l, m, n in the lattice of FFD control points. In an offline-online fashion, for a given $\boldsymbol{x}$, terms $\{b_{lmn}(\boldsymbol{\psi}(\boldsymbol{x}))\}_{l,m,n}$ can be precomputed during the offline stage, resulting in an inexpensive linear combination of $\boldsymbol{x}$-dependent precomputed quantities and $\boldsymbol{\mu}$-dependent control points locations $\{\boldsymbol{P}^{0}_{lmn}(\boldsymbol{\mu}_{lmn})\}_{l,m,n}$. The application of $\boldsymbol{\psi}^{-1}$ does not hinder such offline-online approach as $\boldsymbol{\psi}$ is affine.

We can notice that the deformation does not depend on the topology of the object to be morphed, so this technique is very versatile and nonintrusive, especially for complex geometries or in industrial contexts (see, e. g., [90, 87]).

In the case where the deformation has to satisfy some constraints, like for example continuity constraints, it is possible to increase the number of control points. Often it is the case where at the interface between the undeformed portion of the geometry and the morphed area the continuity has to be prescribed for physical reasons.

As an example, in Figure 1.4 we present an FFD of a bulbous bow, where an STL file of a complete hull is morphed continuously by the displacement of only some control points.
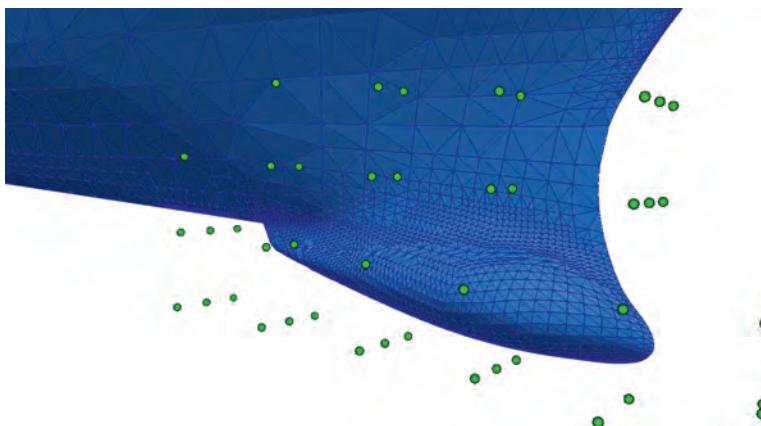
Figure 1.4: Bulbous bow deformation using FFD. In green are shown the FFD control points defining the morphing.

1.2.2 Radial basis function interpolation

Radial basis functions (RBFs) represent a powerful tool for nonlinear multivariate approximation, interpolation between nonconforming meshes ([40]), and shape parameterization due to their approximation properties [15].

An RBF is any smooth real-valued function $\widetilde{\varphi} : \mathbb{R}^d \to \mathbb{R}$ such that $\varphi : \mathbb{R}^+ \to \mathbb{R}$ exists and $\widetilde{\varphi}(\boldsymbol{x}) = \varphi(\|\boldsymbol{x}\|)$, where $\| \cdot \|$ indicates the Euclidean norm in $\mathbb{R}^d$. The most widespread RBFs are the following:

– Gaussian splines ([15]) defined as

$$\varphi(\|\boldsymbol{x}\|) = e^{-\|\boldsymbol{x}\|^2/R};$$

– thin plate splines ([42]) defined as

$$\varphi(\|\boldsymbol{x}\|) = \left(\frac{\|\boldsymbol{x}\|}{R} \right)^2 \ln\left(\frac{\|\boldsymbol{x}\|}{R} \right);$$

– Beckert and Wendland C^2-basis ([11]) defined as

$$\varphi(\|\boldsymbol{x}\|) = \left(1 - \frac{\|\boldsymbol{x}\|}{R}\right)^4_+ \left(4\frac{\|\boldsymbol{x}\|}{R} + 1\right);$$

– multiquadratic biharmonic splines ([92]) defined as

$$\varphi(\|\boldsymbol{x}\|) = \sqrt{\|\boldsymbol{x}\|^2 + R^2};$$

– inverted multiquadratic biharmonic splines ([15]) defined as

$$\varphi(\|\boldsymbol{x}\|) = \frac{1}{\sqrt{\|\boldsymbol{x}\|^2 + R^2}};$$

where $R > 0$ is a given radius and the subscript $_+$ indicates the positive part.

Following [75, 72], given $\mathcal{N}_C$ control points situated on the surface of the body to morph, we can generate a deformation by moving some of these points and imposing the new surface which interpolates them. The displacements of the control points represent the geometrical parameters $\boldsymbol{\mu}$.

We can now define the map $\mathcal{M}$ in equation (1.60) for the RBF interpolation technique, that is,

$$\mathcal{M}(\boldsymbol{x}; \boldsymbol{\mu}) = q(\boldsymbol{x}; \boldsymbol{\mu}) + \sum_{i=1}^{\mathcal{N}_C} \gamma_i(\boldsymbol{\mu})\, \varphi(\|\boldsymbol{x} - \boldsymbol{x}_{C_i}\|), \tag{1.63}$$

where $q(\boldsymbol{x}; \boldsymbol{\mu})$ is a polynomial term, generally of degree 1, $\gamma_i(\boldsymbol{\mu})$ is the weight associated to the basis function φ_i, $\{\boldsymbol{x}_{C_i}\}_{i=1}^{\mathcal{N}_C}$ are control points selected by the user (denoted by spherical green markers in Figure 1.5), and $\boldsymbol{x} \in \Omega$. We underline that in the three-dimensional case (1.63) has $d \times \mathcal{N}_C + d + d^2$ unknowns, which are $d \times \mathcal{N}_C$ for γ_i and $d + d^2$ for the polynomial term $q(\boldsymbol{x}; \boldsymbol{\mu}) = c(\boldsymbol{\mu}) + \mathbf{Q}(\boldsymbol{\mu})\boldsymbol{x}$. To this end we impose the interpolatory constraint

$$\mathcal{M}(\boldsymbol{x}_{C_i}; \boldsymbol{\mu}) = \boldsymbol{y}_{C_i}(\boldsymbol{\mu}) \quad \forall i \in \{1, \dots, \mathcal{N}_C\}, \tag{1.64}$$

where $\boldsymbol{y}_{C_i}$ are the deformed control points obtained applying the displacement $\boldsymbol{\mu}$ to $\boldsymbol{x}_{C_i}$, in particular

$$\boldsymbol{x}_C = [\boldsymbol{x}_{C_1}, \dots, \boldsymbol{x}_{C_{\mathcal{N}_C}}] \in \mathbb{R}^{\mathcal{N}_C \times d}, \tag{1.65}$$

$$\boldsymbol{y}_C(\boldsymbol{\mu}) = [\boldsymbol{y}_{C_1}(\boldsymbol{\mu}), \dots, \boldsymbol{y}_{C_{\mathcal{N}_C}}(\boldsymbol{\mu})] \in \mathbb{R}^{\mathcal{N}_C \times d}. \tag{1.66}$$

For the remaining $d + d^2$ unknowns, due to the presence of the polynomial term, we complete the system with additional constraints that represent the conservation of the total force and momentum [15, 75] as follows:

$$\sum_{i=1}^{\mathcal{N}_C} \gamma_i(\boldsymbol{\mu}) = 0, \tag{1.67}$$
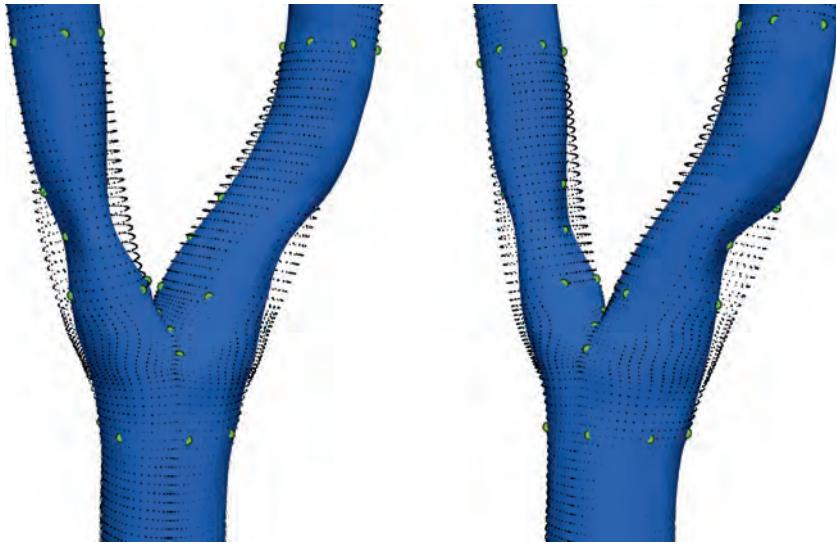
Figure 1.5: Two different views of the same deformed carotid artery model using the RBF interpolation technique. The green dots indicate the RBF control points that define the morphing. The black small points highlight the original undeformed geometry. The occlusion of the two branches is achieved through a displacement along the normal direction with respect to the carotid surface of the control points after the bifurcation.

$$\sum_{i=1}^{\mathcal{N}_C} \gamma_i(\boldsymbol{\mu})[\boldsymbol{x}_{C_i}]_1 = 0, \ldots \sum_{i=1}^{\mathcal{N}_C} \gamma_i(\boldsymbol{\mu})[\boldsymbol{x}_{C_i}]_d = 0, \tag{1.68}$$

where the notation $[\boldsymbol{x}]_d$ denotes the d-th component of the vector $\boldsymbol{x}$.

Following an offline-online strategy, for a given $\boldsymbol{x}$, evaluation of $\varphi(\|\boldsymbol{x} - \boldsymbol{x}_{C_i}\|)$, $i = 1, \ldots, \mathcal{N}_C$, can be precomputed in the offline stage. Further online effort is only required for (i) given $\boldsymbol{\mu}$, solve a $d \times \mathcal{N}_C + d + d^2$ linear system, and (ii) given $\boldsymbol{\mu}$ and $\boldsymbol{x}$, perform linear combinations and the matrix vector product in (1.63) employing either precomputed quantities or coefficients from (i).

1.2.3 Inverse distance weighting interpolation

The Inverse distance weighting (IDW) interpolation method has been proposed in [95] to deal with interpolation of scattered data. We follow [108, 49, 9] for its presentation and the application of IDW to shape parameterization.

As in the previous section, let $\{\boldsymbol{x}_{C_k}\}_{k=1}^{\mathcal{N}_c} \subset \mathbb{R}^d$ be a set of control points. The IDW interpolant $\Pi_{\text{IDW}}(f)$ of a scalar function $f : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$\Pi_{\text{IDW}}(f)(\boldsymbol{x}) = \sum_{k=1}^{\mathcal{N}_c} w_k(\boldsymbol{x}) f(\boldsymbol{x}_{C_k}), \quad \boldsymbol{x} \in \Omega, \tag{1.69}$$

where the weight functions $w_k : \Omega \to \mathbb{R}$, for $k = 1, \ldots, \mathcal{N}_c$ are given by

$$w_k(\boldsymbol{x}) = \begin{cases} \frac{\|\boldsymbol{x} - \boldsymbol{x}_{C_k}\|^{-s}}{\sum_{j=1}^{\mathcal{N}_c} \|\boldsymbol{x} - \boldsymbol{x}_{C_j}\|^{-s}} & \text{if } \boldsymbol{x} \neq \boldsymbol{x}_{C_k}, \\ 1 & \text{if } \boldsymbol{x} = \boldsymbol{x}_{C_k}, \\ 0 & \text{otherwise,} \end{cases} \tag{1.70}$$

where s is a positive integer, modeling the assumption that the influence of the k-th control point $\boldsymbol{x}_{C_k}$ on $\boldsymbol{x}$ diminishes with rate $-s$ as the distance between $\boldsymbol{x}$ and $\boldsymbol{x}_{C_k}$ increases. IDW interpolation trivially extends to vector functions $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d$ by application to each component $f_1, \ldots, f_d$, where the weight functions $w_k : \Omega \to \mathbb{R}$ do not depend on the specific component.

In the case of IDW shape parameterization, for any given $\boldsymbol{\mu}$, the deformed position of the control points $\{\boldsymbol{x}_{C_k}\}_{k=1}^{\mathcal{N}_c}$ is supposed to be known, and equal to $\boldsymbol{y}_{C_k}(\boldsymbol{\mu}) := \boldsymbol{f}(\boldsymbol{x}_{C_k})$ for $k = 1, \ldots, \mathcal{N}_c$. We remark that the analytic expression of $\boldsymbol{f}$ is not known, but only its action through $\{\boldsymbol{x}_{C_k}\}_{k=1}^{\mathcal{N}_c}$. This is indeed the minimum requirement to properly define (1.69). The deformation map is therefore

$$\mathcal{M}(\boldsymbol{x}; \boldsymbol{\mu}) = \sum_{k=1}^{\mathcal{N}_c} w_k(\boldsymbol{x}) \, \boldsymbol{y}_{C_k}(\boldsymbol{\mu}) \quad \forall \boldsymbol{x} \in \Omega.$$

In an offline-online separation effort, efficient deformation can be obtained by noting that the $\boldsymbol{\mu}$-dependent part is decoupled from the $\boldsymbol{x}$-dependent weight function $w_k(\boldsymbol{x})$. Thus, for any $\boldsymbol{x}$, weight terms can be precomputed once and for all and stored. The online cost of the evaluation of $\mathcal{M}(\boldsymbol{x}; \boldsymbol{\mu})$ thus requires an inexpensive linear combination of $\boldsymbol{x}$-dependent precomputed quantities and $\boldsymbol{\mu}$-dependent control point locations. We remark that, in contrast, the RBF approach (even though still based on interpolation) required a further solution of linear system of size $d \times \mathcal{N}_C + d + d^2$.

Application in the context of fluid–structure interaction problems between a wing (structure) and surrounding air (fluid) is shown in Figure 1.6. The IDW deformation of
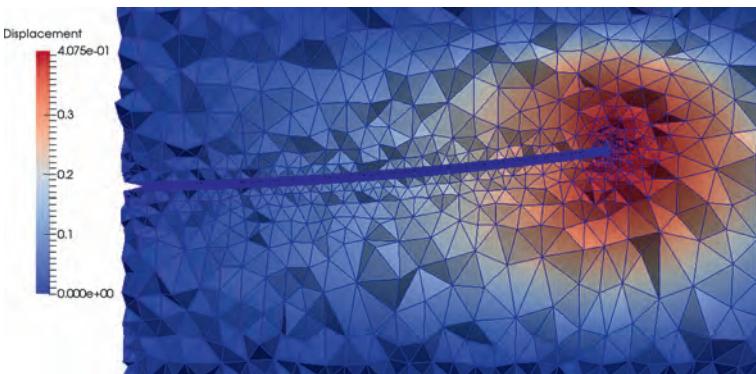

Figure 1.6: Deformation of the fluid mesh of a fluid–structure interaction problem by IDW.

the fluid mesh resulting from a vertical displacement of the tip of the wing is depicted; the structural mesh is omitted from the picture. We refer to [9] for more details.

1.3 Beyond affinity assumptions: parametric interpolation

We describe here several options to deal with cases when an exact affine decomposition of the discretized differential operators, right-hand sides, or outputs of interest does not exist. The section begins with a brief overview concerning the description of general nonaffine problems in Section 1.3.1 and later we describe the so-called EIM family of algorithms. This methodology becomes particularly useful to obtain an efficient offline-online splitting also in cases with nonlinearities and nonaffine parameterization. We provide a full description of the different alternatives, starting from its standard continuous version (EIM), and presenting also its discrete (DEIM) and matrix (M-DEIM) variants. The methodologies are tested for both nonaffine and nonlinear problems. In Section 1.3.2 we explain in detail the basics of the EIM. In Section 1.3.3 we introduce the discrete variant of the EIM at both matrix and vector level and we mention further options to obtain an approximate affine expansion. In Section 1.3.5 we present two examples using the EIM (Section 1.3.5.1) and the M-DEIM algorithm to deal with both nonaffinity and nonlinearity (Section 1.3.5.2).

1.3.1 Nonaffine problems

As already discussed in Section 1.1.3.4, the existence of an affine decomposition of the linear and bilinear forms of the considered problem is crucial in order to obtain a computationally efficient framework (see (1.26)–(1.28)).

This assumption fails to be true in several situations. Such situations occur for example in case of problems with nonaffine parametric dependency, in cases with nonlinear differential operators, and in cases dealing with the nonaffine geometrical parameterizations introduced in Section 1.2.

In fact, in these situations, the differential operators, the right-hand sides, or the outputs of interest cannot be directly written using an exact affine decomposition and we have therefore to rely on an approximate affine decomposition. The EIM is one of the key instruments to recover an approximate affine decomposition.

The EIM is a general tool for the approximation of parameterized or nonlinear functions by a sum of affine terms. In the following expression we report an example for a generic parameterized function f:

$$f(\boldsymbol{x};\boldsymbol{\mu}) \approx \sum_{q=1}^{Q} c_q(\boldsymbol{\mu}) h_q(\boldsymbol{x}). \tag{1.71}$$

The EIM has been firstly proposed in [10] to deal with nonaffine problems in the context of reduced basis methods and later applied to ROM in [53]. In [69] it has been extended to a general context, and a slightly different variant of EIM, DEIM, has been firstly proposed in [22, 23]. For more details on the a posteriori error analysis the interested reader may see [53, 44, 24] while for an extension to *hp*-adaptive EIM we refer to [45]. A generalization of the EIM family of algorithms has been proposed in [68, 24, 67] while a nonintrusive EIM technique is presented in [21] and an extension with special focus on high-dimensional parameter spaces is given in [56].

1.3.2 The empirical interpolation method

The EIM is a general method to approximate a parameterized function $f(\boldsymbol{x}; \boldsymbol{\mu}) : \Omega \times \mathcal{P}_{\text{EIM}} \to \mathbb{R}$ by a linear combination of Q precomputed basis functions in the case where each function $f_{\mu} := (\cdot; \boldsymbol{\mu})$ belongs to some Banach space $\mathcal{X}_{\Omega}$. In what follows $\boldsymbol{\mu} \in \mathcal{P}_{\text{EIM}}$ is the parameter vector and $\mathcal{P}_{\text{EIM}}$ is the parameter space. The EIM approximation is based on an interpolation operator I_Q that interpolates the given function f_{μ} in a set of interpolation points $\{\boldsymbol{x}_i\}_{i=1}^{Q} \in \Omega$. The interpolant function is constructed as a linear combination of hierarchically chosen basis functions $\{h_i\}_{i=1}^{Q} \in \mathbb{V}_{\text{EIM}}$, where $\mathbb{V}_{\text{EIM}}$ is an approximation of the function space $\mathcal{U}$ that contains f, i. e., $\mathbb{V}_{\text{EIM}} \subseteq \mathcal{U}$. On the contrary to other interpolation methods, that usually work with generic and multipurpose basis functions such as polynomial functions, the EIM works with problem-specific basis functions with global support and selected hierarchically. The interpolant function can be then expressed by

$$\text{I}_Q[f_{\mu}](\boldsymbol{x}) = \sum_{q=1}^{Q} c_q(\boldsymbol{\mu}) h_q(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega, \quad \boldsymbol{\mu} \in \mathcal{P}_{\text{EIM}}, \tag{1.72}$$

where c_q are parameter-dependent coefficients. Once the basis functions $h_q(\boldsymbol{x})$ are set, the problem of finding the coefficients $c_q(\boldsymbol{\mu})$ is solved imposing the interpolation condition, i. e.,

$$\text{I}_Q[f_{\mu}](\boldsymbol{x}_q) = \sum_{q=1}^{Q} c_q(\boldsymbol{\mu}) h_q(\boldsymbol{x}_q) = f_{\mu}(\boldsymbol{x}_q), \quad q = 1, \ldots, Q. \tag{1.73}$$

The above problem can be recast in matrix form as $\boldsymbol{T} \boldsymbol{c}_{\mu} = \boldsymbol{f}_{\mu}$ with

$$(\boldsymbol{T})_{ij} = h_j(\boldsymbol{x}_i), \quad (\boldsymbol{c}_{\mu})_j = c_j(\boldsymbol{\mu}), \quad (\boldsymbol{f}(\boldsymbol{\mu}))_j = f(\boldsymbol{x}_i; \boldsymbol{\mu}), \quad i, j = 1, \ldots, Q. \tag{1.74}$$

This problem can be easily solved given the fact that the basis functions $h_q(\boldsymbol{x})$ and the interpolation points $\boldsymbol{x}_q$ are known and that the matrix $\boldsymbol{T}$ is invertible.

The selection of the basis functions $\{h_q\}_{q=1}^{Q}$ and of the interpolation points $\{\boldsymbol{x}_q\}_{q=1}^{Q}$, which are defined by a linear combination of selected function realizations $\{f_{\mu_i}\}_{i=1}^{Q}$,

is done following a greedy approach similar to the one presented in Section 1.1.3.2 (Algorithm 1.2). The procedure provides also a set of sample points $\{\boldsymbol{\mu}_q\}_{q=1}^{Q}$ that are required for the construction of the basis functions.

Since the basis functions are defined as linear combinations of the function realizations inside the parameter space, in order to approximate the function f with a relatively small number of basis functions h_q, the manifold

$$\mathcal{M}_{\mathrm{EIM}} = \{f(\boldsymbol{x};\boldsymbol{\mu}) \mid \boldsymbol{\mu} \in \mathcal{P}_{\mathrm{EIM}}\} \tag{1.75}$$

must have a small Kolmogorov N-width [61].

Once a proper norm on Ω has been defined, where we consider $L^p(\Omega)$-norms for $1 \leq p \leq \infty$, the procedure starts with the selection of the first parameter sample, which is computed as

$$\boldsymbol{\mu}_1 = \arg\sup_{\boldsymbol{\mu}\in\mathcal{P}_{\mathrm{EIM}}} \|f_\mu(\boldsymbol{x})\|_{L^p(\Omega)},$$

while the first interpolation point is computed as

$$\boldsymbol{x}_1 = \arg\sup_{\boldsymbol{x}\in\Omega} |f_{\mu_1}(\boldsymbol{x})|.$$

The first basis function and the interpolation operator at this stage are then defined as

$$h_1(x) = \frac{f_{\mu_1}(\boldsymbol{x})}{f_{\mu_1}(\boldsymbol{x}_1)}, \quad \mathrm{I}_1[f_\mu](\boldsymbol{x}) = f(\boldsymbol{x}_1;\boldsymbol{\mu})h_1(\boldsymbol{x}).$$

At the subsequent steps, the next basis function is selected as the one that is the worse approximated by the current interpolation operator and using a similar concept the interpolation point, often referred as *magic point*, is the one where the interpolation error is maximized. In mathematical terms, at the step k, the sample point is selected as the one that maximizes the error between the function f and the interpolation operator computed at the previous step $\mathrm{I}_{k-1}[f]$:

$$\boldsymbol{\mu}_k = \arg\sup_{\boldsymbol{\mu}\in\mathcal{P}_{\mathrm{EIM}}} \|f_\mu(\boldsymbol{x}) - \mathrm{I}_{k-1}[f_\mu](\boldsymbol{x})\|_{L^p(\Omega)}.$$

Once the sample point has been determined, the interpolation point is selected, in a similar fashion, as the point inside the domain that maximizes the error between the function f and the interpolation operator:

$$\boldsymbol{x}_k = \arg\sup_{\boldsymbol{x}\in\Omega} |f_{\mu_k}(\boldsymbol{x}) - \mathrm{I}_{k-1}[f_{\mu_k}](\boldsymbol{x})|.$$

The next basis function is defined similarly to the first one with

$$h_k(\boldsymbol{x}) = \frac{f_{\mu_k}(\boldsymbol{x}) - \mathrm{I}_{k-1}[f_{\mu_k}](\boldsymbol{x})}{f_{\mu_k}(\boldsymbol{x}_k) - \mathrm{I}_{k-1}[f_{\mu_k}](\boldsymbol{x}_k;)}.$$

Algorithm 1.2: The EIM algorithm – continuous version.

Input: set of parameterized functions $f_\mu : \Omega \to \mathbb{R}$, tolerance tol and maximum
   number of basis functions $N_{\max}$, p order of the chosen p-norm.
Output: basis functions $\{h_1, \ldots, h_Q\}$, interpolation points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_Q\}$;
   $k = 1$; $\varepsilon = \text{tol} + 1$;
   while $k < N_{\max}$ and $\varepsilon > \text{tol}$ **do**
      Pick the sample point:
      $\boldsymbol{\mu}_k = \arg\sup_{\boldsymbol{\mu} \in \mathcal{P}_{\text{EIM}}} \|f_\mu(\boldsymbol{x}) - I_{k-1}[f_\mu](\boldsymbol{x})\|_{L^p(\Omega)}$;
      Compute the corresponding interpolation point:
      $\boldsymbol{x}_k = \arg\sup_{\boldsymbol{x} \in \Omega} |f_{\mu_k}(\boldsymbol{x}) - I_{k-1}[f_{\mu_k}](\boldsymbol{x})|$;
      Define the next basis function:
      $h_k(\boldsymbol{x}) = \frac{f_{\mu_k}(\boldsymbol{x}) - I_{k-1}[f_{\mu_k}](\boldsymbol{x})}{f_{\mu_k}(\boldsymbol{x}_k) - I_{k-1}[f_{\mu_k}](\boldsymbol{x}_k)}$;
      Compute the error level:
      $\varepsilon = \|\varepsilon_p\|_{L^\infty}$ with $\varepsilon_p(\boldsymbol{\mu}) = \|f_\mu(\boldsymbol{x}) - I_{k-1}[f_\mu](\boldsymbol{x})\|_{L^p(\Omega)}$;
      $k = k + 1$;
   end while

The procedure is repeated until a certain tolerance tol is reached or a maximum number of terms $N_{\max}$ are computed (Algorithm 1.2). We remark that by construction the basis functions $\{h_1, \ldots, h_Q\}$ and the functions $\{f_{\mu_1}, \ldots, f_{\mu_Q}\}$ span the same space $\mathbb{V}_{\text{EIM}}$:

$$\mathbb{V}_{\text{EIM}} = \text{span}\{h_1, \ldots, h_Q\} = \text{span}\{f_{\mu_1}, \ldots, f_{\mu_Q}\}.$$

However, the former are preferred for the following reasons (for more details and for the mathematical proofs we refer to [10]):
- they are linearly independent,
- $h_i(\boldsymbol{x}_i) = 1$ for $1 \le i \le Q$ and $h_i(\boldsymbol{x}_j) = 0$ for $1 \le i \le j \le Q$,
- they make the interpolation matrix $\boldsymbol{T}$ of equation (1.74) to be lower triangular and with diagonal elements equal to unity and therefore the matrix is invertible.

The third point implies that the interpolation problem is well-posed.

1.3.2.1 Error analysis

Dealing with interpolation procedures, the error analysis usually involves a Lebesgue constant. In particular, in the case one is using the $L^\infty(\Omega)$-norm the error analysis involves the computation of the Lebesgue constant $\Lambda_q = \sup_{\boldsymbol{x} \in \Omega} \sum_{i=1}^{q} |L_i(\boldsymbol{x})|$ being $L_i \in \mathbb{V}_{\text{EIM}}$ a Lagrange function that satisfies $L_i(x_j) = \delta_{ij}$. It can be proved that the interpola-

tion error is bounded by the following expression [10]:

$$\|f_\mu - \mathrm{I_q}[f_\mu]\|_{L^\infty(\Omega)} \le (1 + \Lambda_q) \inf_{v_q \in \mathbb{V}_{\mathrm{EIM}}} \|f_\mu - v_q\|_{L^\infty(\Omega)}. \tag{1.76}$$

An upper bound for the Lebesgue constant, which in practice has been demonstrated to be very conservative [10], can be computed as

$$\Lambda_q \le 2^q - 1.$$

For more details concerning the estimates of the interpolation error we refer to [10, 69].

1.3.2.2 Practical implementation of the algorithm

Practically, finding the maximum of Algorithm 1.2 is usually not feasible and therefore the continuous version must be transformed into a computable one.

This is done selecting a finite-dimensional set of training points in the parameter space $\{\boldsymbol{\mu}_i\}_{i=1}^N \in \mathcal{P}_{\mathrm{EIM}}^{\mathrm{train}} \subset \mathcal{P}_{\mathrm{EIM}}$ and in the physical domain $\{\boldsymbol{x}_i\}_{i=1}^M \in \Omega_h \subset \Omega$. For this reason we introduce the vector $\boldsymbol{f} : \Omega_h \times \mathcal{P}_{\mathrm{EIM}}^{\mathrm{train}} \to \mathbb{R}^M$ which consists of a discrete representation of the function f:

$$(\boldsymbol{f}_\mu)_i = f_\mu(\boldsymbol{x}_i), \quad i = 1, \dots, M. \tag{1.77}$$

We also define the matrix $\boldsymbol{H}_Q \in \mathbb{R}^{M \times Q}$, which is defined by the discrete basis functions $H_Q = [\boldsymbol{h}_1, \dots, \boldsymbol{h}_Q]$ and the interpolation index vector $\boldsymbol{i}_Q = (i_1, \dots, i_Q)$. The discrete interpolation operator of order Q for the vector function $\boldsymbol{f}$ is then defined by

$$\mathrm{I}_Q[\boldsymbol{f}_\mu] = \boldsymbol{H}_Q \boldsymbol{a}_{f_\mu}, \tag{1.78}$$

where the coefficients $\boldsymbol{a}_{f_\mu}$ are defined such that $\boldsymbol{T} \boldsymbol{a}_{f_\mu} = \boldsymbol{f}_\mu$, where

$$\boldsymbol{T}_{kq} = (\boldsymbol{H}_Q)_{i_k q}, \quad k, q = 1, \dots, Q. \tag{1.79}$$

The implementation of the algorithm is similar to the continuous version and is reported in Algorithm 1.3. In the algorithm we use the notation $\boldsymbol{F}_{:,j}$ to denote the j-th column of the matrix $\boldsymbol{F}$, where $\boldsymbol{F} \in \mathbb{R}^{M \times N}$ is a matrix containing vector representations of the function $\boldsymbol{f}$:

$$(\boldsymbol{F})_{ij} = f(\boldsymbol{x}_i; \boldsymbol{\mu}_j). \tag{1.80}$$

Once the basis and the interpolation indices are defined, during the online stage it is required to make a pointwise evaluation of the f function in the points defined by the interpolation indices.

Algorithm 1.3: The EIM algorithm – practical implementation.

Input: set of parameter samples $\{\boldsymbol{\mu}_i\}_{i=1}^M \in \mathcal{P}_{\text{EIM}}^{\text{train}} \subset \mathcal{P}_{\text{EIM}}$, set of discrete points $\{\boldsymbol{x}_i\}_{i=1}^N \in \Omega^{\text{train}}$, tolerance tol, maximum number of basis functions $N_{\max}$, p order of the chosen p-norm.

Output: basis function matrix $\boldsymbol{H}_Q = \{\boldsymbol{h}_1, \ldots, \boldsymbol{h}_Q\}$, interpolation index vector $\boldsymbol{i}_Q = \{i_1, \ldots, i_Q\}$;

Assemble the matrix:

$(\boldsymbol{F})_{ij} = f(\boldsymbol{x}_i; \boldsymbol{\mu}_j), \quad i = 1, \ldots, M, \quad j = 1, \ldots, N;$

$k = 1, \quad \varepsilon = \text{tol} + 1;$

while $k < N_{\max}$ and $\varepsilon > \text{tol}$ **do**

   Pick the sample index:

   $j_k = \underset{j=1,\ldots,M}{\arg\max} \|\boldsymbol{F}_{:,j} - \mathrm{I}_{k-1}[\boldsymbol{F}_{:,j}]\|_{L^p};$

   and compute the interpolation point index:

   $i_k = \underset{i=1,\ldots,N}{\arg\max} |\boldsymbol{F}_{i,j_k} - (\mathrm{I}_{k-1}[\boldsymbol{F}_{:,j_k}])_i|;$

   define the next approximation column:

   $\boldsymbol{h}_k = \dfrac{\boldsymbol{F}_{:,j_k} - \mathrm{I}_{k-1}[\boldsymbol{F}_{:,j_k}]}{\boldsymbol{F}_{i_k,j_k} - (\mathrm{I}_{k-1}[\boldsymbol{F}_{:,j_k}])_{i_k}}$

   define the error level:

   $\varepsilon = \underset{j=1,\ldots,M}{\max} \|\boldsymbol{F}_{:,j} - \mathrm{I}_{k-1}[\boldsymbol{F}_{:,j}]\|_{L^p}$

   $k = k + 1$

end while

1.3.3 The discrete empirical interpolation method

A computable version of EIM is the so-called DEIM, introduced in [23]. We provide here an example as a special case of EIM where appropriate basis functions are already available [69]. In our example here DEIM basis functions are computed relying on a POD procedure which is performed on a set of discrete snapshots of the parameterized function $\{\boldsymbol{f}_i\}_{i=1}^M$. Each snapshot $\boldsymbol{f}_i$ is already considered in discrete form in a prescribed set of points $\{\boldsymbol{x}_i\}_{i=1}^{N_h}$. The procedure, which is described in detail in Algorithm 1.4, can be summarized into the following steps:

1. Construct the DEIM basis functions using a POD procedure on a set of previously computed snapshots:

$$\boldsymbol{H}_M = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_M] = \text{POD}(\boldsymbol{f}(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M)). \tag{1.81}$$

2. Given a prescribed tolerance tol, determine the indices $\boldsymbol{i}_Q$ and truncate the dimension of the POD space using an iterative greedy approach (Algorithm 1.4).

Algorithm 1.4: The DEIM procedure.

Input: snapshots matrix $S = [f(\mu_1), \ldots, f(\mu_M)]$, tolerance tol.

Output: DEIM basis functions $H_Q = [h_1, \ldots, h_Q]$, interpolation indices
$\quad i_Q = [i_1, \ldots, i_Q]$.
$\quad$ compute the DEIM modes $H_M = [h_1, \ldots, h_M] = \text{POD}(S)$
$\quad \varepsilon = \text{tol} + 1, \quad k = 1$
$\quad i_1 = \underset{j=1,N_h}{\arg\max} |(h_1)_j|$
$\quad H_Q = [h_1], \quad i_Q = [i_1], \quad P = [e_{i_1}]$
$\quad$ **while** $\varepsilon > \text{tol}$ **do**
$\quad\quad k = k + 1$
$\quad\quad$ Solve $(P^T H_Q)c = P^T h_k$
$\quad\quad r = h_k - H_Q c$
$\quad\quad i_k = \underset{j=1,N_h}{\arg\max} |(r)_j|$
$\quad\quad H_Q = [H_Q, h_k], \quad P = [P, e_{i_k}], \quad i_Q = [i_Q, i_k]$
$\quad$ **end while**

In Algorithm 1.4, with the term e_{i_k}, we identify a vector of dimension N_h where the only nonnull element is equal to 1 and is located at the index i_k:

$$(e_{i_k})_j = 1 \text{ for } j = i_k, \quad (e_{i_k})_j = 0 \text{ for } j \neq i_k.$$

During the online stage, when a new value of the parameter μ needs to be tested, it is required to compute the function $f(\mu)$ only in the location identified by the indices i_Q. Therefore, the nonlinear function needs to be evaluated only in a relatively small number of points which is usually much smaller with respect to the total number of degrees of freedom used to discretize the domain.

1.3.4 Further options

Apart from the EIM and the DEIM algorithm, further options are available. We mention here the matrix version of the DEIM algorithm (M-DEIM) [14], which extends the DEIM also to the case of parameterized or nonlinear matrices, the generalized EIM (GEIM) [68], and the gappy POD [16, 20].

$\quad$ The M-DEIM is used to perform MOR on discretized differential operators characterized by nonlinearity or nonaffinity with respect to the parameter vector μ. The algorithm is similar to the one in Algorithm 1.4 with the only difference that a vectorized version of the matrices is used to describe snapshots and POD modes. In Section 1.3.5 we will provide an example dealing with both issues.

The gappy POD generalizes the interpolation condition to the case where the number of basis functions is smaller than the number of interpolation indices, i. e., $\mathrm{card}(\boldsymbol{H}_Q) < \mathrm{card}(\boldsymbol{i}_Q)$. In this case the interpolation condition is substituted by a least-squares regression.

The GEIM replaces the EIM requirement of a pointwise interpolation condition by the following statement:

$$\sigma_j(\mathrm{I}_Q(f(\boldsymbol{\mu}))) = \sigma_j(f(\boldsymbol{\mu})), \quad j = 1, \dots, Q, \tag{1.82}$$

where σ_j are a set of "well-chosen" linear functionals. For more details and for convergence analysis of the present method we refer to [67].

1.3.5 Some examples

In the previous sections we have presented the EIM family of algorithms and we have illustrated how it is possible to recover an approximate affine expansion of the discretized differential operators. In this section we show in more detail two examples on the practical application of the EIM and the M-DEIM algorithm.

1.3.5.1 A heat transfer problem with a parameterized nonaffine dependency forcing term

In this example we illustrate the application of the computable version of the EIM on a steady-state heat conduction problem in a two-dimensional square domain $\Omega = [-1, 1]^2$ with a parameterized forcing term $g(\boldsymbol{\mu})$ and homogeneous Dirichlet boundary conditions on the boundary $\partial\Omega$. The problem is described by the following equation:

$$\begin{cases} -\alpha_t \Delta\theta = g(\boldsymbol{\mu}), & \text{in } \Omega, \\ \theta = 0, & \text{on } \partial\Omega, \end{cases} \tag{1.83}$$

where θ is the temperature field, α_t is the thermal conductivity coefficient, and $g(\boldsymbol{\mu})$ is the parameterized forcing term which is described by the following expression:

$$g(\boldsymbol{x}; \boldsymbol{\mu}) = e^{-2(x_1 - \mu_1)^2 - 2(x_2 - \mu_2)^2}, \tag{1.84}$$

where μ_1 and μ_2 are the first and second components of the parameter vector and x_1 and x_2 are the horizontal and vertical coordinates, respectively. Let V be a Hilbert space. The weak formulation of the problem can be written as follows: Find $\theta \in V$ such that

$$a(\theta(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \quad \forall v \in V, \tag{1.85}$$

where the parameterized bilinear and linear forms are expressed by

$$a(\theta, v; \boldsymbol{\mu}) = \int_{\Omega} \nabla\theta \cdot \nabla v \, d\boldsymbol{x}, \quad f(v; \boldsymbol{\mu}) = \int_{\Omega} g(\boldsymbol{x}; \boldsymbol{\mu}) v \, d\boldsymbol{x}. \tag{1.86}$$

In the above expressions, the bilinear form $a(\cdot, \cdot; \boldsymbol{\mu}) : V \times V \to \mathbb{R}$ is trivially affine while for the linear form $f(\cdot; \boldsymbol{\mu}) : V \to \mathbb{R}$ we have to rely on an approximate affine expansion using the EIM. The problem is discretized using triangular linear finite elements according to the mesh reported on the left side of Figure 1.9.

In the present case it is not possible to write an exact affine decomposition of the linear form f; we rely therefore on the computable version of the EIM of Algorithm 1.3 in order to recover an approximate affine expansion.

The function $g(\boldsymbol{x}; \boldsymbol{\mu})$ is parameterized with the parameter vector $\boldsymbol{\mu} = (\mu_1, \mu_2) \in \mathcal{P}_{\text{EIM}} = [-1, 1]^2$, which describes the position of the center of the Gaussian function. The conductivity coefficient α_t is fixed constant and equal to 1. The testing set for the implementation of the algorithm $\{\boldsymbol{\mu}_i\}_{i=N_{\text{train}}} \in \mathcal{P}_{\text{train}}$ is defined using $N_{\text{train}} = 100$ and a uniform probability distribution. The set of points $\{\boldsymbol{x}_i\}_{i=1}^{N_h} \in \Omega$ that is used for the identification of the magic points is chosen to be coincident with the nodes of the finite element grid reported in Figure 1.7. In Figure 1.8 we report the first four EIM basis functions for the nonlinear function g and the location of the magic points identified by the EIM algorithm. In Figure 1.9 we report the convergence analysis of the EIM algorithm for the nonlinear function g changing the number of EIM basis functions (left plot) and the convergence analysis of the ROM changing the number of reduced basis functions (right plot).
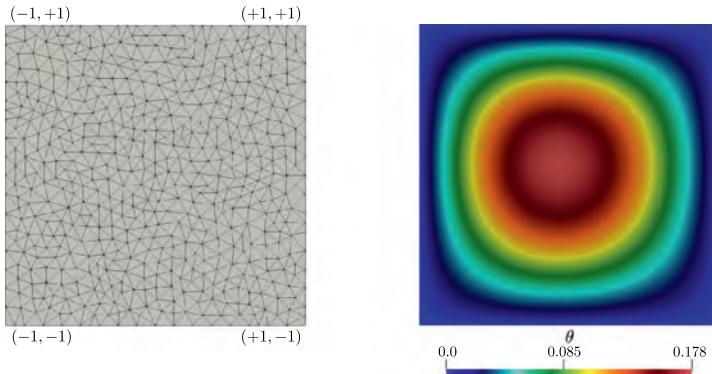
Figure 1.7: Discretized domain into which the parameterized problem is solved (left image), together with an example of the value assumed by the temperature field for one particular sample point inside the parameter space (right image).
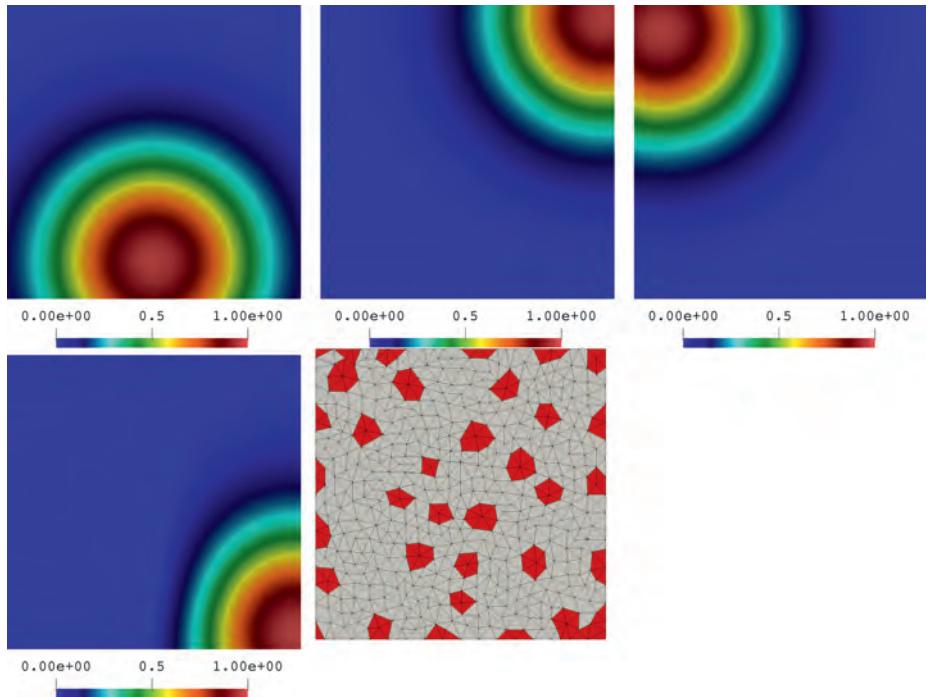
Figure 1.8: Plot of the first four modes identified by the EIM algorithm (first row and left image in the second row) and the location of the first 35 indices i_Q. The magic points are identified by the red elements in the right picture on the second row.
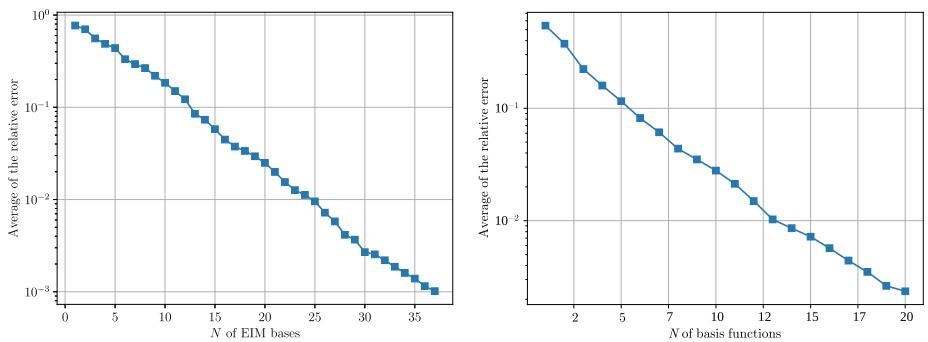
Figure 1.9: Convergence analysis of the numerical example. In the left plot we can see the average value of the L^2 relative error between the exact function g and its EIM approximation. On the right plot we report the average value of the L^2 relative error between the FOM temperature field and the ROM temperature field. The plot is for different numbers of basis functions used to approximate the temperature field and keeping constant the number of basis functions used to approximate the forcing term ($N = 11$).

1.3.5.2 An example in the context of reduced-order models with nonlinearity and nonaffine parametric dependency

In this second illustrative example we show the application of the DEIM algorithm to the stationary parameterized Navier–Stokes equations. In the present case we have both nonlinearity and nonaffinity with respect to the input parameters. Both nonlinearity and nonaffinity have been tackled using the matrix version of the DEIM. The computational domain is given by the unit square $\Omega = [0,1]^2$ and the physical problem is described by the well-known Navier–Stokes equations:

$$\begin{cases} \operatorname{div}(\boldsymbol{u} \otimes \boldsymbol{u}) - \operatorname{div}(2\nu(\boldsymbol{\mu})\nabla^s \boldsymbol{u}) = -\nabla p, & \text{in } \Omega, \\ \operatorname{div} \boldsymbol{u} = \boldsymbol{0}, & \text{in } \Omega, \\ \boldsymbol{u}(x) = (1,0), & \text{on } \Gamma_{\text{TOP}}, \\ \boldsymbol{u}(x) = \boldsymbol{0}, & \text{on } \Gamma_0. \end{cases} \tag{1.87}$$

The physical problem is the classical benchmark of the lid-driven cavity problem with a parameterized diffusivity constant $\nu(\boldsymbol{\mu})$. In this case the impossibility of recovering an affine decomposition of the differential operators is given by the convective term, which is by nature a nonlinear term, and by the parameterized diffusion term. The diffusivity constant $\nu(\boldsymbol{\mu})$ has in fact been parameterized by the following nonlinear function:

$$\nu(\boldsymbol{x}; \boldsymbol{\mu}) = \frac{e^{2(-2(x_1 - \mu_1 - 0.5)^2 - 2(x_2 - \mu_2 - 0.5)^2)}}{100} + 0.01, \tag{1.88}$$

which is a Gaussian function and the position of whose center has been parameterized using the parameter vector $\boldsymbol{\mu} = (\mu_1, \mu_2)$. For the particular case, the discretized algebraic version of the continuous formulation can be rewritten as

$$\begin{pmatrix} \boldsymbol{C}(\boldsymbol{u}) + \boldsymbol{A}(\boldsymbol{\mu}) & \boldsymbol{B}^T \\ \boldsymbol{B} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix}. \tag{1.89}$$

The matrix $\boldsymbol{A}(\boldsymbol{\mu})$ represents the discretized diffusion operator, the matrix $\boldsymbol{C}(\boldsymbol{u})$ represents the discretized nonlinear convective operator, while the term $\boldsymbol{B}$ represents the divergence operator. The term $\boldsymbol{A}(\boldsymbol{\mu})$ is characterized by a nonaffine parametric dependency while the term $\boldsymbol{C}(\boldsymbol{u})$ is characterized by nonlinearity with respect to the solution. The velocity and pressure fields are approximated as

$$\boldsymbol{u}(\boldsymbol{\mu}) \approx \sum_{q=1}^{N_u} c_q^u(\boldsymbol{\mu}) \boldsymbol{h}_q^u, \quad p(\boldsymbol{\mu}) \approx \sum_{q=1}^{N_p} c_q^p(\boldsymbol{\mu}) \boldsymbol{h}_q^p, \tag{1.90}$$

and, in order to achieve an efficient offline-online splitting, the discretized operators are approximated by the matrix version of the DEIM algorithm and expressed as

$$\boldsymbol{A}(\boldsymbol{\mu}) \approx \sum_{q=1}^{N_A} c_q^A(\boldsymbol{\mu}) \boldsymbol{h}_q^A, \quad \boldsymbol{C}(\boldsymbol{u}) \approx \sum_{q=1}^{N_C} c_q^C(\boldsymbol{c}_u) \boldsymbol{h}_q^C. \tag{1.91}$$

The problem is discretized using the finite volume method and a staggered Cartesian grid made of 20×20 cell-centered finite volume elements. The DEIM algorithm has been implemented using 100 samples chosen randomly inside the training space $\mathcal{P}_{\text{EIM}}^{\text{train}} \in [-0.5, 0.5]^2$. The magic points necessary for the implementation of the DEIM algorithm are chosen to be coincident with the cell centers of the discretized problem. The basis functions $\boldsymbol{h}_q^A$ and $\boldsymbol{h}_C^A$ are obtained using the DEIM algorithm applied on the vectorized version of the discretized differential operator snapshots computed during the training stage $\boldsymbol{S}_A = [\text{vec}(\boldsymbol{A}_1), \dots, \text{vec}(\boldsymbol{A}_M)]$ and $\boldsymbol{S}_C = [\text{vec}(\boldsymbol{C}_1), \dots, \text{vec}(\boldsymbol{C}_M)]$. The snapshot matrices $\boldsymbol{S}_A$ and $\boldsymbol{S}_C$ contain in fact the discretized differential operators in vector form obtained for the different samples of the training set.

In Figure 1.10 we report the comparison of the full-order model fields and the ROM ones; the comparison is depicted for a parameter sample not used to train the ROM.

U Magnitude

0.00e+00 0.20.30.40.50.60.70.8 1.00e+00

(a)

Ured Magnitude

0.00e+00 0.20.30.40.50.60.70.8 1.00e+00

(b)

p

-4.20e-01 0 0.2 0.4 7.70e-01

(c)

Pred

-4.20e-01 0 0.2 0.4 7.70e-01

(d)
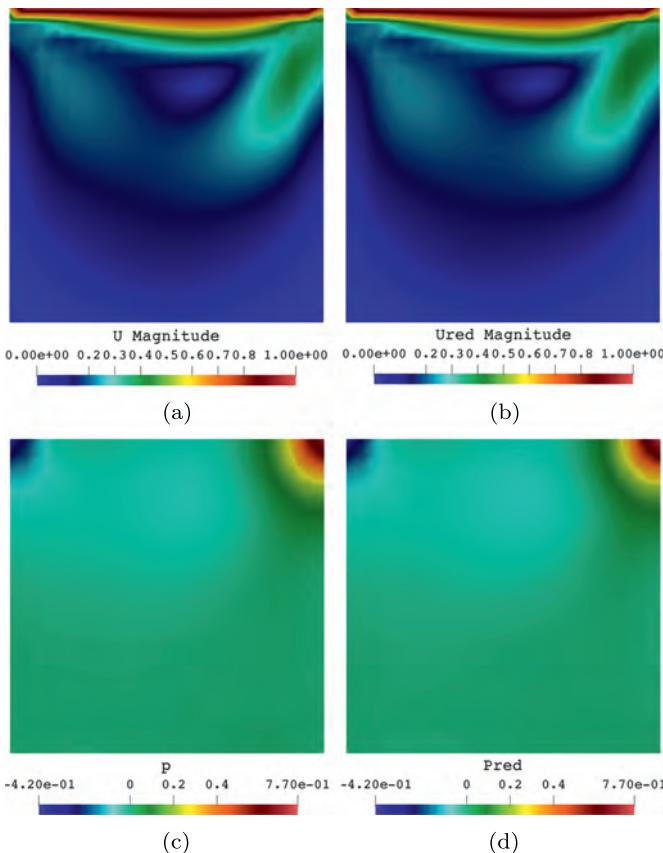
Figure 1.10: Comparison between the FOM velocity (a) and pressure (c) fields and the ROM velocity (b) and pressure (d) fields. The plots are reported for one selected sample value inside the testing set. The ROM solutions have been computed using 14 basis functions for the velocity space, 10 for the pressure space, 10 DEIM basis functions for the convective matrix $\boldsymbol{C}$, and 10 DEIM basis functions for the diffusion matrix $\boldsymbol{B}$.
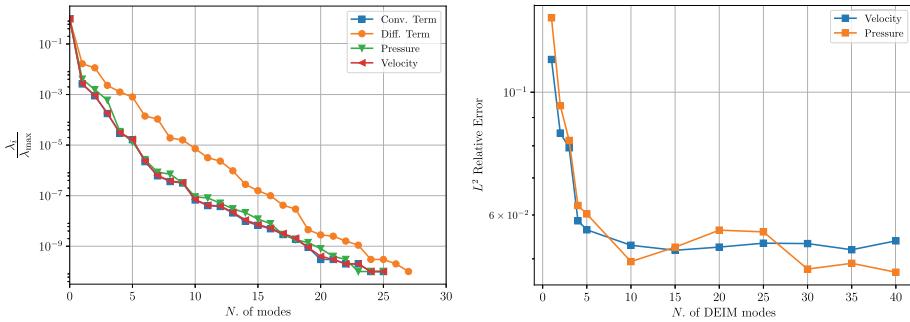
Figure 1.11: Eigenvalue decay of the POD procedure during the DEIM algorithm (left plot). The convergence analysis with respect to the number of DEIM basis functions (right plot), which is computed using the average value over the testing set of the L^2 relative error, has been performed keeping constant the number of basis functions used to approximate the velocity and pressure fields ($N_u = 14, N_p = 10$) and changing the number of DEIM basis functions used to approximate the convective and diffusion terms ($N_C = N_A$).

On the right side of Figure 1.11 we report the convergence analysis for the numerical example. The plots are performed testing the ROM on 100 additional sample values selected randomly inside the parameter space $\mathcal{P}_{\text{EIM}}^{\text{test}} \in [-0.5, 0.5]^2$. In the plots is reported the average value over the testing space of the L^2 relative error.

1.4 Advanced tools: reduction in parameter spaces

Often the use of the aforementioned geometrical morphing techniques in Section 1.2 does not tell us how many control points, i. e., geometrical parameters, are enough to conduct a proper analysis. This leads to self-imposing too few parameters in order to avoid the curse of dimensionality and dealing with intractable problems. To overcome this issue there exist techniques for parameter space dimensionality reduction, both linear and nonlinear. In particular we present here the active subspaces property for linear dimensionality reduction, while in the last section we show an overview of possible nonlinear methods.

These methods are intended as general tools, not restricted to parameterized PDEs. Moreover, the nature of the parameter space can be very diverse, including both geometrical and physical parameters. They are data-driven tools working with couples of input/output data, and they can be used to enhance other MOR techniques.

1.4.1 Active subspaces property and its applications

In this and the following sections we present the active subspaces (AS) property proposed by Trent Russi [89] and developed by Paul Constantine [31]. In brief, active sub-

spaces are defined as the leading eigenspaces of the second moment matrix of the function's gradient and constitute a global sensitivity index.

We present how to exploit AS to reduce the parameter space dimensionality, and use it as a powerful preprocessing tool. Moreover, we show how to combine it with a model reduction methodology and present its application to a cardiovascular problem. In particular, after identifying a lower-dimensional parameter subspace, we sample it to apply further MOR methods. This results in improved computational efficiency.

The main characteristic of AS is the fact that it uses information of both the output function of interest and the input parameter space in order to reduce its dimensionality. The active subspaces have been successfully employed in many engineering fields. We cite, among others, applications in magnetohydrodynamics power generation modeling in [50], in naval engineering for the computation of the total drag resistance with both geometrical and physical parameters in [101, 38], and in constrained shape optimization [66] using the concept of shared active subspaces in [100]. There are also applications to turbomachinery in [7], to uncertainty quantification in the numerical simulation of a scramjet in [34], and to the acceleration of Markov chain Monte Carlo in [35]. Extension of active subspace discovery for time-dependent processes and application to a lithium ion battery model can be found in [32]. A multifidelity approach to reduce the cost of performing dimension reduction through the computation of the active subspace matrix is presented in [62]. In [46] the authors exploit AS for Bayesian optimization, while the coupling with ROMs can be found in [39] for a nonintrusive data-driven approach, and the coupling with POD-Galerkin methods for biomedical engineering will be presented in Section 1.4.4 following [99].

1.4.2 Active subspaces definition

Given a parametric scalar function $f(\boldsymbol{\mu}) : \mathbb{R}^p \rightarrow \mathbb{R}$, where p is the number of parameters representing the output of interest, and given a probability density function $\rho : \mathbb{R}^p \rightarrow \mathbb{R}^+$ that represents uncertainty in the model inputs, active subspaces are low-dimensional subspaces of the input space where f varies the most on average. It is a property of the pair (f, ρ) [31]. In order to uncover AS we exploit the gradients of the function with respect to the input parameters, so it can be viewed as a derivative-based sensitivity analysis that unveils low-dimensional parameterization of f using some linear combinations of the original parameters. Roughly speaking, after a rescaling of the input parameter space to the hypercube $[-1, 1]^p$, we rotate it until the lower-rank approximation of the output of interest is discovered, which means a preferred direction in the input space is identified. Then we can project all the data onto the orthogonal space of this preferred direction and we can construct a surrogate model on this low-dimensional space.

Let us add some hypotheses to f in order to properly construct the matrix we will use to find the active subspaces: Let f be continuous and differentiable with square-integrable partial derivatives in the support of ρ. We define the so-called uncentered covariance matrix $\mathbf{C}$ of the gradients of f as the matrix whose elements are the average products of partial derivatives of the map f, that is,

$$\mathbf{C} = \mathbb{E}\left[\nabla_{\boldsymbol{\mu}} f \, \nabla_{\boldsymbol{\mu}} f^T\right] = \int (\nabla_{\boldsymbol{\mu}} f)(\nabla_{\boldsymbol{\mu}} f)^T \rho \, d\boldsymbol{\mu}, \tag{1.92}$$

where $\mathbb{E}$ is the expected value and $\nabla_{\boldsymbol{\mu}} f = \nabla f(\boldsymbol{\mu}) = [\frac{\partial f}{\partial \mu_1}, \ldots, \frac{\partial f}{\partial \mu_p}]^T$ is the column vector of partial derivatives of f. This matrix is symmetric so it has a real eigenvalue decomposition:

$$\mathbf{C} = \mathbf{W} \boldsymbol{\Lambda} \mathbf{W}^T, \tag{1.93}$$

where $\mathbf{W} \in \mathbb{R}^{p \times p}$ is the orthogonal matrix of eigenvectors and $\boldsymbol{\Lambda}$ is the diagonal matrix of nonnegative eigenvalues arranged in descending order. The eigenpairs of the uncentered covariance matrix define the active subspaces of the pair (f, ρ). Moreover, Lemma 2.1 in [33] states that the eigenpairs are functionals of $f(\boldsymbol{\mu})$ and we have

$$\lambda_i = \mathbf{w}_i^T \mathbf{C} \mathbf{w}_i = \int \left(\nabla_{\boldsymbol{\mu}} f^T \mathbf{w}_i\right)^2 \rho \, d\boldsymbol{\mu}, \tag{1.94}$$

which means that the i-th eigenvalue is the average squared directional derivative of f along the eigenvector $\mathbf{w}_i$. Alternatively we can say that the eigenvalues represent the magnitude of the variance of $\nabla_{\boldsymbol{\mu}} f$ along their eigenvectors orientations. So small values of the eigenvalues correspond to small perturbation of f along the corresponding eigenvectors. It also follows that large gaps between eigenvalues indicate directions where f changes the most on average. Since we consider the lower-dimensional space of dimension $M < p$ where the target function has exactly this property, we define the active subspace of dimension M as the span of the first M eigenvectors (they correspond to the most energetic eigenvalues before a gap). Let us partition $\boldsymbol{\Lambda}$ and $\mathbf{W}$ as

$$\boldsymbol{\Lambda} = \begin{bmatrix} \boldsymbol{\Lambda}_1 & \\ & \boldsymbol{\Lambda}_2 \end{bmatrix}, \quad \mathbf{W} = [\mathbf{W}_1 \quad \mathbf{W}_2], \tag{1.95}$$

where $\boldsymbol{\Lambda}_1 = \mathrm{diag}(\lambda_1, \ldots, \lambda_M)$ and $\mathbf{W}_1$ contains the first M eigenvectors. We can use $\mathbf{W}_1$ to project the original parameters to the active subspace obtaining the reduced parameters, that is, the input space is geometrically transformed and aligned with $\mathbf{W}_1$, in order to retain only the directions where the function variability is high. We call the active variable $\boldsymbol{\mu}_M$ the range of $\mathbf{W}_1^T$ and the inactive variable $\boldsymbol{\eta}$ the range of $\mathbf{W}_2^T$:

$$\boldsymbol{\mu}_M = \mathbf{W}_1^T \boldsymbol{\mu} \in \mathbb{R}^M, \quad \boldsymbol{\eta} = \mathbf{W}_2^T \boldsymbol{\mu} \in \mathbb{R}^{p-M}. \tag{1.96}$$

We can thus express any point in the parameter space $\boldsymbol{\mu} \in \mathbb{R}^p$ in terms of $\boldsymbol{\mu}_M$ and $\boldsymbol{\eta}$ as

$$\boldsymbol{\mu} = \mathbf{W}\mathbf{W}^T \boldsymbol{\mu} = \mathbf{W}_1 \mathbf{W}_1^T \boldsymbol{\mu} + \mathbf{W}_2 \mathbf{W}_2^T \boldsymbol{\mu} = \mathbf{W}_1 \boldsymbol{\mu}_M + \mathbf{W}_2 \boldsymbol{\eta}. \tag{1.97}$$

The lower-dimensional approximation, or surrogate quantity of interest, g : $\mathbb{R}^M \to \mathbb{R}$ of the target function f is a function of only the active variable $\boldsymbol{\mu}_M$ as

$$f(\boldsymbol{\mu}) \approx g(\mathbf{W}_1^T \boldsymbol{\mu}) = g(\boldsymbol{\mu}_M). \qquad (1.98)$$

Such g is called ridge function [77] and, as we can infer from this section, it is constant along the span of $\mathbf{W}_2$.

From a practical point of view, equation (1.92) is estimated through the Monte Carlo method. We draw N_{train} independent samples $\boldsymbol{\mu}^{(i)}$ according to the measure ρ and we approximate

$$\mathbf{C} \approx \hat{\mathbf{C}} = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \nabla_{\boldsymbol{\mu}} f_i \, \nabla_{\boldsymbol{\mu}} f_i^T = \hat{\mathbf{W}} \hat{\boldsymbol{\Lambda}} \hat{\mathbf{W}}^T, \qquad (1.99)$$

where $\nabla_{\boldsymbol{\mu}} f_i = \nabla_{\boldsymbol{\mu}} f(\boldsymbol{\mu}^{(i)})$. In [31] the authors provide a heuristic formula for the number of samples N_{train} needed to properly estimate the first k eigenvalues, that is,

$$N_{\text{train}} = \alpha \, k \ln(p), \qquad (1.100)$$

where α usually is between 2 and 10. Moreover, they prove that for sufficiently large N_{train} the error ε committed in the approximation of the active subspace of dimension n is bounded from above by

$$\varepsilon = \text{dist}(\text{rank}(\mathbf{W}_1), \text{rank}(\hat{\mathbf{W}}_1)) \le \frac{4\lambda_1 \delta}{\lambda_n - \lambda_{n+1}}, \qquad (1.101)$$

where δ is a positive scalar bounded from above by $\frac{\lambda_n - \lambda_{n+1}}{5\lambda_1}$. Here we can clearly see how the gap between two eigenvalues is important in order to properly approximate f exploiting AS.

1.4.3 Some examples

In this section we present two simple examples with the computation of the active subspaces using analytical gradients. To highlight the possibility that the presence of an active subspace is not always guaranteed we also show an example in this direction. We choose for both the cases a three-dimensional input parameter space without loss of generality. In order to identify the low-dimensional structure of the function of interest we use the sufficient summary plots, developed in [36]. In our cases, they are scatter plots of $f(\boldsymbol{\mu})$ against the active variable $\boldsymbol{\mu}_M$.

The presence of an active subspace is not always guaranteed. For example, not every target function that has a radial symmetry has a lower-dimensional representation in terms of active variables. This is due to the fact that there is no rotation of the

input parameter space that aligns it along a preferred direction since all of them are equally important.

Let us consider for example the function $f(\boldsymbol{\mu}) = \frac{1}{2}\boldsymbol{\mu}^T\boldsymbol{\mu}$ representing an n-dimensional elliptic paraboloid, where the parameter $\boldsymbol{\mu}$ is a column vector in $[-1, 1]^3$. In this case we have the exact derivatives, in fact, $\nabla_{\boldsymbol{\mu}} f = \boldsymbol{\mu}$, and we do not have to approximate them. If we draw 1,000 samples and we apply the procedure to find an active subspace and we plot the sufficient summary plot in one dimension, as in Figure 1.12, we clearly see how it is unable to find the active variable along which f varies the most on average. In fact there is not a significant gap between the eigenvalues, since we have $\mathbf{C} = \frac{1}{3}\mathbf{Id}$. Moreover the projection of the data onto the inactive subspace suggests the presence of an n-dimensional elliptic paraboloid.
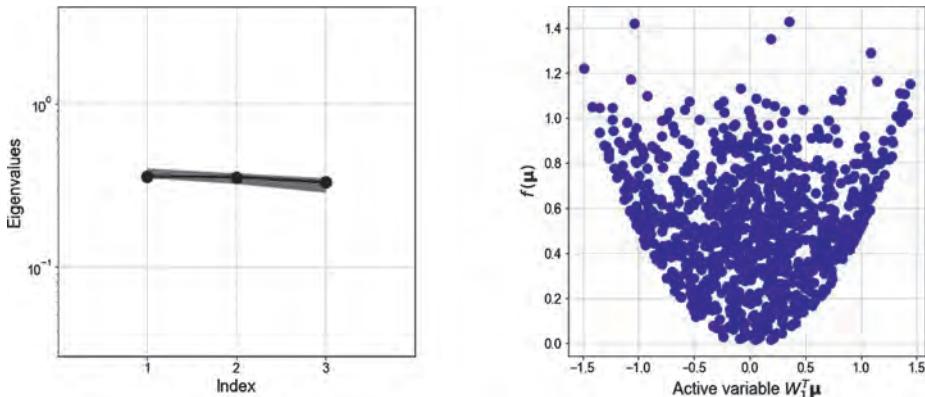
Figure 1.12: Example of an output function with a radial symmetry. On the left the exact eigenvalues of the uncentered covariance matrix are shown. On the right the sufficient summary plot in one dimension ($f(\boldsymbol{\mu})$ against $\boldsymbol{\mu}_M = \mathbf{W}_1^T\boldsymbol{\mu}$) shows how the projection of the data along the inactive directions does not unveil a lower-dimensional structure for f.

Let us consider now another quadratic function in three variables. We define the output of interest f as

$$f(\boldsymbol{\mu}) = \frac{1}{2}\boldsymbol{\mu}^T \mathbf{A}\boldsymbol{\mu}, \tag{1.102}$$

where $\boldsymbol{\mu} \in [-1, 1]^3$ and $\mathbf{A}$ is symmetric positive definite with a major gap between the first and the second eigenvalue. With this form we can compute the exact gradients as $\nabla_{\boldsymbol{\mu}} f(\boldsymbol{\mu}) = \mathbf{A}\boldsymbol{\mu}$ and, taking ρ as a uniform density function, compute $\mathbf{C}$ as

$$\mathbf{C} = \mathbf{A}\left(\int \boldsymbol{\mu}\boldsymbol{\mu}^T \rho \, d\boldsymbol{\mu}\right)\mathbf{A}^T = \frac{1}{3}\mathbf{A}^2. \tag{1.103}$$

So the squared eigenvalues of **A** are the eigenvalues of **C**. Since, by definition, **A** has a significant gap between the first and second eigenvalues, we can easily find an active subspace of dimension one.

In Figure 1.13 we show the sufficient summary plot of f with respect to its active variable. A clear univariate behavior is present, as expected, so we can easily construct g, for instance taking a quadratic one-dimensional function. We can also see the associated eigenvalues of the uncentered covariance matrix.
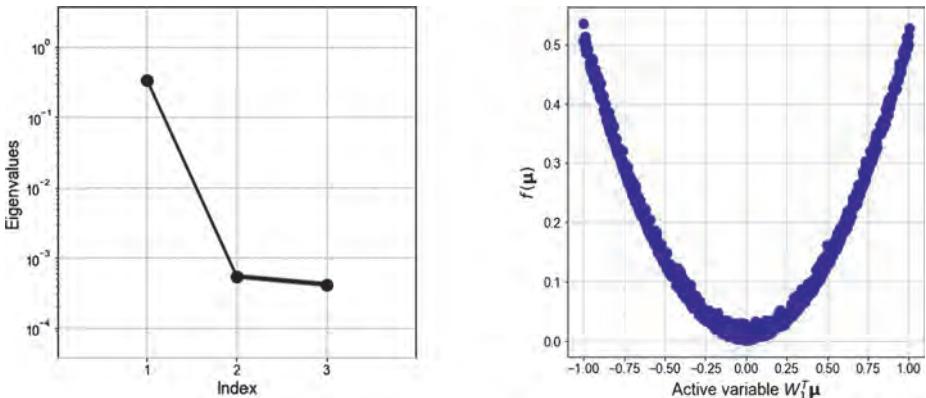
Figure 1.13: Example of a quadratic function with an active subspace of dimension one. On the left the exact eigenvalues of the uncentered covariance matrix are shown. On the right the sufficient summary plot in one dimension ($f(\boldsymbol{\mu})$ against $\boldsymbol{\mu}_M = \mathbf{W}_1^T \boldsymbol{\mu}$) shows how the projection of the data along the inactive directions unveils a univariate structure for f.

1.4.4 Active subspaces as preprocessing tool to enhance model reduction

The presence of an active subspace for an output of interest, derived from the solution of a parametric PDE, can be exploited for further MOR. Thus, in this context, AS can be seen as a powerful preprocessing technique to both reduce the parameter space dimensionality and boost the performance of other model order reduction methods.

In [99] the active subspace for a relative pressure drop in a stenosed carotid artery is used as a reduced sampling space to improve the reconstruction of the output manifold. We used as parameters the displacement of a selection of RBF control points to simulate the occlusion of the carotid artery after the bifurcation. For a review of the RBF interpolation technique, see Section 1.2.2. In Figure 1.5 two different views of the same carotid are shown and the control points are highlighted with green dots. The target function was a relative pressure drop between the two branches computed solving a stationary Navier–Stokes problem.

After the identification of the active subspace we exploit it by sampling the original full parameter space along the active subspace. These sampled parameters were used, in the offline phase, to construct the snapshots matrix for the training of an ROM. This leads to better approximation properties for a given number of snapshots with respect to usual sampling techniques. The natural construction of the uncentered covariance matrix, which uses information from both the inputs and the outputs, is the reason of such improvements.

The same idea has been coupled also with nonintrusive MOR techniques, such as POD with interpolation (PODI), in [102], while for the reconstruction of modal coefficients using PODI with AS for low computational budgets we suggest [39].

1.4.5 About nonlinear dimensionality reduction

There are plenty of other techniques that reduce the dimensionality of a given data set. They do not exploit simultaneously the structure of the output function and the input parameter space like AS, they just express the data vectors we want to reduce in a reduced space embedded in the original one. For a comprehensive overview, see [64] and [104]. The main assumption is that the data set at hand has an intrinsic dimensionality, which is lower than that of the full space where they belong. This means that the data are lying on or near a manifold with dimensionality d embedded in a greater space of dimension D. If we approximate this manifold with a linear subspace we use a linear dimensionality reduction technique; otherwise assuming the data lie on a curved manifold we can achieve better results using a nonlinear method. Unfortunately in general neither the characteristics of the manifold, nor the intrinsic dimensionality are known, so the dimensionality reduction problem is ill-posed. There are several algorithms to detect the intrinsic dimensionality of a data set; we suggest the review in [17]. Among all we cite two of the most popular techniques, i. e., locally linear embedding (LLE), presented in [83], and Isomap [98]. Extensions for the two methods can be found in [12].

LLE seeks to preserve local properties of the high-dimensional data in the embedded space, and it is able to detect nonconvex manifolds. In particular it preserves local reconstruction weights of the neighborhood graph, that is, LLE fits a hyperplane through each data point and its nearest neighbors. Some applications can be found in [51] for biomedical engineering, or in [60] for computational mechanics.

Isomap instead seeks to preserve geodesic (or curvilinear) distances between the high-dimensional data points and the lower-dimensional embedded ones. Its topological stability has been investigated in [8], while it has been used for micromotility reconstruction in [5].

Other approaches include for example a manifold walking algorithm that has been proposed in [73] and in [74].

1.5 Conclusion and outlook

This introductory chapter provided the means to understand projection-based MOR methods in Section 1.1. Various techniques allowing the parameterization of complicated geometries were provided in Section 1.2. Since many geometries of interest introduce nonlinearities or nonaffine parameter dependency, an intermediate step such as the *EIM* is often applied. The basics were presented in Section 1.3 and will be used further in the chapter on hyperreduction (Chapter 5) of this volume. The reduction in parameter space becomes necessary if high-dimensional parameter spaces are considered. Active subspaces (Section 1.4) provide a mean to tackle the curse of dimensionality.

Each chapter of the handbook gives in-depth technical details upon a particular topic of interest. This includes common MOR methods, several application areas of interest, and a survey of current software frameworks for model reduction. Whenever a method does not rely only on the PDE-based functional analysis setting introduced in this chapter, corresponding requirements will be mentioned within each technical chapter.

Bibliography

[1]   R. A. Adams, *Sobolev Spaces*, Academic Press New York, 1975.
[2]   M. Ainsworth and J. T. Oden, A posteriori error estimation in finite element analysis, *Computer Methods in Applied Mechanics and Engineering*, **142** (1) (1997), 1–88.
[3]   B. O. Almroth, P. Stern and F. A. Brogan, Automatic choice of global shape functions in structural analysis, *AIAA Journal*, **16** (5) (1978), 525–528.
[4]   D. Arnold et al., Unified analysis of discontinuous Galerkin methods for elliptic problems, *SIAM Journal on Numerical Analysis*, **39** (5) (2002), 1749–1779.
[5]   M. Arroyo et al., Reverse engineering the euglenoid movement, *Proceedings of the National Academy of Sciences*, **109** (44) (2012), 17874–17879.
[6]   I. Babuska, Error-bounds for finite element method, *Numerische Mathematik*, **16** (1970/1971), 322–333.
[7]   S. Bahamonde et al., Active subspaces for the optimal meanline design of unconventional turbomachinery, *Applied Thermal Engineering*, **127** (2017), 1108–1118.
[8]   M. Balasubramanian and E. L. Schwartz, The isomap algorithm and topological stability, *Science*, **295** (5552) (2002), 7.
[9]   F. Ballarin et al., A POD-selective inverse distance weighting method for fast parametrized shape morphing, *International Journal for Numerical Methods in Engineering*, **117** (8) (2019), 860–884, 10.1002/nme.5982.
[10] M. Barrault et al., An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, *Comptes Rendus. Mathematique*, **339** (9) (2004), 667–672, 10.1016/j.crma.2004.08.006.
[11] A. Beckert and H. Wendland, Multivariate interpolation for fluid-structure-interaction problems using radial basis functions, *Aerospace Science and Technology*, **5** (2) (2001), 125–134.

[12] Y. Bengio et al., Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering, in *Advances in Neural Information Processing Systems*, pp. 177–184, 2004.

[13] D. Boffi, F. Brezzi and M. Fortin, *Mixed Finite Element Methods and Applications*, Springer-Verlag Berlin Heidelberg, 2013.

[14] D. Bonomi, A. Manzoni and A. Quarteroni, A matrix DEIM technique for model reduction of nonlinear parametrized problems in cardiac mechanics, *Computer Methods in Applied Mechanics and Engineering*, **324** (2017), 300–326, 10.1016/j.cma.2017.06.011.

[15] M. D. Buhmann, *Radial Basis Functions: Theory and Implementations*, vol. 12, Cambridge university press, 2003.

[16] T. Bui-Thanh, M. Damodaran and K. Willcox, Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics, in *21st AIAA Applied Aerodynamics Conference*, American Institute of Aeronautics and Astronautics, jun 2003, 10.2514/6.2003-4213.

[17] F. Camastra, Data dimensionality estimation methods: a survey, *Pattern Recognition*, **36** (12) (2003), 2945–2954.

[18] C. Canuto, T. Tonn and K. Urban, A posteriori error analysis of the reduced basis method for nonaffine parametrized nonlinear PDEs, *SIAM Journal on Numerical Analysis*, **47** (3) (2009), 2001–2022, 10.1137/080724812.

[19] C. Canuto et al., *Spectral Methods: Fundamentals in Single Domains*, Springer-Verlag Berlin Heidelberg, 2006.

[20] K. Carlberg, C. Bou-Mosleh and C. Farhat, Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations, *International Journal for Numerical Methods in Engineering*, **86** (2) (2010), 155–181, 10.1002/nme.3050.

[21] F. Casenave, A. Ern and T. Lelièvre, A nonintrusive reduced basis method applied to aeroacoustic simulations, *Advances in Computational Mathematics*, **41** (5) (2014), 961–986, 10.1007/s10444-014-9365-0.

[22] S. Chaturantabut and D. C. Sorensen, Discrete empirical interpolation for nonlinear model reduction, in:*Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, IEEE, dec 2009, 10.1109/cdc.2009.5400045.

[23] S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM Journal on Scientific Computing*, **32** (5) (2010), 2737–2764, 10.1137/090766498.

[24] P. Chen, A. Quarteroni and G. Rozza, A weighted empirical interpolation method: a priori convergence analysis and applications, *ESAIM: Mathematical Modelling and Numerical Analysis*, **48** (4) (2014), 943–953, 10.1051/m2an/2013128.

[25] Y. Chen, A certified natural-norm successive constraint method for parametric inf–sup lower bounds, *Applied Numerical Mathematics*, **99** (2016), 98–108, ISSN: 0168-9274.

[26] Y. Chen et al., A monotonic evaluation of lower bounds for inf-sup stability constants in the frame of reduced basis approximations, *Comptes Rendus. Mathématique*, **346** (23) (2008), 1295–1300.

[27] Y. Chen et al., Improved successive constraint method based a posteriori error estimate for reduced basis approximation of 2D Maxwell's problem, *ESAIM: Mathematical Modelling and Numerical Analysis*, **43** (6) (2009), 1099–1116.

[28] F. Chinesta et al., Model order reduction, in *Encyclopedia of Computational Mechanics*, Second Edition, pp. 1–36, Elsevier, 2017.

[29] P. Ciarlet, *The Finite Element Method for Elliptic Problems*, Classics in Applied Mathematics, vol. 40, Society for Industrial and Applied Mathematics, Philadelphia, 2002.

[30] P. Ciarlet, *Linear and Nonlinear Functional Analysis with Applications*, Society for Industrial and Applied Mathematics, Philadelphia, 2014.

[31] P. G. Constantine, *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*, vol. 2, SIAM, 2015.

[32] P. G. Constantine and A. Doostan, Time-dependent global sensitivity analysis with active subspaces for a lithium ion battery model, *Statistical Analysis and Data Mining: The ASA Data Science Journal*, **10** (5) (2017), 243–262.

[33] P. G. Constantine, E. Dow and Q. Wang, Active subspace methods in theory and practice: applications to kriging surfaces, *SIAM Journal on Scientific Computing*, **36** (4) (2014), A1500–A1524.

[34] P. G. Constantine, et al., Exploiting active subspaces to quantify uncertainty in the numerical simulation of the HyShot II scramjet, *Journal of Computational Physics*, **302** (2015), 1–20.

[35] P. G. Constantine, C. Kent and T. Bui-Thanh, Accelerating Markov chain Monte Carlo with active subspaces, *SIAM Journal on Scientific Computing*, **38** (5) (2016), A2779–A2805.

[36] R. D. Cook, *Regression Graphics: Ideas for Studying Regressions through Graphics*, vol. 482, John Wiley & Sons, 2009.

[37] N. Demo et al., Shape optimization by means of proper orthogonal decomposition and dynamic mode decomposition, in *Technology and Science for the Ships of the Future: Proceedings of NAV 2018: 19th International Conference on Ship & Maritime Research*, pp. 212–219, IOS Press, 2018, 10.3233/978-1-61499-8709-212.

[38] N. Demo et al., An efficient shape parametrisation by free-form deformation enhanced by active subspace for hull hydrodynamic ship design problems in open source environment, in *The 28th International Ocean and Polar Engineering Conference*, 2018.

[39] N. Demo, M. Tezzele and G. Rozza, A non-intrusive approach for proper orthogonal decomposition modal coefficients reconstruction through active subspaces, *Comptes Rendus de l'Académie des Sciences DataBEST* **2019** (Special Issue) (2019).

[40] S. Deparis, D. Forti and A. Quarteroni, A rescaled localized radial basis function interpolation on non-Cartesian and nonconforming grids, *SIAM Journal on Scientific Computing*, **36** (6) (2014), A2745–A2762.

[41] S. Deparis and G. Rozza, Reduced basis method for multi-parameter-dependent steady Navier–Stokes equations: applications to natural convection in a cavity, *Journal of Computational Physics*, **228** (12) (2009), 4359–4378.

[42] J. Duchon, Splines minimizing rotation-invariant semi-norms in Sobolev spaces, *Constructive Theory of Functions of Several Variables* (1977), 85–100.

[43] C. Eckart and G. Young, The approximation of one matrix by another of lower rank. Psychometrika, *Psychometrika*, **1** (1936), 211–218, 10.1007/BF02288367.

[44] J. L. Eftang, M. A. Grepl and A. T. Patera, A posteriori error bounds for the empirical interpolation method, *Comptes Rendus. Mathématique*, **348** (9) (2010), 575–579, 10.1016/j.crma.2010.03.004.

[45] J. L. Eftang and B. Stamm, Parameter multi-domain 'hp' empirical interpolation, *International Journal for Numerical Methods in Engineering*, **90** (4) (2012), 412–428, 10.1002/nme.3327.

[46] D. Eriksson et al., Scaling Gaussian process regression with derivatives, in *Advances in Neural Information Processing Systems*, pp. 6867–6877, 2018.

[47] L. C. Evans, *Partial Differential Equations*, American Mathematical Society, 1998.

[48] R. Eymard, T. R. Gallouët and R. Herbin, *The finite volume method*, in P. G. Ciarlet and J. L. Lions (eds.) Handbook of Numerical Analysis, vol. 7, pp. 713–1020, 2000.

[49] D. Forti and G. Rozza, Efficient geometrical parametrisation techniques of interfaces for reduced-order modelling: application to fluid–structure interaction coupling problems, *International Journal of Computational Fluid Dynamics*, **28** (3-4) (2014), 158–169.

[50] A. Glaws et al., Dimension reduction in magnetohydrodynamics power generation models: Dimensional analysis and active subspaces, *Statistical Analysis and Data Mining: The ASA Data Science Journal*, **10** (5) (2017), 312–325.

[51] D. González, E. Cueto and F. Chinesta, Computational patient avatars for surgery planning, *Annals of Biomedical Engineering*, **44** (1) (2016), 35–45.

[52] T. Graetsch and K.-J. Bathe, A posteriori error estimation techniques in practical finite element analysis, *Computers & Structures*, **83** (4) (2005), 235–265.

[53] M. A. Grepl et al., Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, *ESAIM: Mathematical Modelling and Numerical Analysis*, **41** (3) (2007), 575–605, 10.1051/m2an:2007031.

[54] M. W. Hess, S. Grundel and P. Benner, Estimating the inf-sup constant in reduced basis methods for time-harmonic Maxwell's equations, *IEEE Transactions on Microwave Theory and Techniques*, **63** (2015), 3549–3557.

[55] J. Hesthaven, B. Stamm and S. Zhang, Certified reduced basis method for the electric field integral equation, *SIAM Journal on Scientific Computing*, **34** (3) (2012), A1777–A1799.

[56] 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: Mathematical Modelling and Numerical Analysis*, **48** (1) (2014), 259–283, 10.1051/m2an/2013100.

[57] D. B. P. Huynh et al., A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants, *Comptes Rendus. Mathématique*, **345** (8) (2007), 473–478.

[58] D. B. P. Huynh et al., A natural-norm successive constraint method for inf-sup lower bounds, *Computer Methods in Applied Mechanics and Engineering*, **199** (29-32) (2010), 1963–1975.

[59] L. Iapichino, S. Ulbrich and S. Volkwein, Multiobjective PDE-constrained optimization using the reduced-basis method, *Advances in Computational Mathematics*, **43** (5) (2017), 945–972.

[60] R. Ibanez et al., A manifold learning approach to data-driven computational elasticity and inelasticity, *Archives of Computational Methods in Engineering*, **25** (1) (2018), 47–57.

[61] A. Kolmogoroff, Uber Die Beste Annaherung Von Funktionen Einer Gegebenen Funktionenklasse, *Annals of Mathematics*, **37** (1) (1936), 107, 10.2307/1968691.

[62] R. R. Lam et al., Multifidelity dimension reduction via active subspaces, *SIAM Journal on Scientific Computing*, **42** (2) (2020), A929–A956.

[63] T. Lassila and G. Rozza, Parametric free-form shape design with PDE models and reduced basis method, *Computer Methods in Applied Mechanics and Engineering*, **199** (23-24) (2010), 1583–1592.

[64] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*, Springer Science & Business Media, 2007.

[65] M. Lombardi et al., Numerical simulation of sailing boats: dynamics, FSI, and shape optimization, in *Variational Analysis and Aerospace Engineering: Mathematical Challenges for Aerospace Design*, p. 339, Springer, 2012.

[66] T. W. Lukaczyk et al., Active subspaces for shape optimization, in *10th AIAA Multidisciplinary Design Optimization Conference*, p. 1171, 2014.

[67] Y. Maday, O. Mula and G. Turinici, Convergence analysis of the generalized empirical interpolation method, *SIAM Journal on Numerical Analysis*, **54** (3) (2016), 1713–1731, 10.1137/140978843.

[68] Y. Maday and O. Mula, A generalized empirical interpolation method: application of reduced basis techniques to data assimilation, in *Analysis and Numerics of Partial Differential Equations*, pp. 221–235, Springer Milan, 2013, 10.1007/978-88-470-2592-9_13.

[69]   Y. Maday et al., A general multipurpose interpolation procedure: the magic points, *Communications on Pure and Applied Analysis*, **8** (1) (2008), 383–404, 10.3934/cpaa.2009.8.383.

[70]   A. Manzoni, An efficient computational framework for reduced basis approximation and a posteriori error estimation of parametrized Navier–Stokes flows, *ESAIM: Mathematical Modelling and Numerical Analysis*, **48** (4) (2014), 1199–1226.

[71]   A. Manzoni and F. Negri, Heuristic strategies for the approximation of stability factors in quadratically nonlinear parametrized PDEs, *Advances in Computational Mathematics*, **41** (5) (2015), 1255–1288.

[72]   A. Manzoni, A. Quarteroni and G. Rozza, Model reduction techniques for fast blood flow simulation in parametrized geometries, *International Journal for Numerical Methods in Biomedical Engineering*, **28** (6-7) (2012), 604–625.

[73]   L. Meng et al., Identification of material properties using indentation test and shape manifold learning approach, *Computer Methods in Applied Mechanics and Engineering*, **297** (2015), 239–257.

[74]   L. Meng et al., Nonlinear shape-manifold learning approach: concepts, tools and applications, *Archives of Computational Methods in Engineering*, **25** (1) (2018), 1–21.

[75]   A. M. Morris, C. B. Allen and T. C. S. Rendall, CFD-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation, *International Journal for Numerical Methods in Fluids*, **58** (8) (2008), 827–860.

[76]   A. K. Noor, On making large nonlinear problems small, *Computer Methods in Applied Mechanics and Engineering*, **34** (1982), 955–985.

[77]   A. Pinkus, *Ridge Functions*, vol. 205, Cambridge University Press, 2015.

[78]   C. Prud'Homme et al., Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods, *Journal of Fluids Engineering*, **124** (1) (2002), 70–80.

[79]   PyGeM, *Python Geometrical Morphing*, 2017. https://github.com/mathLab/PyGeM (visited on 01/2017).

[80]   A. Quarteroni, *Numerical Models for Differential Problems*, Modeling, Simulation and Applications vol. 16, Springer International Publishing, 2017.

[81]   T. Rebollo et al., On a certified Smagorinsky reduced basis turbulence model, *SIAM Journal on Numerical Analysis*, **55** (6) (2017), 3047–3067.

[82]   M. Renardy and R. C. Rogers, *An Introduction to Partial Differential Equations*, Springer-Verlag New York, 2004.

[83]   S. T. Roweis and L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science*, **290** (5500) (2000), 2323–2326.

[84]   G. Rozza et al., Real-time reliable simulation of heat transfer phenomena, in *ASME -American Society of Mechanical Engineers – Heat Transfer Summer Conference, paper HT2009-88212, volume 3*, pp. 851–860, 2009.

[85]   G. Rozza, D. B. P. Huynh and A. T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, *Archives of Computational Methods in Engineering*, **15** (2008), 229–275.

[86]   G. Rozza, A. Koshakji and A. Quarteroni, Free form deformation techniques applied to 3D shape optimization problems, *Communications in Applied and Industrial Mathematics*, **4** (2013), 1–26, 10.1685/journal.caim.452.

[87]   G. Rozza et al., Advances in reduced order methods for parametric industrial problems in computational fluid dynamics, in *ECCOMAS ECCM - ECFD Conference Proceedings*, Glasgow, UK, 2018.

[88]   W. Rudin, *Principles of Mathematical Analysis*, International Series in Pure and Applied Mathematics, McGraw-Hill, 1976.

[89]   T. M. Russi, *Uncertainty Quantification with Experimental Data and Complex System Models*, Ph.D. thesis, UC Berkeley, 2010.

[90]   F. Salmoiraghi et al., Advances in geometrical parametrization and reduced order models and methods for computational fluid dynamics problems in applied sciences and engineering: Overview and perspectives, *ECCOMAS Congress 2016 - Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering*, **1** (2016), 1013–1031, 10.7712/100016.1867.8680.

[91]   F. Salmoiraghi et al., Free-form deformation, mesh morphing and reduced-order methods: enablers for efficient aerodynamic shape optimisation, *International Journal of Computational Fluid Dynamics*, **32** (4-5) (2018), 233–247, 10.1080/10618562.2018.1514115.

[92]   D. T. Sandwell, Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data, *Geophysical Research Letters*, **14** (2) (1987), 139–142.

[93]   W. H. Schilders, H. A. van der Vorst and J. Rommes, *Model Order Reduction: Theory, Research Aspects and Applications*, Springer-Verlag Berlin Heidelberg, 2008.

[94]   T. W. Sederberg and S. R. Parry, Free-form deformation of solid geometric models, in *Proceedings of SIGGRAPH - Special Interest Group on GRAPHics and Interactive Techniques*, pp. 151–159, SIGGRAPH, 1986.

[95]   D. Shepard, A two-dimensional interpolation function for irregularly-spaced data, in *Proceedings-1968 ACM National Conference*, pp. 517–524, ACM, 1968.

[96]   D. Sieger, S. Menzel and M. Botsch, On shape deformation techniques for simulation-based design optimization, in *New Challenges in Grid Generation and Adaptivity for Scientific Computing*, pp. 281–303, Springer, 2015.

[97]   G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Clarendon Press, Oxford Applied Mathematics and Computing Science Series, 1985.

[98]   J. B. Tenenbaum, V. De Silva and J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science*, **290** (5500) (2000), 2319–2323.

[99]   M. Tezzele, F. Ballarin and G. Rozza, Combined parameter and model reduction of cardiovascular problems by means of active subspaces and POD-Galerkin methods, in D. Boffi et al., (eds.) *Mathematical and Numerical Modeling of the Cardiovascular System and Applications*, pp. 185–207, Springer International Publishing, 2018, 10.1007/978-3-319-96649-6_8.

[100]   M. Tezzele et al., Model order reduction by means of active subspaces and dynamic mode decomposition for parametric hull shape design hydrodynamics, in *Technology and Science for the Ships of the Future: Proceedings of NAV 2018: 19th International Conference on Ship & Maritime Research*, pp. 569–576, IOS Press, 2018, 10.3233/978-1-61499-870-9-569.

[101]   M. Tezzele et al., Dimension reduction in heterogeneous parametric spaces with application to naval engineering shape design problems, *Advanced Modeling and Simulation in Engineering Sciences*, **5** (1) (2018), 25, ISSN: 2213-7467, 10.1186/s40323-018-0118-3.

[102]   M. Tezzele, N. Demo and G. Rozza, Shape optimization through proper orthogonal decomposition with interpolation and dynamic mode decomposition enhanced by active subspaces, in *Proceedings of MARINE 2019: VIII International Conference on Computational Methods in Marine Engineering*, pp. 122–133, 2019.

[103]   S. Vallaghé et al., A successive constraint method with minimal offline constraints for lower bounds of parametric coercivity constant, https://hal.archives-ouvertes.fr/hal-00609212, 2011.

[104]   L. Van Der Maaten, E. Postma and J. Van den Herik, Dimensionality reduction: a comparative review, *Journal of Machine Learning Research*, **10** (2009), 66–71.

[105] R. Verfuerth, *A Posteriori Error Estimation Techniques for Finite Element Methods*, Oxford Univ. Press, 2013.

[106] 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, *International Journal for Numerical Methods in Fluids*, **47** (8-9) (2005), 773–788.

[107] K. Veroy, D. V. Rovas and A. T. Patera, A posteriori error estimation for reduced-basis approximation of parametrized elliptic coercive partial differential equations: "convex inverse" bound conditioners, *ESAIM. Control, Optimisation and Calculus of Variations*, **8** (2002), 1007–1028.

[108] J. A. S. Witteveen and H. Bijl, Explicit mesh deformation using inverse distance weighting interpolation, in *19th AIAA Computational Fluid Dynamics*, AIAA, 2009.

[109] M. Yano, A space-time Petrov–Galerkin certified reduced basis method: application to the Boussinesq equations, *SIAM Journal on Scientific Computing*, **36** (1) (2014), A232–A266.

[110] K. Yosida, *Functional Analysis*, Springer-Verlag Berlin Heidelberg, 1995.

[111] S. Zhang, *Efficient greedy algorithms for successive constraints methods with high-dimensional parameters*, Tech. Report 23, http://www.dam.brown.edu/people/shzhang/greedy_scm.pdf, 2011.

Carmen Gräßle, Michael Hinze, and Stefan Volkwein

2 Model order reduction by proper orthogonal decomposition

Abstract: We provide an introduction to proper orthogonal decomposition (POD) model order reduction with focus on (nonlinear) parametric partial differential equations (PDEs) and (nonlinear) time-dependent PDEs, and PDE-constrained optimization with POD surrogate models as application. We cover the relation of POD and singular value decomposition, POD from the infinite-dimensional perspective, reduction of nonlinearities, certification with a priori and a posteriori error estimates, spatial and temporal adaptivity, input dependency of the POD surrogate model, POD basis update strategies in optimal control with surrogate models, and sketch related algorithmic frameworks. The perspective of the method is demonstrated with several numerical examples.

Keywords: POD model order reduction, (discrete) empirical interpolation, adaptivity, parametric PDEs, evolutionary PDEs, certification with error analysis

MSC 2010: 35B30, 37M99, 41A05, 65K99, 93A15, 93C05

2.1 Introduction

Proper orthogonal decomposition (POD) is a method which comprises the essential information contained in data sets. Data sets may have their origin in various sources, like, e. g., (uncertain) measurements of geophysical processes, numerical simulations of (parameter-dependent) complex physical problems, or (dynamical) imaging. In order to illustrate the POD idea of information extraction, let $\{y_1, \ldots, y_n\} \subset \mathbb{R}^m$ denote a vector cloud (which here serves as our data set), where we suppose at least one of the vectors y_j is nonzero. Let us collect the vectors y_j in the data matrix

$$Y = [y_1 \,|\, \ldots \,|\, y_n] \in \mathbb{R}^{m \times n}.$$

Then we have r = rank $Y \in \{1, \ldots, \min(m, n)\}$. Our aim now is to find a vector $\bar{\psi} \in \mathbb{R}^m$ with length one which carries as much information of this vector cloud as possible. Of course, we here have to specify what *information* in this context means. For this

Carmen Gräßle, Germany
Michael Hinze, University of Koblenz-Landau, Universitätsstr. 1, 56070 Koblenz, Germany
Stefan Volkwein, University of Konstanz, Mathematics and Statistics, Universitätsstrasse 10, 78457 Konstanz, Germany

purpose we equip $\mathbb{R}^m$ with some inner product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \|$. We define the information content of vector y with respect to some unit vector ψ by the quantity $|\langle y, \psi \rangle|$. Then we determine the special vector $\bar{\psi} \in \mathbb{R}^m$ by solving the maximization problem

$$\bar{\psi} \in \arg\max \left\{ \sum_{j=1}^{n} |\langle y_j, \psi \rangle|^2 \mid \psi \in \mathbb{R}^m \text{ with } \|\psi\| = 1 \right\}. \tag{2.1}$$

Note that the solution to the maximization problem in (2.1) is not unique. If $\bar{\psi}$ is a vector, where the maximum is attained, then $-\bar{\psi}$ is an optimal solution, too. Let us label the vector $\bar{\psi}$ by ψ_1. We now iterate this procedure; suppose that for $2 \le \ell \le r$ we have already computed such $\ell - 1$ orthonormal vectors $\{\psi_i\}_{i=1}^{\ell-1}$ and then seek a unit vector $\psi_\ell \in \mathbb{R}^m$ which is perpendicular to the $(\ell - 1)$-dimensional subspace

$$\mathcal{V}^{\ell-1} = \text{span}\{\psi_1, \ldots, \psi_{\ell-1}\} \subset \mathbb{R}^m,$$

and which carries as much information of our vector cloud as possible, i. e., satisfies

$$\psi_\ell = \arg\max \left\{ \sum_{j=1}^{n} |\langle y_j, \psi \rangle|^2 \mid \psi \in \mathbb{R}^m \text{ with } \|\psi\| = 1 \text{ and } \psi \perp \mathcal{V}^{\ell-1} \right\}.$$

It is now straightforward to see that the vectors $\{\psi_i\}_{i=1}^{r}$ are given by

$$W^{1/2}\psi_i = \tilde{\psi}_i, \quad 1 \le i \le r, \tag{2.2}$$

where the $\tilde{\psi}_i$'s solve the eigenvalue problem (cf. [42, 53])

$$\bar{Y}\bar{Y}^\top \tilde{\psi}_i = \lambda_i \tilde{\psi}_i, \quad i = 1, \ldots, r \text{ and } \lambda_1 \ge \cdots \ge \lambda_r > 0,$$

where $\bar{Y} = W^{1/2}Y \in \mathbb{R}^{m \times n}$ with the symmetric, positive definite (weighting) matrix

$$W = ((\langle e_i, e_j \rangle))_{1 \le i,j \le m}. \tag{2.3}$$

In (2.3) the vector e_i denotes the i-th unit vector in $\mathbb{R}^m$. The modes $\{\psi_i\}_{i=1}^{\ell}$ obtained in this way are called *POD modes* or *principal components* of our data cloud. If now $m \gg n \ge r$, it is advantageous to consider the eigenvalue problem

$$\bar{Y}^\top \bar{Y} \phi_i = \lambda_i \phi_i, \quad i = 1, \ldots, r \text{ and } \lambda_1 \ge \cdots \ge \lambda_r > 0,$$

which admits the same eigenvalues λ_i as before. The modes ψ_i and ϕ_i, $i = 1, \ldots, r$, are related by *singular value decomposition* (SVD):

$$\psi_i = \frac{1}{\sigma_i} \bar{Y} \phi_i, \quad i = 1, \ldots, r,$$

and $\sigma_i = \sqrt{\lambda_i} > 0$ is the i-th singular value of the weighted data matrix $\bar{Y}$. Note that in contrast to (2.2) the square root matrix $W^{1/2}$ is not required.

It is now clear that a vector cloud also could be replaced by a function cloud $\{y(\boldsymbol{\mu}_j) \,|\, j = 1, \ldots, n\} \subset X$ in some Hilbert space $(X, \langle \cdot, \cdot \rangle_X)$, where $\{\boldsymbol{\mu}_j\}_{j=1}^n$ are parameters which may refer to, e. g., time instances of a dynamic process, or stochastic variables, and the concept of information extraction by the above maximization problems directly carries over to this situation. As shown in the next section, we can even extend this concept to general Hilbert spaces. This will be formalized in Section 2.2.1. From the considerations above it also becomes clear that POD is closely related to SVD. This is outlined in Section 2.2.2. The POD method for abstract nonlinear evolution problems is explained in Section 2.2.3. The Hilbert space perspective also allows us to treat spatially discrete evolution equations, which include adaptive concepts for the spatial discretization. This is outlined in Section 2.2.4. The POD-Galerkin procedure is explained in Section 2.3, including a discussion of the treatment of nonlinearities. The certification of the POD method with a priori and a posteriori error bounds is outlined in Section 2.4. The POD approach heavily relies on the choice of the snapshots. Related approaches are discussed in Section 2.5. In Section 2.6 we briefly address the scope of the POD method in the context of optimal control of partial differential equations (PDEs). Finally, in Section 2.7 we sketch further important research trends related to POD. Our analytical exposition is supported by several numerical experiments which give an impression of the power of the approach.

POD is one of the most successfully used model reduction techniques for nonlinear dynamical systems; see, e. g., [23, 42, 53, 75, 90] and the references therein. It is applied in a variety of fields, including fluid dynamics, coherent structures [4, 9], and inverse problems [13]. Moreover, in [11] POD is successfully applied to compute reduced-order controllers. The relationship between POD and balancing was considered in [61, 82, 100]. An error analysis for nonlinear dynamical systems in finite dimensions was carried out in [78] and a missing point estimation in models described by POD was studied in [10].

2.2 POD

In this section we introduce a discrete variant of the POD method, where we follow partially [42, Section 1.2.1]. For a continuous variant of the POD method and its relationship to the discrete one we refer the reader to [58] and [42, Sections 1.2.2 and 1.2.3].

2.2.1 The POD method

Suppose that $K, n_1, \ldots, n_K$ are fixed natural numbers. Let the so-called *snapshot ensembles* $\{y_j^k\}_{j=1}^{n_k} \subset X$ be given for $1 \le k \le K$, where X is a separable real Hilbert space.

For POD in complex Hilbert spaces we refer the reader to [96]. We set $n = n_1 + \cdots + n_K$. To avoid a trivial case we suppose that at least one of the y_j^k's is nonzero. Then we introduce the finite-dimensional, linear *snapshot space*

$$\mathcal{V} = \mathrm{span}\,\{y_j^k \mid 1 \le j \le n_k \text{ and } 1 \le k \le K\} \subset X \tag{2.4}$$

with finite dimension $d \le n$. We distinguish two cases:

1) The separable Hilbert space X has finite dimension m: Then X is isomorphic to $\mathbb{R}^m$; see, e. g., [81, p. 47]. We define the finite index set $\mathbb{I} = \{1, \ldots, m\}$. Clearly, we have $1 \le \mathrm{r} \le \min(n, m)$. Especially in the case of $X = \mathbb{R}^m$, the snapshots $y_j^k = (y_{ij}^k)_{1 \le i \le m}$ are vectors in $\mathbb{R}^m$ for $k = 1, \ldots, K$.

2) X is infinite-dimensional: Since X is separable, each orthonormal basis of X has countably many elements. In this case X is isomorphic to the set ℓ_2 of sequences $\{x_i\}_{i \in \mathbb{N}}$ of complex numbers which satisfy $\sum_{i=1}^{\infty} |x_i|^2 < \infty$; see [81, p. 47], for instance. The index set $\mathbb{I}$ is now the countable but infinite set $\mathbb{N}$.

The *POD method* consists in choosing a complete orthonormal basis $\{\psi_i\}_{i \in \mathbb{I}}$ in X such that for every $\ell \in \{1, \ldots, \mathrm{r}\}$ the information content of the given snapshots y_j^k is maximized in the following sense:

$$\left.\begin{array}{l} \max \displaystyle\sum_{i=1}^{\ell} \sum_{k=1}^{K} \sum_{j=1}^{n_k} \alpha_j^k \,|\langle y_j^k, \psi_i\rangle_X|^2 \\[2mm] \text{s. t. } \{\psi_i\}_{i=1}^{\ell} \subset X \text{ and } \langle \psi_i, \psi_j\rangle_X = \delta_{ij},\ 1 \le i, j \le \ell \end{array}\right\} \tag{$\mathbf{P}^{\ell}$}$$

with positive weighting parameters α_j^k, $j = 1, \ldots, n_k$ and $k = 1, \ldots, K$. Here, the symbol δ_{ij} denotes the Kronecker symbol satisfying $\delta_{ii} = 1$ and $\delta_{ij} = 0$ for $i \ne j$.

An optimal solution $\{\Psi_i\}_{i=1}^{\ell}$ to $(\mathbf{P}^{\ell})$ is called a *POD basis of rank ℓ*. It is proved in [42, Theorem 1.8] that for every $\ell \in \{1, \ldots, \mathrm{r}\}$ a solution $\{\Psi_i\}_{i=1}^{\ell}$ to $(\mathbf{P}^{\ell})$ is characterized by the eigenvalue problem

$$\mathcal{R}\Psi_i = \lambda_i \Psi_i \quad \text{for } 1 \le i \le \ell, \tag{2.5}$$

where $\lambda_1 \ge \cdots \ge \lambda_r > 0$ denote the largest eigenvalues of the linear, bounded, nonnegative, and self-adjoint operator $\mathcal{R} : X \to X$ given as

$$\mathcal{R}\Psi = \sum_{k=1}^{K} \sum_{j=1}^{n_k} \alpha_j^k \, \langle \Psi, y_j^k\rangle_X \, y_j^k \quad \text{for } \Psi \in X. \tag{2.6}$$

Moreover, the operator $\mathcal{R}$ can be presented in the form

$$\mathcal{R} = \mathcal{Y}\mathcal{Y}^* \tag{2.7}$$

with the mapping

$$\mathcal{Y} : \mathbb{R}^n \to X, \quad \mathcal{Y}(\Phi) = \sum_{k=1}^{K} \sum_{j=1}^{n_k} \sqrt{\alpha_j^k} \phi_j^k y_j^k \quad \text{for } \Phi = (\phi_1^1, \ldots, \phi_{n_K}^K) \in \mathbb{R}^n,$$

where $\mathcal{Y}^* : X \to \mathbb{R}^n$ denotes the Hilbert space adjoint of $\mathcal{Y}$, whose action is given by

$$\mathcal{Y}^*(\Psi) = (\langle \Psi, \sqrt{\alpha_1^1} y_1^1 \rangle_X, \ldots, \langle \Psi, \sqrt{\alpha_{n_K}^K} y_{n_K}^K \rangle_X)^\top \quad \text{for } \Psi \in X.$$

The operator $\mathcal{K} : \mathbb{R}^n \to \mathbb{R}^n$, $\mathcal{K} := \mathcal{Y}^* \mathcal{Y}$ then admits the same nonzero eigenvalues $\lambda_1 \geq \cdots \geq \lambda_r > 0$ with corresponding eigenvectors $\Phi_1, \ldots, \Phi_r$, and its action is given by

$$\mathcal{K}\Phi = \sum_{k=1}^{K} \sum_{j=1}^{n_k} (\sqrt{\alpha_1^1 \alpha_j^k} \phi_j^k \langle y_j^k, y_1^1 \rangle_X, \ldots, \sqrt{\alpha_{n_K}^K \alpha_j^k} \phi_j^k \langle y_j^k, y_{n_K}^K \rangle_X)^\top \tag{2.8}$$

with the vector $\Phi = (\phi_1^1, \ldots, \phi_{n_K}^K) \in \mathbb{R}^n$. For the eigensystems of $\mathcal{R}$ and $\mathcal{K}$ we have the relation

$$\Phi_i = \frac{1}{\sqrt{\lambda_i}} \mathcal{Y}^* \Psi_i \quad \text{and} \quad \Psi_i = \frac{1}{\sqrt{\lambda_i}} \mathcal{Y} \Phi_i, \quad \text{for } i = 1, \ldots, r. \tag{2.9}$$

Furthermore, we obtain

$$\sum_{i=1}^{\ell} \sum_{k=1}^{K} \sum_{j=1}^{n_k} \alpha_j^k |\langle y_j^k, \Psi_i \rangle_X|^2 = \sum_{i=1}^{\ell} \lambda_i,$$

and for the POD projection error we get

$$\sum_{k=1}^{K} \sum_{j=1}^{n_k} \alpha_j^k \left\| y_j^k - \sum_{i=1}^{\ell} \sum_{k=1}^{K} \sum_{j=1}^{n_k} \langle y_j^k, \Psi_i \rangle_X \Psi_i \right\|_X^2 = \sum_{i=\ell+1}^{r} \lambda_i. \tag{2.10}$$

Thus, the decay rate of the positive eigenvalues $\{\lambda_i\}_{i=1}^r$ plays an essential role for a successful application of the POD method. In general, one has to utilize a complete orthonormal basis $\{\Psi_i\}_{i \in \mathbb{I}} \subset X$ to represent elements in the snapshot space $\mathcal{V}$ by their Fourier sum. This leads to a high-dimensional or even infinite-dimensional approximation scheme. Nevertheless, if the term $\sum_{i=\ell+1}^r \lambda_i$ is sufficiently small for a not too large ℓ, elements in the subspace $\mathcal{V}$ can be approximated by a linear combination of the few basis elements $\{\Psi_i\}_{i=1}^{\ell}$. This offers the chance to reduce the number of terms in the Fourier series using the POD basis of rank ℓ, as shown in the following examples. For this reason it is useful to define information content of the basis $\{\Psi_i\}_{i=1}^{\ell}$ in $\mathcal{V}$ by the quantity

$$\mathcal{E}(\ell) = \frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{i=1}^{r} \lambda_i} \in [0, 1]. \tag{2.11}$$

It can, e. g., be utilized to determine a basis of length $\ell \in \{1, \ldots, r\}$ containing $\approx 99\%$ of the information contained in $\mathcal{V}$ by requiring $\mathcal{E}(\ell) \approx 99\%$. Now it is shown in [42, Section 1.2.1] that

$$\sum_{i=1}^{r} \lambda_i = \sum_{k=1}^{K} \sum_{j=1}^{n_k} \alpha_j^k \|y_j^k\|_X^2$$

holds true. This implies

$$\mathcal{E}(\ell) = \frac{\sum_{i=1}^{\ell} \lambda_i}{\sum_{k=1}^{K} \sum_{j=1}^{n_k} \alpha_j^k \, \|y_j^k\|_X^2} \in [0,1],$$

so that the quantity $\mathcal{E}(\ell)$ can be computed without knowing the eigenvalues $\lambda_{\ell+1}, \ldots, \lambda_r$.

2.2.2 SVD and POD

To investigate the relationship between SVD and POD, let us discuss the POD method for the specific case $X = \mathbb{R}^m$. Then we define the matrices

$$D^k = \begin{pmatrix} \alpha_1^k & & 0 \\ & \ddots & \\ 0 & & \alpha_{n_k}^k \end{pmatrix} \in \mathbb{R}^{n_k \times n_k} \quad \text{for } 1 \le k \le K,$$

$$D = \begin{pmatrix} D^1 & & 0 \\ & \ddots & \\ 0 & & D^K \end{pmatrix} \in \mathbb{R}^{n \times n},$$

$$Y^k = [y_1^k \,|\ldots|\, y_{n_k}^k] \in \mathbb{R}^{m \times n_k} \quad \text{for } 1 \le k \le K,$$

$$Y = [Y^1 \,|\ldots|\, Y^K] \in \mathbb{R}^{m \times n}, \quad \bar{Y} = W^{1/2} Y D^{1/2} \in \mathbb{R}^{m \times n},$$

where we have introduced the weighting matrix $W \in \mathbb{R}^{m \times m}$ in (2.3).

Remark 2.1. Let us mention that $\bar{Y} = Y$ holds true provided all α_j^k are equal to one (i. e., D is the identity matrix) and the inner product in X is given by the Euclidean inner product (i. e., W is the identity matrix).

Now (2.5) is equivalent to the $m \times m$ eigenvalue problem

$$\bar{Y}\bar{Y}^\top \bar{\Psi}_i = \lambda_i \bar{\Psi}_i \quad \text{for } 1 \le i \le \ell \tag{2.12}$$

with $\Psi_i = W^{-1/2}\bar{\Psi}_i$ and the $n \times n$ eigenvalue problem

$$\bar{Y}^\top \bar{Y} \bar{\Phi}_i = \lambda_i \bar{\Phi}_i \quad \text{for } 1 \le i \le \ell \tag{2.13}$$

with $\Psi_i = YD^{1/2}\bar{\Phi}_i/\sqrt{\lambda_i}$. If $m \ll n$ holds, we solve (2.12). However, we have to solve the linear system $W^{1/2}\Psi_i = \bar{\Psi}_i$ for any $i = 1, \ldots, \ell$ in order to get the POD basis $\{\Psi_i\}_{i=1}^{\ell}$. Thus, if $n \le m$ holds, we will compute the solution $\{\bar{\Phi}_i\}_{i=1}^{\ell}$ to (2.13) and get the POD basis by the formula $\Psi_i = YD^{1/2}\bar{\Phi}_i/\sqrt{\lambda_i}$. In that case we also have $\bar{Y}^\top \bar{Y} = Y^\top WY$ so that we do not have to compute the square root matrix $W^{1/2}$. On the other hand, the diagonal matrix $D^{1/2}$ can be computed easily. The relationship between (2.12) and (2.13) is given

by SVD: There exist real numbers $\sigma_1 \geq \cdots \geq \sigma_r > 0$ and orthogonal matrices $\Psi \in \mathbb{R}^{m \times m}$, $\Phi \in \mathbb{R}^{n \times n}$ with column vectors $\{\bar{\Psi}_i\}_{i=1}^{m}$, $\{\bar{\Phi}_i\}_{i=1}^{n}$, respectively, such that

$$\Psi^\top \bar{Y} \Phi = \begin{pmatrix} \Sigma^r & 0 \\ 0 & 0 \end{pmatrix} =: \Sigma \in \mathbb{R}^{m \times n}, \tag{2.14}$$

where $\Sigma^r = \operatorname{diag}(\sigma_1, \ldots, \sigma_r) \in \mathbb{R}^{r \times r}$ and the zeros in (2.14) denote matrices of appropriate dimensions. Moreover, the vectors $\{\bar{\Psi}_i\}_{i=1}^{r}$ and $\{\bar{\Phi}_i\}_{i=1}^{r}$ are eigenvectors of $\bar{Y}\bar{Y}^\top$ and $\bar{Y}^\top \bar{Y}$, respectively, with eigenvalues $\lambda_i = (\sigma_i)^2 > 0$ for $i = 1, \ldots, r$. The vectors $\{\bar{\Psi}_i\}_{i=r+1}^{m}$ and $\{\bar{\Phi}_i\}_{i=r+1}^{n}$ (if $r < m$ respectively $r < n$) are eigenvectors of $\bar{Y}\bar{Y}^\top$ and $\bar{Y}^\top \bar{Y}$ with eigenvalue 0. We summarize the computation of the POD basis in the pseudo-code **function** $[\Psi, \Lambda] = \text{POD}(Y, W, D, \ell, \texttt{flag})$.

---

function $[\Psi, \Lambda] = \text{POD}(Y, W, D, \ell, \texttt{flag})$

Require: Snapshots matrix $Y = [Y^1, \ldots, Y^K]$ with rank r, weighting matrices W, D, number ℓ of POD functions, and `flag` for the solver;

1: **if** `flag` $= 0$ **then**
2:      Set $\bar{Y} = W^{1/2} Y D^{1/2}$;
3:      Compute singular value decomposition $[\Psi, \Sigma, \Phi] = \text{svd}(\bar{Y})$;
4:      Define $\bar{\Psi}_i$ as the i-th column of Ψ and $\sigma_i = \Sigma_{ii}$ for $1 \leq i \leq \ell$;
5:      Set $\Psi_i = W^{-1/2} \bar{\Psi}_i$ and $\lambda_i = \sigma_i^2$ for $i = 1, \ldots, \ell$;
6: **else if** `flag` $= 1$ **then**
7:      Compute eigenvalue decomposition $[\Psi, \Lambda] = \text{eig}(\bar{Y}\bar{Y}^\top)$;
8:      Define $\bar{\Psi}_i$ as the i-th column of Ψ and $\lambda_i = \Lambda_{ii}$ for $1 \leq i \leq \ell$;
9:      Set $\Psi_i = W^{-1/2} \bar{\Psi}_i$ for $i = 1, \ldots, \ell$;
10: **else if** `flag` $= 2$ **then**
11:      Compute eigenvalue decomposition $[\Phi, \Lambda] = \text{eig}(\bar{Y}^\top \bar{Y})$;
12:      Define $\bar{\phi}_i$ as the i-th column of Φ and $\lambda_i = \Lambda_{ii}$ for $1 \leq i \leq \ell$;
13:      Set $\Psi_i = Y D^{1/2} \bar{\phi}_i / \sqrt{\lambda_i}$ for $i = 1, \ldots, \ell$;
14: **end if**
15: **return** $\Psi = [\Psi_1 | \ldots | \Psi_\ell]$ and $\Lambda = [\lambda_1 | \ldots | \lambda_\ell]$;

2.2.3 The POD method for nonlinear evolution problems

In this subsection we explain the POD method for abstract nonlinear evolution problems. We focus on the numerical realization. For detailed theoretical investigations we refer the reader to [42, 50, 51, 57, 58], for instance.

2.2.3.1 Nonlinear evolution problems

Let us formulate the nonlinear evolution problem. For that purpose we suppose the following hypotheses.

Assumption 2.1. *Suppose that $T > 0$ holds, where $[0, T]$ is the considered finite time horizon.*

1) *V and H are real, separable Hilbert spaces and suppose that V is dense in H with compact embedding. By $\langle \cdot, \cdot \rangle_H$ and $\langle \cdot, \cdot \rangle_V$ we denote the inner products in H and V, respectively. We identify H with its dual (Hilbert) space H' by the Riesz isomorphism so that we have the Gelfand triple*

$$V \hookrightarrow H \simeq H' \hookrightarrow V',$$

where each embedding is continuous and dense. The last embedding is understood as follows: For every element $h \in H'$ and $v \in V$, we also have $v \in H$ by the embedding $V \hookrightarrow H$, so we can define $\langle h', v \rangle_{V', V} = \langle h', v \rangle_{H', H}$.

2) *For almost all $t \in [0, T]$ we define a time-dependent bilinear form $a(t; \cdot, \cdot) : V \times V \to \mathbb{R}$ satisfying*

$$|a(t; \varphi, \phi)| \le \gamma \|\varphi\|_V \|\phi\|_V \qquad \text{for all } \varphi, \phi \in V, \quad t \in [0, T] \text{ a. e.,} \qquad (2.15a)$$

$$a(t; \varphi, \varphi) \ge \gamma_1 \|\varphi\|_V^2 - \gamma_2 \|\varphi\|_H^2 \qquad \text{for all } \varphi \in V, \quad t \in [0, T] \text{ a. e.,} \qquad (2.15b)$$

for time-independent constants $\gamma, \gamma_2 \ge 0, \gamma_1 > 0$ and where "a. e." stands for "almost everywhere".

3) *Assume that $y_\circ \in V, f \in L^2(0, T; H)$ holds. Here we refer to [27, pp. 469-472] for vector-valued function spaces.*

Recall the function space

$$W(0, T) = \{\varphi \in L^2(0, T; V) \mid \varphi_t \in L^2(0, T; V')\},$$

which is a Hilbert space endowed with the standard inner product; cf. [27, pp. 472–479]. Furthermore, we have

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle \varphi(t), \phi \rangle_H = \langle \varphi_t(t), \phi \rangle_{V', V} \quad \text{for } \varphi \in W(0, T), \ \phi \in V$$

in the sense of distributions in $[0, T]$. Here, $\langle \cdot, \cdot \rangle_{V', V}$ stands for the dual pairing between V and its dual V'.

Now the evolution problem is given as follows: Find the state $y \in W(0, T) \cap C([0, T]; V)$ such that

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle y(t), \varphi \rangle_H + a(t; y(t), \varphi) + \langle \mathcal{N}(y(t)), \varphi \rangle_{V', V} = \langle f(t), \varphi \rangle_H$$

$$\forall \varphi \in V, t \in (0, T] \text{ a. e.,} \qquad (2.16)$$

$$\langle y(0), \varphi \rangle_H = \langle y_\circ, \varphi \rangle_H \quad \forall \varphi \in H.$$

Throughout we assume that (2.16) admits a unique solution $y \in W(0, T) \cap C([0, T]; V)$. Of course, this requires some properties for the nonlinear mapping $\mathcal{N}$ which we will not specify here.

Example 2.1 (Semi-linear heat equation). Let $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$, be a bounded open domain with Lipschitz-continuous boundary $\partial\Omega$ and let $T > 0$ be a fixed end time. We set $Q := (0, T) \times \Omega$ and $\Sigma := (0, T) \times \partial\Omega$ and $c \geq 0$. For a given forcing term $f \in L^2(Q)$ and initial condition $y_\circ \in L^2(\Omega)$, we consider the semi-linear heat equation with homogeneous Dirichlet boundary condition:

$$
\left.
\begin{aligned}
y_t(t, \mathbf{x}) - \Delta y(t, \mathbf{x}) + cy^3(t, \mathbf{x}) &= f(t, \mathbf{x}) && \text{in } Q, \\
y(t, \mathbf{x}) &= 0 && \text{on } \Sigma, \\
y(0, \mathbf{x}) &= y_\circ(\mathbf{x}) && \text{in } \Omega.
\end{aligned}
\right\}
\tag{2.17}
$$

The existence of a unique solution to (2.17) is proved in [83], for example. We can write (2.17) as an abstract evolution problem of type (2.16) by deriving a variational formulation for (2.17) with $V = H_0^1(\Omega)$ as the space of test functions $H = L^2(\Omega)$ and integrating over the space Ω. The bilinear form $a : V \times V \to \mathbb{R}$ is introduced by

$$
a(\varphi, \phi) = \int_\Omega \nabla\varphi \cdot \nabla\phi \, d\mathbf{x} \quad \text{for } \varphi, \phi \in V
$$

and the operator $\mathcal{N} : V \to V'$ is defined as $\mathcal{N}(\varphi) = c\varphi^3$ for $\varphi \in V$. For $c \equiv 0$, the heat equation (2.17) is linear.

Example 2.2 (Cahn–Hilliard equations). Let Ω, T, Q and Σ be defined as in Example 2.1. The Cahn–Hilliard system was proposed in [21] as a model for phase separation in binary alloys. Introducing the chemical potential w, the Cahn–Hilliard equations can be formulated in the common setting as a coupled system for the phase field c and the chemical potential w:

$$
\left.
\begin{aligned}
c_t(t, \mathbf{x}) + y \cdot \nabla c(t, \mathbf{x}) &= m\Delta w(t, \mathbf{x}) && \text{in } Q, \\
w(t, \mathbf{x}) &= -\sigma\varepsilon\Delta c(t, \mathbf{x}) + \frac{\sigma}{\varepsilon}W'(c(t, \mathbf{x})) && \text{in } Q, \\
\nabla c(t, \mathbf{x}) \cdot \nu_\Omega &= \nabla w(t, \mathbf{x}) \cdot \nu_\Omega = 0 && \text{on } \Sigma, \\
c(0, x) &= c_\circ(\mathbf{x}) && \text{in } \Omega.
\end{aligned}
\right\}
\tag{2.18}
$$

By ν_Ω we denote the outward normal on $\partial\Omega$, $m \geq 0$ is a constant mobility, $\sigma > 0$ denotes the surface tension, and $0 < \varepsilon \ll 1$ represents the interface parameter. Note that the convective term $y \cdot \nabla c$ describes the transport with (constant) velocity y. The transport term represents the coupling to the Navier–Stokes equations in the context of multiphase flow; see, e. g., [52] and [2]. The phase field function c describes the phase of a binary material with components A and B and takes the values $c \equiv -1$ in the pure A-phase and $c \equiv +1$ in the pure B-phase. The interfacial region is described

by $c \in (-1, 1)$ and admits a thickness of order $\mathcal{O}(\varepsilon)$; see, e. g., Figure 2.5, left column, where the binary phases are colored in blue and green, respectively, and the interfacial region is depicted in white. The function $W(c)$ represents the free energy and is of double well-type. A typical choice for W is the polynomial free energy function

$$W^p(c) = (1 - c^2)^2/4 \tag{2.19}$$

with two minima at $c = \pm 1$, which describe the energetically favorable states. It is infinitely often differentiable. Another choice for W is the C^1 relaxed double obstacle free energy

$$W_s^{\text{rel}}(c) = \frac{1}{2}(1 - c^2) + \frac{s}{2}\left(\max(c - 1, 0)^2 + \min(c + 1, 0)^2\right), \tag{2.20}$$

with relaxation parameter $s \gg 0$, which is introduced in [43] as the Moreau–Yosida relaxation of the double obstacle free energy

$$W^\infty(c) = \begin{cases} \frac{1}{2}(1 - c^2), & \text{if } c \in [-1, 1], \\ +\infty, & \text{otherwise.} \end{cases}$$

The energies $W^p(c)$ and $W_s^{\text{rel}}(c)$ later will be used to compare the performance of POD on systems with smooth and less smooth nonlinearities. For more details on the choices for W we refer to [1] and [19], for example. Concerning existence, uniqueness, and regularity of a solution to (2.18), we refer to [19]. In order to derive a variational form of type (2.16), we rewrite (2.18) as a single fourth-order parabolic equation for c by

$$\left.\begin{aligned} c_t(t, \boldsymbol{x}) + y \cdot \nabla c(t, \boldsymbol{x}) &= \mathsf{m}\Delta\left(-\sigma\varepsilon\Delta c(t, \boldsymbol{x}) + \frac{\sigma}{\varepsilon}W'(c(t, \boldsymbol{x}))\right) &&\text{in } Q, \\ 0 = \nabla c(t, \boldsymbol{x}) \cdot \nu_\Omega &= \nabla\left(-\sigma\varepsilon\Delta c(t, \boldsymbol{x}) + \frac{\sigma}{\varepsilon}W'(c(t, \boldsymbol{x}))\right) \cdot \nu_\Omega &&\text{on } \Sigma, \\ c(0, \boldsymbol{x}) &= c_\circ(\boldsymbol{x}) &&\text{in } \Omega. \end{aligned}\right\} \tag{2.21}$$

We choose $V = \{v \in H^1(\Omega) : \frac{1}{|\Omega|}\int_\Omega v = 0\}$ equipped with the inner product $(u, v)_V := \int_\Omega \nabla u \nabla v$, so that the dual space of V is given by $V' = \{f \in (H^1(\Omega))' : \langle f, 1 \rangle = 0\}$ such that $V \hookrightarrow H = V'$ and $\langle ., . \rangle$ denotes the duality pairing. We note that $(V, (., .)_V)$ is a Hilbert space. We define the V'-inner product for $f, g \in V'$ as $(f, g)_{V'} := \int_\Omega \nabla(-\Delta)^{-1}f \cdot \nabla(-\Delta)^{-1}g$ where $(-\Delta)^{-1}$ denotes the inverse of the negative Laplacian with zero Neumann boundary data. Note that $(f, g)_{V'} = (f, (-\Delta)^{-1}g)_{L^2(\Omega)} = ((-\Delta)^{-1}f, g)_{L^2(\Omega)}$. We introduce the bilinear form $a : V \times V \to \mathbb{R}$ by

$$a(u, v) = \sigma\varepsilon(\nabla u, \nabla v)_{L^2(\Omega)} + \frac{1}{\mathsf{m}}(y \cdot \nabla u, v)_{V'}$$

and define the nonlinear operator $\mathcal{N}$ by $\mathcal{N}(c) = \frac{\sigma}{\varepsilon}W'(c)$. The evolution problem can be written in the form

$$\frac{1}{\mathrm{m}}\big(c_t(t), v\big)_{V'} + a\big(c(t), v\big) + \big\langle \mathcal{N}(c(t)), v \big\rangle = 0 \quad \forall v \in V \text{ and a. a. } t \in (0, T].^{[1]} \qquad (2.22)$$

We note that this fits our abstract setting formulated in (2.16) with the Gelfand triple $V \hookrightarrow H \equiv V' \hookrightarrow V'$.

2.2.3.2 Temporal discretization and the POD method

Let $0 = t_1 < \cdots < t_{n_t} = T$ be a given time grid with step sizes $\Delta t_j = t_j - t_{j-1}$ for $j = 2, \ldots, n_t$. Suppose that for any $j \in \{1, \ldots, n_t\}$ the element $y_j \in V \subset H$ is an approximation of $y(t_j)$ computed by applying a temporal integration method (e. g., the implicit Euler method) to (2.16). Then we consider the snapshot ensemble

$$\mathcal{V} = \operatorname{span}\{y_j \mid 1 \leq j \leq n\} \subset V \subset H$$

with $n = n_t$ and $\mathrm{r} = \dim \mathcal{V} \leq n$. In the context of $(\mathbf{P}^\ell)$ we choose $K = 1$ and $n = n_1 = n_t$. Moreover, X can be either V or H. For the weighting parameters we take the trapezoidal weights

$$\alpha_1 = \frac{\Delta t_1}{2}, \quad \alpha_j = \frac{\Delta t_j + \Delta t_{j-1}}{2} \text{ for } j = 2, \ldots, n_t - 1, \quad \alpha_{n_t} = \frac{\Delta t_{n_t}}{2}. \qquad (2.23)$$

Of course, other quadrature weights are also possible. Now, instead of $(\mathbf{P}^\ell)$ we consider the maximization problem

$$\left.\begin{aligned} &\max \sum_{i=1}^{\ell} \sum_{j=1}^{n} \alpha_j \,\big|\langle y_j, \Psi_i \rangle_X\big|^2 \\ &\text{s. t. } \{\Psi_i\}_{i=1}^{\ell} \subset X \text{ and } \langle \Psi_i, \Psi_j \rangle_X = \delta_{ij},\ 1 \leq i, j \leq \ell \end{aligned}\right\} \qquad (2.24)$$

with either $X = V$ or $X = H$.

Remark 2.2. In [42, Sections 1.2.2 and 1.3.2] a continuous variant of the POD method is considered. In that case the trapezoidal approximation in (2.24) is replaced by integrals over the time interval $[0, T]$. More precisely, we consider

$$\left.\begin{aligned} &\max \sum_{i=1}^{\ell} \int_0^T \big|\langle y(t), \Psi_i \rangle_X\big|^2 \, \mathrm{d}t \\ &\text{s. t. } \{\Psi_i\}_{i=1}^{\ell} \subset X \text{ and } \langle \Psi_i, \Psi_j \rangle_X = \delta_{ij},\ 1 \leq i, j \leq \ell \end{aligned}\right\} \qquad (2.25)$$

---

1 We acknowledge a hint of Harald Garcke who pointed this form of the weak formulation of (2.21) to us.

with either $X = V$ or $X = H$. For the relationship between solutions to (2.24) and (2.25) we refer to [58] and [42, Section 1.2.3].

To compute the POD basis $\{\Psi_i\}_{i=1}^{\ell}$ of rank ℓ we have to evaluate the inner products $\langle y_j, \Psi_i \rangle_X$, where either $X = V$ or $X = H$ holds. In typical applications the space X is usually infinite-dimensional. Therefore, a discretization of X is required in order to get a POD method that can be realized on a computer. This is the topic of the next subsection.

2.2.3.3 Galerkin discretization

We discretize the state equation by applying any spatial approximation scheme. Let us consider here a Galerkin scheme for (2.16). For this reason we are given linearly independent elements $\varphi_1, \ldots, \varphi_m \in V$ and define the m-dimensional subspace

$$V^h = \operatorname{span}\{\varphi_1, \ldots, \varphi_m\} \subset V$$

endowed with the V topology. Then a Galerkin scheme for (2.16) is given as follows: Find $y^h \in W(0, T) \cap C([0, T]; V^h)$ satisfying

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle y^h(t), \varphi^h \rangle_H + a(t; y^h(t), \varphi^h) + \langle \mathcal{N}(y^h(t)), \varphi \rangle_{V', V}$$
$$= \langle f(t), \varphi^h \rangle_H \quad \forall \varphi^h \in V^h, \ t \in (0, T] \text{ a. e.,} \tag{2.26}$$
$$\langle y^h(0), \varphi^h \rangle_H = \langle y_\circ, \varphi^h \rangle_H \quad \forall \varphi^h \in V^h.$$

Inserting the representation $y^h(t) = \sum_{i=1}^{m} y_i^h(t)\varphi_i \in V^h, t \in [0, T]$, in (2.26) and choosing $\varphi^h = \varphi_i$ for $i = 1, \ldots, m$ we derive the following m-dimensional initial value problem:

$$M^h \dot{y}^h(t) + A^h(t) y^h(t) + N^h(y^h(t)) = F^h(t) \quad \text{for } t \in (0, T],$$
$$M^h y^h(0) = y_\circ^h, \tag{2.27}$$

where we have used the matrices and vectors

$$M^h = ((\langle \varphi_j, \varphi_i \rangle_H))_{1 \le i, j \le m}, \qquad y^h(t) = (y_i^h)_{1 \le i \le m} \quad \text{for } t \in [0, T] \text{ a. e.,}$$
$$A^h(t) = ((a(t; \varphi_j, \varphi_i)))_{1 \le i, j \le m}, \quad y_\circ^h = (\langle y_\circ, \varphi_i \rangle_H)_{1 \le i \le m},$$
$$N^h(v) = \left(\left\langle \mathcal{N}\left(\sum_{j=1}^{m} v_j \varphi_j \right), \varphi_i \right\rangle_{V', V} \right)_{1 \le i \le m} \quad \text{for } v = (v_j)_{1 \le j \le m},$$
$$F^h(t) = (\langle f(t), \varphi_i \rangle_H)_{1 \le i \le m} \quad \text{for } t \in [0, T].$$

In the pseudo-code **function** $[Y] = \texttt{StateSol}(y_\circ^h)$ we present a solution method for (2.27) using the implicit Euler method.

In the next subsection we discuss how a POD basis $\{\Psi_j\}_{j=1}^{\ell}$ of rank $\ell \le r$ can be computed from numerical approximations for the solution y^h to (2.27).

function $[Y]=\texttt{StateSol}(\mathrm{y}_\circ^h~)$

Require: Initial condition $\mathrm{y}_\circ^h$;
  1: Compute $\mathrm{y}_1^h \in \mathbb{R}^m$ solving $\mathrm{M}^h \mathrm{y}_1^h = \mathrm{y}_\circ^h$;
  2: **for** $j = 2$ **to** n_t **do**
  3:   Set $\mathrm{A}_j^h = \mathrm{A}^h(t_j) \in \mathbb{R}^{m \times m}$ and $\mathrm{F}_j^h = \mathrm{F}^h(t_j) \in \mathbb{R}^m$;
  4:   Solve (e. g., by applying Newton's method) for $\mathrm{y}_j^h \in \mathbb{R}^m$

$$(\mathrm{M}^h + \Delta t_j \mathrm{A}_j^h)\mathrm{y}_j^h + \Delta t_j \mathrm{N}^h(\mathrm{y}_j^h) = \mathrm{M}^h \mathrm{y}_{j-1}^h + \Delta t_j \mathrm{F}_j^h;$$

  5: **end for**
  6: **return** matrix $Y = [\mathrm{y}_1^h \,|\ldots|\, \mathrm{y}_{n_t}^h] \in \mathbb{R}^{m \times n_t}$;

2.2.3.4 POD method for the fully discretized nonlinear evolution problem

Recall that we have introduced the temporal grid $\{t_j\}_{j=1}^{n_t} \subset [0, T]$ and set $n = n_t$. Let $y_1^h, \ldots, y_n^h \in V^h$ be numerical approximations to the solution $y^h(t)$ to (2.27) at time instances $t = t_j$, $j = 1 \ldots, n_t$. Then, a coefficient matrix $Y \in \mathbb{R}^{m \times n}$ is defined by the elements Y_{ij} given by

$$y_j^h = \sum_{i=1}^m Y_{ij}\varphi_i \in V^h \quad \text{for } 1 \le j \le n.$$

The j-th column of Y (denoted by $\mathrm{y}_j = Y_{\cdot,j}$) contains the Galerkin coefficients of the snapshot $y_j^h \in V^h$. We set $\mathrm{r} = \mathrm{rank}\, Y \le \min(m, n)$ and

$$\mathcal{V}^h = \mathrm{span}\,\{y_j^h \,|\, 1 \le j \le n\} \subset V^h.$$

Due to $\mathcal{V}^h \subset V^h$ we have $\Psi_j \in V^h$ for $1 \le j \le \ell$. Therefore, there exists a coefficient matrix $\Psi \in \mathbb{R}^{m \times \ell}$ that is defined by the elements Ψ_{ij} satisfying

$$\Psi_j = \sum_{i=1}^m \Psi_{ij}\varphi_i \in V^h \quad \text{for } 1 \le j \le \ell,$$

where the j-th column $\Psi_{\cdot,j}$ of the matrix Ψ consists of the Galerkin coefficients of the element Ψ_j. Note that

$$\langle v^h, w^h \rangle_H = (\mathrm{v}^h)^\top \mathrm{M}^h \mathrm{w}^h, \quad \langle v^h, w^h \rangle_H = (\mathrm{v}^h)^\top \mathrm{S}^h \mathrm{w}^h$$

hold for $v^h = \sum_{i=1}^m \mathrm{v}_i^h \varphi_i$, $w^h = \sum_{i=1}^m \mathrm{w}_i^h \varphi_i \in V^h$ and for the symmetric, positive definite stiffness matrix

$$\mathrm{S}^h = ((\langle \varphi_j, \varphi_i \rangle_V))_{1 \le i, j \le m}.$$

Then, we have for $X = H$

$$\langle y_j^h, \Psi_i \rangle_X = y_j^\top M^h \Psi_{.,i} = Y_{.,j}^\top M^h \Psi_{.,i} \quad \text{for } 1 \le j \le n, \, 1 \le i \le \ell,$$

and for $X = V$

$$\langle y_j^h, \Psi_i \rangle_X = y_j^\top S^h \Psi_{.,i} = Y_{.,j}^\top S^h \Psi_{.,i} \quad \text{for } 1 \le j \le n, \, 1 \le i \le \ell.$$

Thus, we can apply the approach presented in Section 2.2.2 choosing $W = M^h$ for $X = H$ and $W = S^h$ for $X = V$. Moreover, we set $K = 1$, $n_1 = n_t = n$, and $\alpha_j^1 = \alpha_j$ defined in (2.23). Now a POD basis of rank ℓ for (2.27) can be computed by the pseudo-code **function** $[Y,\Psi]$=PODState($y_\circ^h$, W, D, ℓ, flag).

---

function $[Y, \Psi]$=PODState($y_\circ^h$, W, D, ℓ, flag)

Require: Initial condition $y_\circ^h$, weighting matrices W, D, number ℓ of POD functions, and flag for the solver;
 1: Call $[Y]$=StateSol($y_\circ^h$);
 2: Call $[\Psi,\Lambda]$=POD(Y, W, D, ℓ, flag);
 3: **return** $Y = [y_1^h | \ldots | y_{n_t}^h]$ and $\Psi = [\Psi_1 | \ldots | \Psi_\ell]$;

In the next subsection we will discuss in detail how the POD method has to be applied in that case if we have – instead of V^h – different spaces V^{h_j} for each $j = 1, \ldots, n$.

2.2.4 The POD method with snapshots generated by spatially adaptive finite element methods

In practical applications it often is desirable to provide POD models for time-dependent PDE systems, whose numerical treatment requires adaptive numerical techniques in space and/or time. Snapshots generated by those methods are not directly amenable to the POD procedure described in Section 2.2.3.4, since the application of spatial adaptivity means that the snapshots at each time instance may have different lengths due to their different spatial resolutions. In fact, there is not one single discrete Galerkin space V^h for all snapshots generated by the fully discrete evolution, but at every time instance t_j the adaptive procedure generates a discrete Galerkin space $V^{h_j} \subset X$, so that in this case $y_j^h \equiv y_j^{h_j} \in V^{h_j}$. For this reason, no snapshot matrix Y can be formed with columns containing the basis coefficient vectors of the snapshots.

To obtain also a POD basis in this situation we inspect the operator $\mathcal{K}$ of (2.8) and observe that its action can be computed if the inner products $\langle y_j^k, y_i^l \rangle_X$ can be evaluated for all $1 \le i \le n_l$, $1 \le j \le n_k$ and $1 \le k, l \le K$.

Let us next demonstrate how to compute a POD basis for snapshots residing in arbitrary finite element (FE) spaces. To begin with we drop the superindex h and set $V_j := V^{h_j}$. For each time instant $j = 1, \ldots, n$ of our time-discrete PDE system the snapshots $\{y_j\}_{j=1}^n$ are taken from different finite element spaces $V_j \subseteq X$ ($j = 1, \ldots, n$), where X denotes a common (real) Hilbert space. Let $V_j = \mathrm{span}\,\{\varphi_1^j, \ldots, \varphi_{m_j}^j\}$. Then we have the expansions

$$y_j = \sum_{i=1}^{m_j} \mathrm{y}_j^i \varphi_i^j \quad \text{for } j = 1, \ldots, n \tag{2.28}$$

with coefficient vectors

$$\mathrm{y}_j = (\mathrm{y}_j^i) \in \mathbb{R}^{m_j} \quad \text{for } j = 1, \ldots, n$$

containing the finite element coefficients. The inner product of the associated functions can thus be computed as

$$\langle y_i, y_j \rangle_X = \sum_{k=1}^{m_i} \sum_{l=1}^{m_j} \mathrm{y}_i^k \mathrm{y}_j^l \langle \varphi_k^i, \varphi_l^j \rangle_X \quad \text{for } i, j = 1, \ldots, n,$$

so that the evaluation of the action $\mathcal{K}\Phi$ only relies on the evaluation of the inner products $\langle \varphi_k^i, \varphi_l^j \rangle_X$ ($1 \le i, j \le n$, $1 \le k \le m_i$, $1 \le l \le m_j$). In other words, once we are able to compute those inner products we are in the position to set up the eigensystem $\{(\lambda_i, \Phi_i)\}_{i=1}^r$ of $\mathcal{K}$ from (2.8). The POD modes $\{\Psi_i\}_{i=1}^r$ can then be computed according to (2.9) by

$$\Psi_i = \frac{1}{\sqrt{\lambda_i}} \mathcal{Y}\Phi_i \quad \text{for } i = 1, \ldots, r.$$

Details on this procedure can be found in [62, 39].

To illustrate how this procedure can be implemented we summarize Examples 6.1–6.3 from [39], which deal with nested and nonnested meshes. All coding was done in C++ with FEniCS [8, 66] for the solution of the differential equations and ALBERTA [87] for dealing with hierarchical meshes. The numerical tests were run on a compute server with 512 GB RAM.

Run 2.1 ([39, Example 6.1]). We consider Example 2.1 with homogeneous Dirichlet boundary condition and vanishing nonlinearity, i. e., we set $c \equiv 0$ so that the equation becomes linear. The spatial domain is chosen as $\Omega = (0, 1) \times (0, 1) \subset \mathbb{R}^2$, the time interval is $[0, T] = [0, 1.57]$. Furthermore, we choose $X = L^2(\Omega)$. For the temporal discretization we introduce the uniform time grid by

$$t_j = (j - 1)\Delta t \quad \text{for } j = 1, \ldots, n_t = 1571$$

with $\Delta t = 0.001$. For the spatial discretization we use h-adapted piecewise linear, continuous finite elements on hierarchical and nested meshes. Snapshots of the analytical solution at three different time points are shown in Figure 2.1. Details on the construction of the analytical solution and the corresponding right-hand side f are given in [39, Example 6.1].
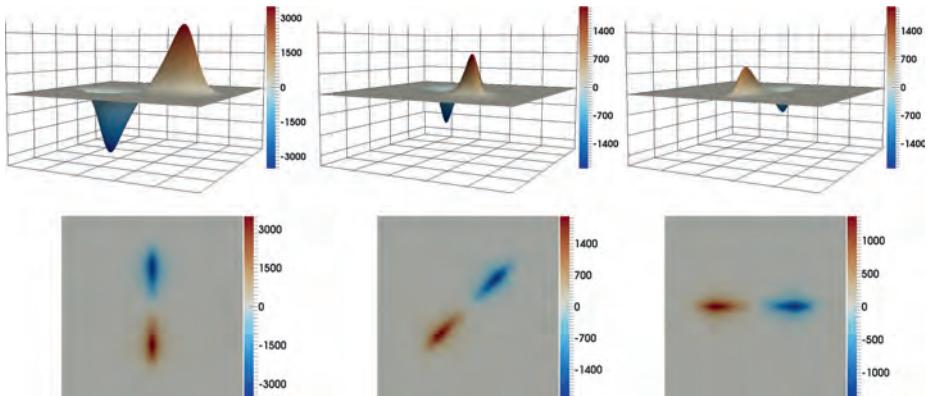
Figure 2.1: Run 2.1. Surface plot (top) and view from above (bottom) of the analytical solution of (2.17) at $t = t_1$ (left), $t = T/2$ (middle), and $t = T$ (right).

Due to the steep gradients in the neighborhoods of the minimum and the maximum, the use of an adaptive finite element discretization is justified. The resulting computational meshes as well as the corresponding finest mesh (reference mesh at the end of the simulation which is the union of all adaptive meshes generated during the simulation) are shown in Figure 2.2.
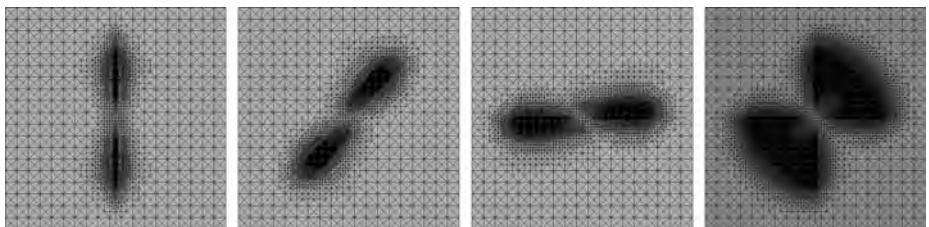
Figure 2.2: Run 2.1. Adaptive finite element meshes at $t = t_1$ (left), $t = T/2$ (middle left), and $t = T$ (middle right), and finest mesh (right).

The number of nodes of the adaptive meshes varies between 3,637 and 7,071 points. The finest mesh has 18,628 degrees of freedom. A uniform mesh with grid size of order of the diameter of the smallest triangles in the adaptive grids ($h_{min} = 0.0047$) would have 93,025 degrees of freedom. This clearly reveals the benefit of using adap-

tive meshes for snapshot generation, which is also well reflected in the comparison of the computational times needed for the snapshot generation on the adaptive mesh taking 944 seconds compared to 8,808 seconds on the uniform mesh (see Table 2.4 for the speedup factors obtained by spatial adaptation). In Figure 2.3, the resulting normalized eigenspectrum of the correlation matrix K is shown for snapshots obtained by uniform spatial discretization ("uniform FE mesh"), for snapshots obtained by interpolation on the finest mesh ("adaptive FE mesh"), and for snapshots without interpolation ("infPOD"), where K is associated to the operator $\mathcal{K}$ from (2.8); see also (2.30).
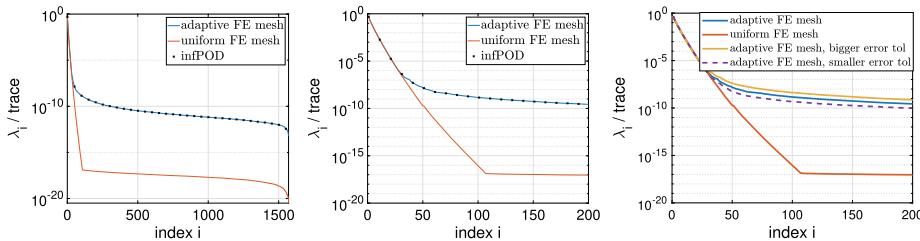
Figure 2.3: Run 2.1. Comparison of the normalized eigenvalues using an adaptive and a uniform spatial mesh, respectively. Left: all eigenvalues; middle: first 200 largest eigenvalues; right: first 200 largest eigenvalues with different error tolerances for the adaptivity (1.5 times bigger and smaller error tolerances, respectively).

We observe that the eigenvalues for both adaptive approaches coincide. This numerically validates what we expect from theory: The information content which is contained in the matrix $\mathcal{K}$ when we explicitly compute the entries without interpolation is the same as the information content contained within the eigenvalue problem which is formulated when using the finest mesh. No information is added or lost. Moreover, we recognize that about the first 28 eigenvalues computed corresponding to the adaptive simulation coincide with the simulation on a uniform mesh. From index 29 on, the methods deliver different results: For the uniform discretizations, the normalized eigenvalues fall below machine precision at around index 100 and stagnate. On the contrary, the normalized eigenvalues for both adaptive approaches flatten in the order around 10^{-10}. If the error tolerance for the spatial discretization error is chosen larger (or smaller), the stagnation of the eigenvalues in the adaptive method takes place at a higher (or lower) order (Figure 2.3, right). Concerning dynamical systems, the magnitude of the eigenvalue corresponds to the characteristic properties of the underlying dynamical system: the larger the eigenvalue, the more information is contained in the corresponding eigenfunction. Since all adaptive meshes are contained in the uniform mesh, the difference in the amplitude of the eigenvalues is due to the interpolation errors during refinement and coarsening. This is the price we have to pay for faster snapshot generation using adaptive methods. A further aspect gained from the decay

behavior of the eigenvalues in the adaptive case is the following: The adaptive approach filters out the noise in the system which is related to the modes corresponding to the singular values that are not matched by the eigenvalues of the adaptive approach. This in the language of frequencies means that the overtones in the systems which get lost in the adaptive computations live in the space which is neglected by the POD method based on adaptive finite element snapshots. From this point of view, adaptivity can be interpreted as a smoother.

The first, second, and fifth POD modes of Run 2.1 obtained by the adaptive approach are depicted in Figure 2.4. We observe the classical appearance of the basis functions. The initial condition is reflected by the first POD basis function. The next basis functions admit a number of minima and maxima corresponding to the index in the basis: Ψ_2 has two minima and two maxima, etc. This behavior is similar to the increasing oscillations in higher frequencies in trigonometric approximations. The POD basis functions corresponding to the uniform spatial discretization have a similar appearance.
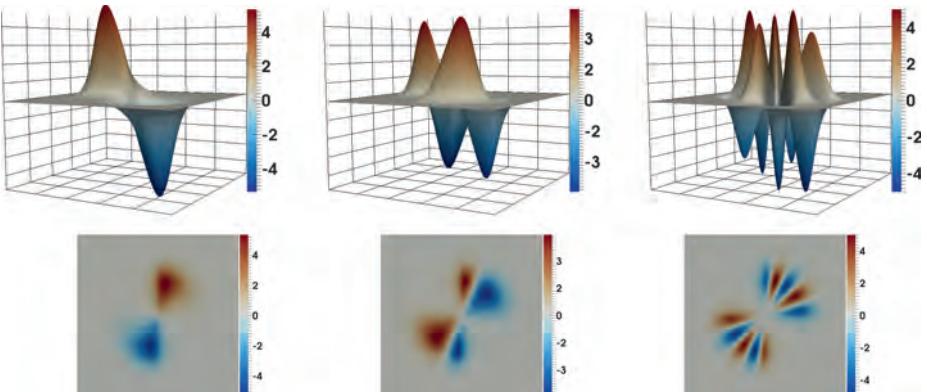
Figure 2.4: Run 2.1. Surface plot (top) and view from above (bottom) of the POD basis functions Ψ_1 (left), Ψ_2 (middle), and Ψ_5 (right).

Run 2.2 ([39, Example 6.2]). (Cahn–Hilliard system). We consider Example 2.2 in the form (2.18) with $\Omega = (0, 1.5) \times (0, 0.75)$, $T = 0.025$, constant mobility $m \equiv 0.00002$, and constant surface tension $\sigma \equiv 24.5$. The interface parameter ε is set to $\varepsilon = 0.02$, with resulting interface thickness $\pi \cdot \varepsilon \approx 0.0628$. We use the relaxed double obstacle free energy W_s^{rel} from (2.20) with $s = 10^4$. As initial condition, we choose a circle with radius $r = 0.25$ and center $(0.375, 0.375)$. The initial condition is transported horizontally with constant velocity $v = (30, 0)^T$. We set

$$t_j = (j-1)\Delta t \quad \text{for } j = 1, \ldots, n_t = 1001,$$

so that $\Delta t = 2.5 \cdot 10^{-5}$. The numerical computations are performed with the semi-implicit Euler scheme. For this purpose let $c^{j-1} \in V$ and $c^j \in V$ denote the time-discrete solution at t_{j-1} and t_j, respectively. Based on the variational formulation (2.22) we tackle the time discrete version of (2.18) in the following form: Given c^{j-1}, find c^j, w^j solving

$$\left. \begin{aligned} \frac{1}{\Delta t} \langle c^j - c^{j-1}, \varphi_1 \rangle_{L^2} + \langle v \cdot \nabla c^{j-1}, \varphi_1 \rangle_{L^2} + m \langle \nabla w^j, \nabla \varphi_1 \rangle_{L^2} &= 0, \\ -\langle w^j, \varphi_2 \rangle_{L^2} + \sigma\varepsilon \langle \nabla c^j, \nabla \varphi_2 \rangle_{L^2} + \frac{\sigma}{\varepsilon} \langle W'_+(c^j) + W'_-(c^{j-1}), \varphi_2 \rangle_{L^2} &= 0 \end{aligned} \right\} \tag{2.29}$$

for all $\varphi_1, \varphi_2 \in V$ and $j = 2, \ldots, n_t$ with $c^1 = c_\circ$. According to (2.22), here it is $V = \{v \in H^1(\Omega), \frac{1}{|\Omega|} \int_\Omega v dx = 0\}$. Note that the free energy function W is split into a convex part W_+ and a concave part W_-, such that $W = W_+ + W_-$ and W'_+ is treated implicitly, whereas W'_- is treated explicitly with respect to time. This leads to an unconditionally energy-stable time marching scheme; compare [33]. The system (2.29) is discretized in space using piecewise linear and continuous finite elements. The resulting nonlinear equation systems are solved using a semi-smooth Newton method.

Figure 2.5 shows the phase field (left) and the chemical potential (right) for the finite element simulation using adaptive meshes. The initial condition $c_\circ$ is transported horizontally with constant velocity.
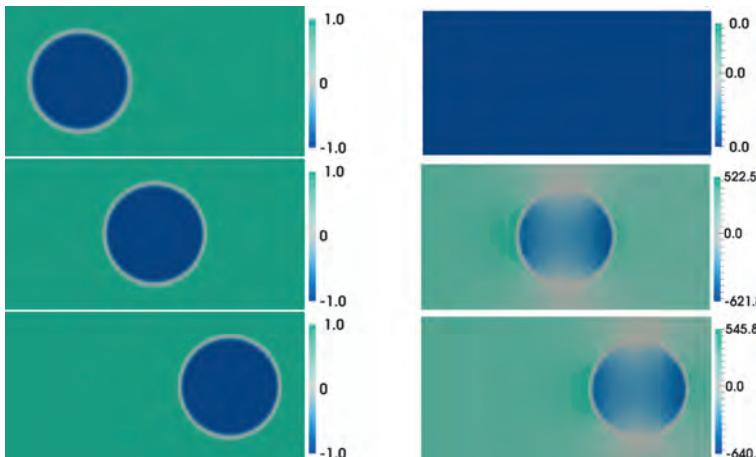
Figure 2.5: Run 2.2. Phase field c (left) and chemical potential w (right) computed on adaptive finite element meshes at $t = t_1$ (top), $t = T/2$ (middle), and $t = T$ (bottom).

The adaptive finite element meshes and the finest mesh which is generated during the adaptive finite element simulation are shown in Figure 2.6. The number of degrees of freedom in the adaptive meshes varies between 6,113 and 8,795. The finest mesh (overlay of all adaptive meshes) has 54,108 degrees of freedom, whereas a uniform mesh
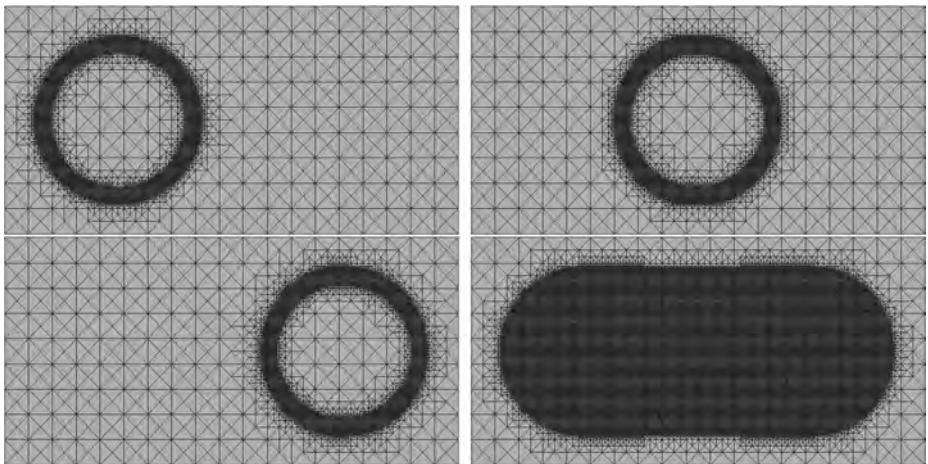
Figure 2.6: Run 2.2. Adaptive finite element meshes at $t = t_1$ (top left), $t = T/2$ (top right), and $t = T$ (bottom left) together with the finest mesh (bottom right).

with discretization fineness as small as the smallest triangle in the adaptive meshes has 88,450 degrees of freedom.

Figure 2.7 shows the first, second, and fifth POD modes for the phase field c and the chemical potential w. Analogously to Run 2.1, we observe a periodicity in the POD basis functions corresponding to their basis index numbers.
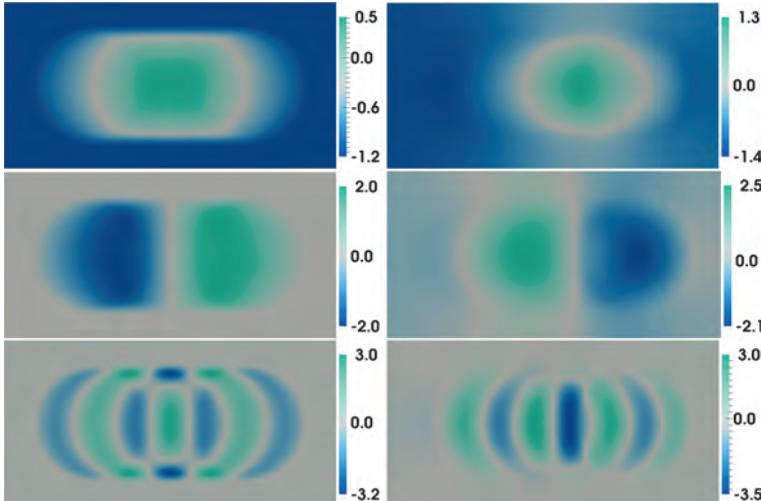
Figure 2.7: Run 2.2. First, second, and fifth POD modes for c (left) and w (right).

In the present example we only compare the POD procedure for two kinds of snapshot discretizations, namely, the adaptive approach with using a finest mesh and the uniform mesh approach, where the grid size is chosen to be of the same size as the smallest triangle in the adaptive meshes. We choose $X = L^2(\Omega)$ and compute a separate POD basis for each of the variables c and w.

In Figure 2.8, a comparison is visualized concerning the normalized eigenspectrum for the phase field c and the chemical potential w using uniform and adaptive finite element discretization. We note for the phase field c that about the first 180 eigenvalues computed corresponding to the adaptive simulation coincide with the eigenvalues of the simulation on the finest mesh. Then, the eigenvalues corresponding to the uniform simulation decay faster. Similar observations apply for the chemical potential w.
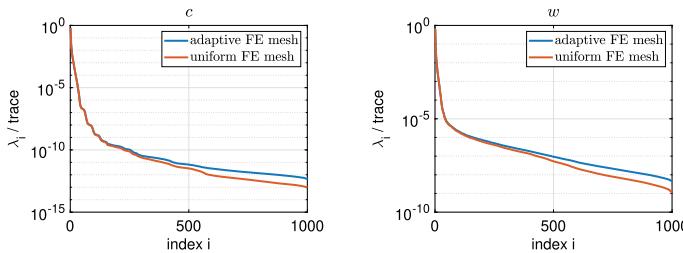
Figure 2.8: Run 2.2. Comparison of the normalized eigenvalues for the phase field c (left) and the chemical potential w (right) using an adaptive and a uniform spatial mesh, respectively.

We use the criterion (2.11) to determine the basis length ℓ which is required to represent a prescribed information content with the respective POD space. We will choose the POD basis length ℓ_c for the phase field c and the number of POD modes ℓ_w for the chemical potential, such that

$$\ell_{\min} = \arg\min \{\mathcal{E}(\ell) : \mathcal{E}(\ell) > 1 - p\}, \quad \text{with } \ell = \ell_c \text{ and } \ell_w, \text{ respectively,}$$

for a given value p representing the loss of information. Alternatively, the POD basis length could be chosen in alignment with the POD projection error (2.10) with the expected spatial and/or temporal discretization error; compare, e. g., [39, Theorem 5.1]. Let us also refer, e. g., to the recent paper [12], where different adaptive POD basis extension techniques are discussed. Table 2.1 summarizes how to choose ℓ_c and ℓ_w in order to capture a desired amount of information. Moreover, it tabulates the POD projection error (2.10) depending on the POD basis length, where λ_i^c and λ_i^w denote the eigenvalues for the phase field c and the chemical potential w, respectively. The results in Table 2.1 agree with our expectations: the smaller the loss of information p is, the more POD modes are needed and the smaller is the POD projection error.

Table 2.1: Run 2.2. Number of needed POD bases in order to achieve a loss of information below the tolerance p using adaptive finite element meshes (columns 2–5) and uniform finite element discretization (columns 6–9) and POD projection error.

p	ℓ_c^{ad}	$\sum_{i>\ell} \lambda_i^c$	ℓ_w^{ad}	$\sum_{i>\ell} \lambda_i^w$	ℓ_c^{uni}	$\sum_{i>\ell} \lambda_i^c$	ℓ_w^{uni}	$\sum_{i>\ell} \lambda_i^w$
10^{-1}	3	$2.0 \cdot 10^{-3}$	4	$156.9 \cdot 10^0$	3	$2.0 \cdot 10^{-3}$	4	$157.6 \cdot 10^0$
10^{-2}	10	$2.1 \cdot 10^{-4}$	13	$15.8 \cdot 10^0$	10	$2.1 \cdot 10^{-4}$	13	$15.6 \cdot 10^0$
10^{-3}	19	$2.5 \cdot 10^{-5}$	26	$1.8 \cdot 10^0$	19	$2.5 \cdot 10^{-5}$	25	$1.8 \cdot 10^0$
10^{-4}	29	$2.0 \cdot 10^{-6}$	211	$1.8 \cdot 10^{-1}$	28	$2.6 \cdot 10^{-6}$	160	$1.9 \cdot 10^{-1}$
10^{-5}	37	$2.5 \cdot 10^{-7}$	644	$1.1 \cdot 10^{-2}$	37	$2.4 \cdot 10^{-7}$	419	$2.5 \cdot 10^{-2}$

Run 2.3 ([39, Example 6.3]). (Linear heat equation revisited). We again consider Example 2.1 with $c \equiv 0$. The purpose of this example is to confirm that our POD approach also is applicable in the case of nonnested meshes like it appears in the case of r-adaptivity, for example. We set up the matrix K for snapshots generated on sequences of nonnested spatial discretizations. This requires the integration over cut elements; see [39]. We choose $\Omega = (0,1) \times (0,1) \subset \mathbb{R}^2$, $[0,T] = [0,1]$, and we apply a uniform temporal discretization with time step size $\Delta t = 0.01$. The analytical solution in the present example is given by

$$y(t, \boldsymbol{x}) = \sin(\pi x_0) \cdot \sin(\pi x_1) \cdot \cos(2\pi t x_0),$$

with $\boldsymbol{x} = (x_0, x_1)$, source term $f := y_t - \Delta y$, and the initial condition $g := y(0, \cdot)$. The initial condition is discretized using piecewise linear and continuous finite elements on a uniform spatial mesh, which is shown in Figure 2.9 (left). Then, at each time step, the mesh is disturbed by relocating each mesh node according to the assignment

$$\begin{aligned} x_0 &\leftarrow x_0 + \theta \cdot x_0 \cdot (x_0 - 1) \cdot (\Delta t/10), \\ x_1 &\leftarrow x_1 + \theta \cdot 0.5 \cdot x_1 \cdot (x_1 - 1) \cdot (\Delta t/10), \end{aligned}$$

where $\theta \in \mathbb{R}_+$ is sufficiently small such that all coordinates of the interior nodes fulfill $0 < x_0 < 1$ and $0 < x_1 < 1$. After relocating the mesh nodes, the heat equation is solved on this mesh for the next time instance. We use Lagrange interpolation to transfer the finite element solution of the previous time step onto the new mesh. The disturbed meshes at $t = 0.5$ and $t = 1.0$ and an overlap of two meshes are shown in Figure 2.9. To compute the matrix K from (2.30) we have to evaluate the corresponding inner products of the snapshots, where we need to integrate over cut elements.

We compute the eigenvalue decomposition of the matrix representation K of the operator $\mathcal{K}$ (cf. (2.30)) for different values of θ and compare the results with a uniform mesh (i. e., $\theta = 0$) in Figure 2.10. We note that the eigenvalues of the disturbed mesh are converging to the eigenvalues of the uniform mesh for $\theta \to 0$. As expected, the eigenvalue spectrum depends only weakly on the underlying mesh given that the mesh size is sufficiently small. Concerning the computational complexity of POD with nonnested
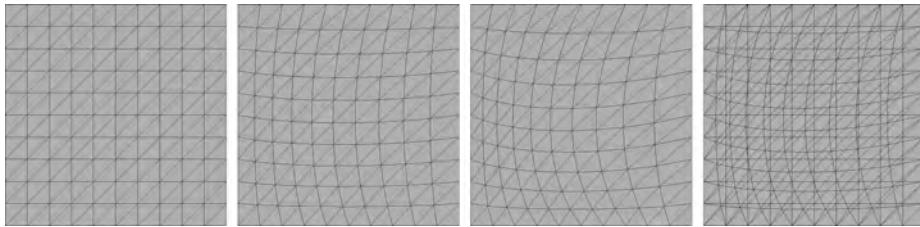
Figure 2.9: Run 2.3. Uniform mesh (left), disturbed meshes at $t = 0.5$ and $t = 1.0$ (middle left, middle right), and overlap of the mesh at $t = 0$ with the mesh at $t = 1.0$ (right). Here, we use $\theta = 10$.
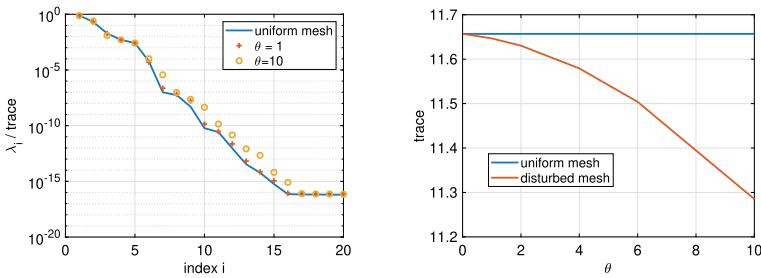
Figure 2.10: Run 2.3: Decay of eigenvalues of matrix K with different meshes.

meshes let us note that solving the heat equation takes 2.1 seconds on the disturbed meshes and 1.8 seconds on the uniform mesh. The computational time needed to compute each entry of the matrix K is 0.022 seconds and computing the eigenvalue decomposition for K takes 0.0056 seconds. Note that the cut element integration problem for each matrix entry takes a fraction of the time required to solve the finite element problem.

2.3 The POD-Galerkin procedure

Once the POD basis is generated it can be used to set up a POD-Galerkin approximation of the original dynamical system. This is discussed in the present section. In this context we recall that the space spanned by the POD basis is used with a Galerkin method to approximate the original system for, e. g., other inputs and/or parameters than those used to generate the snapshots for constructing the POD basis. A typical application is given by the PDE-constrained optimization, where the PDE system during the optimization is substituted by POD-Galerkin surrogates; see Section 2.6 for more details.

2.3.1 The POD-Galerkin procedure

Suppose that for given snapshots $y_j^h \in V^{h_j} \subset X$, $1 \leq j \leq n$, we have computed the symmetric matrix

$$K = ((\sqrt{\alpha_i}\sqrt{\alpha_j}\,\langle y_i^h, y_j^h\rangle_X))_{1 \leq i,j \leq n} \quad \text{with rank } K = r \leq n \tag{2.30}$$

associated to the operator $\mathcal{K}$ from (2.8) together with its eigensystem. Its $\ell \in \{1,\ldots,r\}$ largest eigenvalues are $\{\lambda_i\}_{i=1}^\ell$ with corresponding eigenvectors $\{\Phi_i\}_{i=1}^\ell \subset \mathbb{R}^n$. The POD basis $\{\Psi_i\}_{i=1}^\ell$ is then given by (2.9), i. e.,

$$\Psi_i = \frac{1}{\sqrt{\lambda_i}}\,\mathcal{Y}\Phi_i \quad \text{for } i = 1,\ldots,\ell.$$

This POD basis is utilized in order to compute a reduced-order model for (2.16) along the lines of Section 2.2.3.3, where the space V^h is replaced by the space $V^\ell = \mathrm{span}\{\Psi_1,\ldots,\Psi_\ell\} \subset V$. More precisely, we make the POD-Galerkin ansatz

$$y^\ell(t) = \sum_{i=1}^\ell \eta_i(t)\Psi_i = \sum_{i=1}^\ell \eta_i(t)\frac{1}{\sqrt{\lambda_i}}\,\mathcal{Y}\Phi_i \quad \text{for all } t \in [0,T], \tag{2.31}$$

as an approximation for $y(t)$, with the Fourier coefficients

$$\eta_i(t) = \langle y^\ell(t), \Psi_i\rangle_X = \left\langle y^\ell(t), \frac{1}{\sqrt{\lambda_i}}\,\mathcal{Y}\Phi_i\right\rangle_X \quad \text{for } 1 \leq i \leq \ell.$$

Inserting y^ℓ into (2.16) and choosing $V^\ell \subset V$ as the test space leads to the system

$$\left.\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\langle y^\ell(t), \Psi\rangle_H + a(y^\ell(t), \Psi) + \langle \mathcal{N}(y^\ell(t)), \Psi\rangle_{V',V} = \langle f(t), \Psi\rangle_{V',V}, \\
\langle y^\ell(0), \Psi\rangle_H = \langle y_\circ, \Psi\rangle_H
\end{aligned}\right\} \tag{2.32}$$

for all $\Psi \in V^\ell$ and for almost all $t \in (0,T]$. The system (2.32) is called POD reduced-order model (POD-ROM). Using the ansatz (2.31), we can write (2.32) as an ℓ-dimensional ordinary differential equation system for the POD mode coefficients $\eta(t) = (\eta_i(t))_{1 \leq i \leq \ell}$, $t \in (0,T]$ as follows:

$$\left.\begin{aligned}
\sum_{j=1}^\ell \dot{\eta}_j(t)\,\langle \Psi_i, \Psi_j\rangle_H + \sum_{j=1}^\ell \eta_j(t)\,a(\Psi_j, \Psi_i) = \langle f(t) - \mathcal{N}(y^\ell(t)), \Psi_i\rangle_{V',V}, \\
\sum_{j=1}^\ell \eta_j(0)\,\langle \Psi_i, \Psi_j\rangle_H = \langle y_\circ, \Psi_i\rangle_H
\end{aligned}\right\} \tag{2.33}$$

for $i = 1,\ldots,\ell$. Note that $\langle \Psi_i, \Psi_j\rangle_H = \delta_{ij}$ if we choose $X = H$ in the context of Section 2.2.3. In a next step we rewrite this system using the relation between Ψ_i and Φ_i

given in (2.9). This leads to

$$
\sum_{j=1}^{\ell} \dot{\eta}_j(t)\, \frac{\langle \mathcal{Y}\Phi_i, \mathcal{Y}\Phi_j \rangle_H}{\sqrt{\lambda_i \lambda_j}} + \sum_{j=1}^{\ell} \eta_j(t)\, \frac{a(\mathcal{Y}\Phi_j, \mathcal{Y}\Phi_i)}{\sqrt{\lambda_i \lambda_j}} = \frac{\langle f(t) - \mathcal{N}(y^{\ell}(t)), \mathcal{Y}\Phi_i \rangle_{V',V}}{\sqrt{\lambda_i}}
$$

$$
\text{for } t \in (0, T],
$$

$$
\sum_{j=1}^{\ell} \eta_j(0)\, \frac{\langle \mathcal{Y}\Phi_i, \mathcal{Y}\Phi_j \rangle_H}{\sqrt{\lambda_i \lambda_j}} = \frac{\langle y_\circ, \mathcal{Y}\Phi_i \rangle_H}{\sqrt{\lambda_i}}
$$

$$
(2.34)
$$

for $i = 1, \ldots, \ell$. In order to write (2.34) in a compact matrix-vector form, let us introduce the diagonal matrix

$$
\Lambda := \mathrm{diag}\left(\frac{1}{\sqrt{\lambda_1}}, \ldots, \frac{1}{\sqrt{\lambda_\ell}} \right) \in \mathbb{R}^{\ell \times \ell}.
$$

From the first ℓ eigenvectors $\{\Phi_i\}_{i=1}^{\ell}$ of K we build the matrix

$$
\Phi = [\Phi_1 \,|\, \ldots \,|\, \Phi_\ell] \in \mathbb{R}^{n \times \ell}.
$$

Then, the system (2.34) can be written as the system

$$
\begin{aligned}
(\Lambda \Phi^{\top} K \Phi \Lambda)\dot{\eta}(t) + (\Lambda A^{\ell} \Lambda)\eta(t) + \Lambda N^{\ell}(\eta(t)) &= \Lambda F^{\ell}(t) \quad \text{for } t \in (0, T], \\
\Lambda \Phi^{\top} K \Phi \Lambda \eta(0) &= \Lambda \eta_\circ.
\end{aligned}
$$

$$
(2.35)
$$

for the vector-valued mapping $\eta = (\eta_1, \ldots, \eta_\ell)^{\top} : [0, T] \to \mathbb{R}^{\ell}$, for the nonlinearity $N^{\ell} = (N_i^{\ell}(\cdot))_{1 \le i \le \ell} : \mathbb{R}^{\ell} \to \mathbb{R}^{\ell}$ with

$$
N_i^{\ell}(v) = \left\langle \mathcal{N}\left(\textstyle\sum_{j=1}^{\ell} v_j \Psi_j \right), \varphi_i \right\rangle_{V',V} = \left\langle \mathcal{N}\left(\textstyle\sum_{j=1}^{\ell} v_j \mathcal{Y}\Phi_j / \sqrt{\lambda_j} \right), \varphi_i \right\rangle_{V',V},
$$

and for the stiffness matrix $A^{\ell} = ((A_{ij}^{\ell})) \in \mathbb{R}^{\ell \times \ell}$ given as

$$
A_{ij}^{\ell} = a(\mathcal{Y}\Phi_j, \mathcal{Y}\Phi_i) \quad \text{for } 1 \le i, j \le \ell.
$$

Note that the right-hand side $F^{\ell}(t) = (F_i^{\ell}(t))_{1 \le i \le \ell}$ and the initial condition $\eta_\circ = (\eta_{\circ i})_{1 \le i \le \ell}$ are given by

$$
F_i^{\ell}(t) = \langle f(t), \mathcal{Y}\Phi_i \rangle_{V',V} = \langle \mathcal{Y}^* f(t), \Phi_i \rangle_{\mathbb{R}^n}, \quad t \in [0, T] \text{ a. e.,}
$$

and

$$
\eta_{\circ i} = \langle y_\circ, \mathcal{Y}\Phi_i \rangle_H = \langle \mathcal{Y}^* y_\circ, \Phi_i \rangle_{\mathbb{R}^n},
$$

for $i = 1, \ldots, \ell$, respectively. Their calculation can be done explicitly for any arbitrary finite element discretization. For a given function $w \in V$ (for example $w = f(t)$ or

$w = y_\circ$) with finite element discretization $w = \sum_{i=1}^{m_w} \mathsf{w}_i \chi_i$, nodal basis $\{\chi_i\}_{i=1}^{m_w} \subset V$, and appropriate mode coefficients $\{\mathsf{w}_i\}_{i=1}^{m_w}$ we can compute

$$(\mathcal{Y}^* w)_j = \langle w, y_j \rangle_X = \left\langle \sum_{i=1}^{m_w} \mathsf{w}_i \chi_i, \sum_{k=1}^{m_j} \mathsf{y}_k^j \varphi_k^j \right\rangle_X = \sum_{i=1}^{m_w} \sum_{k=1}^{m_j} \mathsf{w}_i \mathsf{y}_k^j \langle \chi_i, \varphi_k^j \rangle_X$$

for $j = 1, \ldots, n$, where $y_j^h = \sum_{k=1}^{m_j} \mathsf{y}_k^j \varphi_k^j \in V^{h_j}$ denotes the j-th snapshot. Again, for any $i = 1, \ldots, m_w$ and $k = 1, \ldots, m_j$, the computation of the inner product $\langle \chi_i, \varphi_k^j \rangle_X$ can be done explicitly.

Obviously, for linear evolution equations the POD-ROM (2.35) can be set up and solved using snapshots with arbitrary finite element discretizations. The computation of the nonlinear component $\mathsf{N}^\ell(\eta(t))$ needs particular attention. In Section 2.3.3 we discuss the options to treat the nonlinearity.

2.3.2 Time-discrete reduced-order model

In order to solve the reduced-order system (2.32) numerically, we apply the implicit Euler method for time discretization and use for simplicity the same temporal grid $\{t_j\}_{j=1}^n$ as for the snapshots. It is also possible to use a different time grid; cf. [58]. The time-discrete reduced-order model reads

$$\left. \begin{aligned} \frac{\langle y_j^\ell - y_{j-1}^\ell, \Psi \rangle_H}{\Delta t_j} + a(y_j^\ell, \Psi) + \langle \mathcal{N}(y_j^\ell), \Psi \rangle_{V',V} = \int_{t_{j-1}}^{t_j} \frac{\langle f(\tau), \Psi \rangle_{V',V}}{\Delta t_j} \, d\tau, \\ \langle y_1^\ell, \Psi \rangle_H = \langle y_\circ, \Psi \rangle_H \end{aligned} \right\} \qquad (2.36)$$

for all $\Psi \in V^\ell$ and $j = 2, \ldots, n$. Equivalently the following system holds for the coefficient vector $\eta(t) \in \mathbb{R}^\ell$ (cf. (2.35)):

$$\left. \begin{aligned} (\Lambda \Phi^\top K \Phi \Lambda)\left(\frac{\eta^j - \eta^{j-1}}{\Delta t_j} \right) + (\Lambda A^\ell \Lambda)\eta^j + \Lambda N^\ell(\eta^j) = \Lambda F_j^\ell, \; j = 2, \ldots, n, \\ \Lambda \Phi^\top K \Phi \Lambda \eta^1 = \Lambda \eta_\circ \end{aligned} \right\} \qquad (2.37)$$

with the inhomogeneity $F_j^\ell = (F_{ji}^\ell)_{1 \le i \le \ell}$, $j = 2, \ldots, n$, given as

$$F_{ji}^\ell = \int_{t_{j-1}}^{t_j} \frac{\langle f(\tau), \mathcal{Y}\Phi_i \rangle_{V',V}}{\Delta t_j} \, d\tau = \int_{t_{j-1}}^{t_j} \frac{\langle \mathcal{Y}^* f(\tau), \Phi_i \rangle_{\mathbb{R}^n}}{\Delta t_j} \, d\tau.$$

2.3.3 Discussion of the computation of the nonlinear term

Let us now consider the computation of the nonlinear term $\wedge \mathrm{N}^{\ell}(\eta^j) \in \mathbb{R}^{\ell}$ of the POD-ROM (2.35). We have

$$(\wedge \mathrm{N}^{\ell}(\eta^j))_k = \left\langle \mathcal{N}(y^{\ell}(t)), \Psi_k \right\rangle_{V',V} = \left\langle \mathcal{N}\left(\sum_{i=1}^{\ell} \eta_i(t) \Psi_i \right), \Psi_k \right\rangle_{V',V}$$

for $k = 1, \ldots, \ell$. It is well known that the evaluation of nonlinearities in the reduced-order modeling context is computationally expensive. To make this clear, let us assume we are given a uniform finite element discretization with m degrees of freedom. Then, in the fully discrete setting, the nonlinear term has the form

$$\Psi^{\top} W \mathrm{N}^h(\Psi \eta(t)) \in \mathbb{R}^{\ell}, \quad t \in [0, T] \text{ a. e.,}$$

where $\Psi = [\Psi_1 | \ldots | \Psi_{\ell}] \in \mathbb{R}^{m \times \ell}$ is the matrix in which the POD modes are stored columnwise and $W \in \mathbb{R}^{m \times m}$ is a weighting matrix related to the utilized inner product (cf. (2.3)). Hence, the treatment of the nonlinearity requires the expansion of $\Psi \eta(t) \in \mathbb{R}^m$ in the full space for $t \in [0, T]$ a. e. Then the nonlinearity can be evaluated and finally the result is projected back to the POD space. Obviously, this means that the reduced-order model is not fully independent of the high-order dimension m and efficient simulation cannot be guaranteed. Therefore, it is convenient to seek for hyper-reduction, i. e., for a treatment of the nonlinearity, where the model evaluation cost is related to the low dimension ℓ. Common choices are empirical interpolation methods like, e. g., the empirical interpolation method (EIM) [14], the discrete EIM (DEIM) [24], and the QR decomposition-based DEIM [31]. Another option is dynamic mode decomposition for nonlinear model order reduction; see, e. g., [7]. Furthermore, in [98] nonlinear model reduction is realized by replacing the nonlinear term by its interpolation in the finite element space. Alternative approaches for the treatment of the nonlinearity are missing point estimation [10] and best points interpolation [70].

Most of these methods need a common reference mesh for the computations. To overcome this restriction we propose different paths which allow for more general discrete settings like r-adaptivity, discussed in Run 2.3.

One option is to use EIM [14]. Alternatively, we can linearize and project the nonlinearity onto the POD space. For this approach, let us consider the linear reduced-order system for a fixed given state $\bar{y}$, which takes the form

$$\left. \begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \left\langle y^{\ell}(t), \Psi \right\rangle_H + a(y^{\ell}(t), \Psi) + \left\langle \mathcal{N}(\bar{y}(t)), \Psi \right\rangle_{V',V} &= \left\langle f(t), \Psi \right\rangle_{V',V}, \\ \left\langle y^{\ell}(0), \Psi \right\rangle_H &= \left\langle y_{\circ}, \Psi \right\rangle_H \end{aligned} \right\} \quad (2.38)$$

for all $\Psi \in V^{\ell}$ and for almost all $t \in (0, T]$. The linear evolution problem (2.38) can be set up and solved explicitly without spatial interpolation. In the numerical examples

in Section 2.6, we take the finite element solution as given state in each time step, i. e., $\bar{y}(t_j) = y_j$ for $j = 2, \ldots, n$.

Furthermore, the linearization of the reduced-order model (2.32) can be considered:

$$\left.\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} \langle y^\ell(t), \Psi \rangle_H + a(y^\ell(t), \Psi) + \langle \mathcal{N}'(\bar{y}(t))y^\ell(t), \Psi \rangle_{V',V} \\ = \langle f(t) - \mathcal{N}(\bar{y}(t)) + \mathcal{N}'(\bar{y}(t)))\bar{y}(t), \Psi \rangle_{V',V}, \\ \langle y^\ell(0), \Psi \rangle_H = \langle y_\circ, \Psi \rangle_H \end{aligned}\right\} \tag{2.39}$$

for all $\Psi \in V^\ell$ and for almost all $t \in (0, T]$, where $\mathcal{N}'$ denotes the Fréchet derivative of the nonlinear operator $\mathcal{N}$. This linearized problem is of interest, e. g., in the context of optimal control, where it occurs in each iteration level within sequential quadratic programming (SQP) methods; see [49], for example. Choosing the finite element solution as given state in each time instance and using (2.9) leads to

$$\langle \mathcal{N}(y_j), \Psi_i \rangle_{V',V} = \frac{1}{\sqrt{\lambda_i}} \sum_{k=1}^n \sqrt{\alpha_k}(\Phi_i)_k \langle \mathcal{N}(y_j), y_k \rangle_{V',V},$$

$$\langle \mathcal{N}'(y_j)y^\ell(t_j), \Psi_i \rangle_{V',V} = \left\langle \mathcal{N}'(y_j) \left(\sum_{k=1}^\ell \eta_k(t_j)\Psi_k \right), \Psi_i \right\rangle_{V',V}$$

$$= \sum_{k=1}^\ell \eta_k(t_j) \frac{1}{\sqrt{\lambda_k \lambda_i}} \sum_{\nu=1}^n \sum_{\mu=1}^n \sqrt{\alpha_\nu \alpha_\mu}(\Phi_k)_\nu (\Phi_i)_\mu \langle \mathcal{N}'(y_j)y_\nu, y_\mu \rangle_{V',V},$$

$$\langle \mathcal{N}'(y_j)y_j, \Psi_i \rangle_{V',V} = \frac{1}{\sqrt{\lambda_i}} \sum_{k=1}^n \sqrt{\alpha_k}(\Phi_i)_k \langle \mathcal{N}'(y_j)y_j, y_k \rangle_{V',V}$$

for $j = 2, \ldots, n$ and $i = 1, \ldots, \ell$. Finally, we approximate the nonlinearity $\Lambda \mathsf{N}^\ell(\eta^j) \in \mathbb{R}^\ell$ in (2.37) by

$$(\Lambda \mathsf{N}^\ell(\eta^j))_i \approx \langle \mathcal{N}(y_j) + \mathcal{N}'(y_j)(y^\ell(t_j) - y_j), \Psi_i \rangle_{V',V}$$

for $j = 2, \ldots, n$ and $i = 1, \ldots, \ell$, which can be written as

$$\Lambda \mathsf{N}^\ell(\eta^j) \approx \Lambda \Phi^\top \mathsf{N}^j + \Lambda \Phi^\top \mathsf{N}_y^j \Phi \Lambda \eta^j - \Lambda \Phi^\top \mathsf{N}_y^j,$$

where

$$\mathsf{N}^j = \begin{pmatrix} \langle \mathcal{N}(y_j), \sqrt{\alpha_1}y_1 \rangle_{V',V} \\ \vdots \\ \langle \mathcal{N}(y_j), \sqrt{\alpha_n}y_n \rangle_{V',V} \end{pmatrix} \in \mathbb{R}^n, \quad \mathsf{N}_y^j = \begin{pmatrix} \langle \mathcal{N}'(y_j)y_j, \sqrt{\alpha_1}y_1 \rangle_{V',V} \\ \vdots \\ \langle \mathcal{N}'(y_j)y_j, \sqrt{\alpha_n}y_n \rangle_{V',V} \end{pmatrix} \in \mathbb{R}^n,$$

and with $\tilde{y}_j = \sqrt{\alpha_j}y_j$, $j = 1, \ldots, n$,

$$\mathsf{N}_y^j = \begin{pmatrix} \langle \mathcal{N}'(y_j)\tilde{y}_1, \tilde{y}_1 \rangle_{V',V} & \cdots & \langle \mathcal{N}'(y_j)\tilde{y}_n, \tilde{y}_1 \rangle_{V',V} \\ \vdots & & \vdots \\ \langle \mathcal{N}'(y_j)\tilde{y}_1, \tilde{y}_1 \rangle_{V',V} & \cdots & \langle \mathcal{N}'(y_j)\tilde{y}_n, \tilde{y}_n \rangle_{V',V} \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

For weakly nonlinear systems this approximation may be sufficient, depending on the problem and its goal. A great advantage of linearizing the semi-linear partial differential equation is that only linear equations need to be solved, which leads to a further speedup; see Table 2.6. However, if a more precise approximation is desired or necessary, we can think of approximations including higher-order terms, like quadratic approximation, see, e. g., [25] and [84], or Taylor expansions, see, e. g., [73, 74] and [35]. Nevertheless, the efficiency of higher-order approximations is limited due to growing memory and computational costs.

2.3.4 Expressing the POD solution in the full spatial domain

Having determined the solution $\eta(t)$ to (2.35), we can set up the reduced solution $y^\ell(t)$ in a continuous framework:

$$y^\ell(t) = \sum_{i=1}^{\ell} \eta_i(t)\left(\frac{1}{\sqrt{\lambda_i}} \sum_{j=1}^{n} \sqrt{\alpha_j}(\Phi_i)_j y_j \right). \tag{2.40}$$

Now, let us turn to the fully discrete formulation of (2.40). For a time-discrete setting, we introduce for simplicity the same temporal grid $\{t_j\}_{j=1}^{n}$ as for the snapshots. The snapshots (2.28) admit the expansion

$$y_j = \sum_{i=1}^{m_j} y_j^i \varphi_i^j \quad \text{for } j = 1, \dots, n.$$

Let $\{Q_r^j\}_{r=1}^{l_j}$ denote an arbitrary set of grid points for the reduced system at time level t_j. The fully discrete POD solution can be computed by evaluation:

$$y^\ell(t_j, Q_r^j) = \sum_{i=1}^{\ell} \eta_i(t_j)\left(\frac{1}{\sqrt{\lambda_i}} \sum_{v=1}^{n} \sqrt{\alpha_v}(\Phi_i)_v \left(\sum_{k=1}^{m_v} y_k^v \varphi_k^v(Q_r^j) \right) \right) \tag{2.41}$$

for $r = 1, \dots, l_j$ and $j = 1, \dots, n$. This allows us to use any grid for expressing the POD solution in the full spatial domain. For example, we can use the same nodes at time level j for the POD simulation as we have used for the snapshots, i. e., for $j = 1, \dots, n$ we have $l_j = m_j$ and $Q_r^j = P_k^j$ for all $r, k = 1, \dots, m_j$. Another option can be to choose

$$\{Q_r^j\}_{r=1}^{l_j} = \bigcup_{j=1}^{n} \bigcup_{k=1}^{m_j} \{P_k^j\} \quad \text{for } j = 1, \dots, n,$$

i. e., the common finest grid. Obviously, a special and probably the easiest case concerning the implementation is to choose snapshots which are expressed with respect to the same finite element basis functions and utilize the common finest grid for the simulation of the reduced-order system, which is proposed by [94]. After expressing

the adaptively sampled snapshots with respect to a common finite element space, the subsequent steps coincide with the common approach of taking snapshots which are generated without adaptivity. Then, expression (2.41) simplifies to

$$y^\ell(t_j, P_r) = \sum_{i=1}^{\ell} \eta_i(t_j) \left(\frac{1}{\sqrt{\lambda_i}} \sum_{v=1}^{n} \sqrt{\alpha_v}(\Phi_i)_v y_v \right) \quad \text{for } j = 1, \dots, n, \tag{2.42}$$

where $\{P_r\}_{r=1}^{m}$ are the nodes of the common finite element space.

Run 2.4 ([39, Example 6.1]). Let us revisit Run 2.1 and consider its POD-Galerkin solutions. The POD solutions for $\ell = 10$ and $\ell = 50$ POD basis functions using spatial adaptive snapshots which are interpolated onto the finest mesh are shown in Figure 2.11. As expected, the more POD basis functions we use (until stagnation of the corresponding eigenvalues), the fewer oscillations appear in the POD solution and the better the approximation is. Table 2.2 compares the approximation quality in the rel-
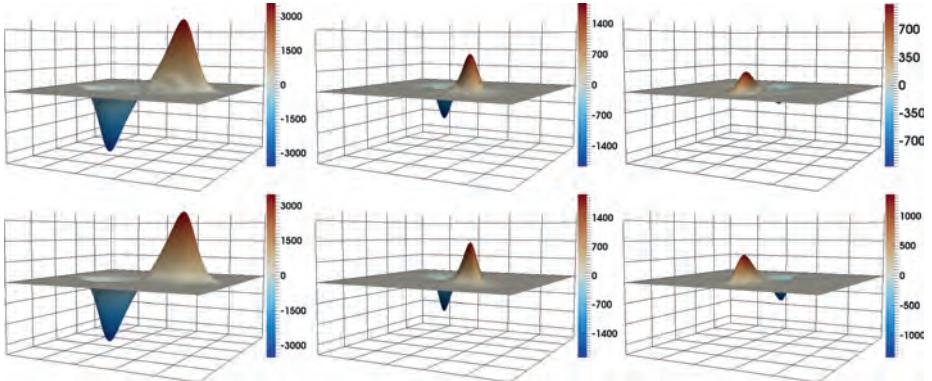
Figure 2.11: Run 2.4. Surface plot of the POD solution using $\ell = 10$ (top) and $\ell = 50$ (bottom) POD basis functions at $t = t_1$ (left), $t = T/2$ (middle), and $t = T$ (right).

ative $L^2(0, T; L^2(\Omega))$-norm of the POD solution using adaptively generated snapshots which are interpolated onto the finest mesh with snapshots of uniform spatial discretization depending on different POD basis lengths. Then, for $\ell = 20$ we obtain a relative $L^2(0, T; L^2(\Omega))$-error between the POD solution and the finite element solution of size $\varepsilon_{\text{FE}}^{\text{ad}} = 3.08 \cdot 10^{-2}$ and a relative $L^2(0, T; L^2(\Omega))$-error between the POD solution and the true solution of size $\varepsilon_{\text{true}}^{\text{ad}} = 2.17 \cdot 10^{-2}$.

We note that $\varepsilon_{\text{FE}}^{\text{uni}}$ decays down to 10^{-8} ($\ell = 100$) and then stagnates if using a uniform mesh. This behavior is clear, since the more POD basis elements we include (up to stagnation of the corresponding eigenvalues), the better an approximation is the POD solution for the finite element solution. On the other hand, both $\varepsilon_{\text{true}}^{\text{uni}}$ and $\varepsilon_{\text{true}}^{\text{ad}}$ start to stagnate after $\ell = 30$ in Table 2.2, columns 4 and 5. This is due to the fact that at this

Table 2.2: Run 2.4. Relative $L^2(0, T; L^2(\Omega))$-error between the POD solution and the finite element solution (columns 2 and 3) and the true solution (columns 4 and 5), respectively, using adaptive finite element snapshots which are interpolated onto the finest mesh and using a uniform mesh.

ℓ	ε_{FE}^{ad}	ε_{FE}^{uni}	ε_{true}^{ad}	ε_{true}^{uni}
1	$1.30 \cdot 10^{0}$	$1.30 \cdot 10^{0}$	$1.28 \cdot 10^{0}$	$1.30 \cdot 10^{0}$
3	$7.49 \cdot 10^{-1}$	$7.58 \cdot 10^{-1}$	$7.46 \cdot 10^{-1}$	$7.60 \cdot 10^{-1}$
5	$4.39 \cdot 10^{-1}$	$4.45 \cdot 10^{-1}$	$4.39 \cdot 10^{-1}$	$4.46 \cdot 10^{-1}$
10	$1.37 \cdot 10^{-1}$	$1.37 \cdot 10^{-1}$	$1.36 \cdot 10^{-1}$	$1.38 \cdot 10^{-1}$
20	$3.08 \cdot 10^{-2}$	$1.56 \cdot 10^{-2}$	$2.17 \cdot 10^{-2}$	$1.60 \cdot 10^{-2}$
30	$2.59 \cdot 10^{-2}$	$2.04 \cdot 10^{-3}$	$1.49 \cdot 10^{-2}$	$3.00 \cdot 10^{-3}$
50	$2.63 \cdot 10^{-2}$	$5.67 \cdot 10^{-5}$	$1.41 \cdot 10^{-2}$	$2.07 \cdot 10^{-3}$
100	$2.61 \cdot 10^{-2}$	$6.48 \cdot 10^{-8}$	$1.40 \cdot 10^{-2}$	$2.06 \cdot 10^{-3}$
150	$2.61 \cdot 10^{-2}$	$8.13 \cdot 10^{-7}$	$1.39 \cdot 10^{-2}$	$2.07 \cdot 10^{-3}$

Table 2.3: Run 2.4. Relative $L^2(0, T; H^1(\Omega))$-error between the POD solution and the finite element solution (columns 2 and 3) and the true solution (columns 4 and 5), respectively, using adaptive finite element snapshots which are interpolated onto the finest mesh and using a uniform mesh.

ℓ	ε_{FE}^{ad}	ε_{FE}^{uni}	ε_{true}^{ad}	ε_{true}^{uni}
1	$1.46 \cdot 10^{0}$	$1.46 \cdot 10^{0}$	$1.46 \cdot 10^{0}$	$1.47 \cdot 10^{0}$
3	$1.21 \cdot 10^{0}$	$1.22 \cdot 10^{0}$	$1.22 \cdot 10^{0}$	$1.22 \cdot 10^{0}$
5	$9.39 \cdot 10^{-1}$	$9.45 \cdot 10^{-1}$	$9.47 \cdot 10^{-1}$	$9.51 \cdot 10^{-1}$
10	$4.22 \cdot 10^{-1}$	$4.25 \cdot 10^{-1}$	$4.33 \cdot 10^{-1}$	$4.31 \cdot 10^{-1}$
20	$7.76 \cdot 10^{-2}$	$7.27 \cdot 10^{-2}$	$1.02 \cdot 10^{-1}$	$8.19 \cdot 10^{-2}$
30	$2.92 \cdot 10^{-2}$	$1.22 \cdot 10^{-2}$	$7.26 \cdot 10^{-2}$	$3.52 \cdot 10^{-2}$
50	$2.61 \cdot 10^{-2}$	$4.74 \cdot 10^{-4}$	$7.05 \cdot 10^{-2}$	$3.27 \cdot 10^{-2}$
100	$2.79 \cdot 10^{-2}$	$4.78 \cdot 10^{-7}$	$6.94 \cdot 10^{-2}$	$3.27 \cdot 10^{-2}$
150	$2.93 \cdot 10^{-2}$	$2.84 \cdot 10^{-7}$	$6.87 \cdot 10^{-2}$	$3.27 \cdot 10^{-2}$

point the spatial (and temporal) discretization error dominates the modal error. This is in accordance with the decay of the eigenvalues shown in Figure 2.3 and is accounted for, e. g., in the error estimation presented in [39, Theorem 5.1]. Similar observations hold true for the relative $L^2(0, T; H^1(\Omega))$-error listed in Table 2.3 with the difference that the $L^2(0, T; H^1(\Omega))$-error is larger than the respective $L^2(0, T; L^2(\Omega))$-error.

The computational times for the full and the low-order simulation using uniform finite element discretizations and adaptive finite element snapshots, which are interpolated onto the finest mesh, respectively, are listed in Table 2.4.

Once the POD basis is computed in the offline phase, the POD simulation corresponding to adaptive snapshots is 13,485 times faster than the finite element simulation using adaptive finite element meshes. This speedup factor is important when one considers, e. g., optimal control problems with time-dependent PDEs, where the POD-ROM can be used as surrogate model in repeated solution of the underlying PDE

Table 2.4: Run 2.4. CPU times for FE and POD simulation using uniform finite element meshes and adaptive finite element snapshots which are interpolated onto the finest mesh, respectively, and using $\ell = 50$ POD modes.

	Adaptive FE mesh	Uniform FE mesh	Speedup factor
FE simulation	944 sec	8,808 sec	9.3
POD offline computations	264 sec	1,300 sec	4.9
POD simulation	0.07 sec	0.07 sec	–
Speedup factor	13,485	125,828	–

model. In the POD offline phase, the most expensive task is to express the snapshots with respect to the common finite element space, which takes 226 seconds. Since K (2.30) is symmetric, it suffices to calculate the entries on and above the diagonal, which are $\sum_{k=1}^{n} k = (n^2 + n)/2$ entries. Thus, the computation of each entry in the correlation matrix K using a common finite element space takes around 0.00018 seconds. We note that in the approach explained in Sections 2.2.4 and 2.3, the computation of the matrix K is expensive. For each entry the calculation time is around 0.03 seconds, which leads to a computation time of around 36,997 seconds for matrix K. The same effort is needed to build $A^{\ell} = a(\mathcal{Y}\Phi_j, \mathcal{Y}\Phi_i)$. In this case, the offline phase takes therefore around 88,271 seconds. For this reason, the approach to interpolate the adaptively generated snapshots onto the finest mesh is computationally more favorable. But since the computation of K can be parallelized, the offline computation time can be reduced provided that the appropriate hardware is available.

Run 2.5 (Cahn–Hilliard equations). Now let us revisit Run 2.2, where in the following we run the numerical simulations for different combinations of numbers for ℓ_c and ℓ_w of Table 2.1. The approximation quality of the POD solution using adaptive meshes is compared to the use of a uniform mesh in Table 2.5. As expected, Table 2.5 shows that the error between the POD surrogate solution and the high-fidelity solution gets smaller for an increasing number of utilized POD basis functions. Moreover, a larger number of POD modes is needed for the chemical potential w than for the phase field c in order to get an error in the same order, which is in accordance with the fact that the decay of the eigenvalues for w is slower than for c, as seen in Figure 2.8.

We now discuss the treatment of the nonlinearity and also investigate the influence of nonsmoothness of the model equations to the POD procedure. Using the convex-concave splitting for W, we obtain for the Moreau–Yosida relaxed double obstacle free energy the concave part $W_{-}^{\text{rel}}(c) = \frac{1}{2}(1 - c^2)$ and the convex part $W_{+}^{\text{rel}}(c) = \frac{s}{2}(\max(c - 1, 0)^2 + \min(c + 1, 0)^2)$. This means that the first derivative of the concave part is linear with respect to the phase field variable c. The challenging part is the convex term with nonsmooth first derivative. For a comparison, we consider the smooth polynomial free energy with concave part $W_{-}^{p}(c) = \frac{1}{4}(1 - 2c^2)$ and convex part $W_{+}^{p}(c) = \frac{1}{4}c^4$.

Table 2.5: Run 2.5. Relative $L^2(0, T; L^2(\Omega))$-error between the POD solution and the finite element solution using adaptive meshes (columns 3 and 4) and using a uniform mesh (columns 5 and 6), respectively.

| ℓ^c	ℓ^w	$c : \varepsilon_{\text{FE}}^{\text{ad}}$	$w : \varepsilon_{\text{FE}}^{\text{ad}}$	$c : \varepsilon_{\text{FE}}^{\text{uni}}$	$w : \varepsilon_{\text{FE}}^{\text{uni}}$
3	4	$8.44 \cdot 10^{-3}$	$3.00 \cdot 10^{0}$	$8.44 \cdot 10^{-3}$	$3.75 \cdot 10^{0}$
10	13	$3.30 \cdot 10^{-3}$	$3.77 \cdot 10^{-1}$	$3.30 \cdot 10^{-3}$	$4.32 \cdot 10^{-1}$
19	26	$1.57 \cdot 10^{-3}$	$2.12 \cdot 10^{-1}$	$1.57 \cdot 10^{-3}$	$2.39 \cdot 10^{-1}$
29	26	$7.34 \cdot 10^{-4}$	$1.09 \cdot 10^{-1}$	$7.32 \cdot 10^{-4}$	$1.16 \cdot 10^{-1}$
37	26	$3.57 \cdot 10^{-4}$	$4.82 \cdot 10^{-2}$	$3.55 \cdot 10^{-4}$	$5.04 \cdot 10^{-2}$
50	50	$1.88 \cdot 10^{-4}$	$2.17 \cdot 10^{-2}$	$1.86 \cdot 10^{-4}$	$2.33 \cdot 10^{-2}$
65	26	$9.74 \cdot 10^{-5}$	$1.11 \cdot 10^{-2}$	$9.56 \cdot 10^{-5}$	$1.15 \cdot 10^{-2}$
100	100	$3.37 \cdot 10^{-5}$	$3.56 \cdot 10^{-3}$	$3.22 \cdot 10^{-5}$	$3.42 \cdot 10^{-3}$
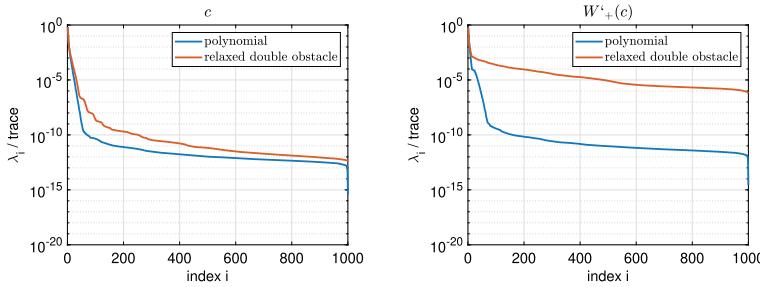
Figure 2.12: Run 2.5. Comparison of the normalized eigenvalues for c (left) and the first derivative of the convex part W'_+ of the free energy (right) using polynomial and relaxed double obstacle energy, respectively.

Figure 2.12 shows the decay of the normalized eigenspectrum for the phase field c (left) and the first derivative of the convex part $W'_+(c)$ (right) for the polynomial and the relaxed double obstacle free energy. Obviously, in the nonsmooth case more POD modes are needed for a good approximation than in the smooth case. This behavior is similar to the decay of the Fourier coefficients in the context of trigonometric approximation, where the decay of the Fourier coefficients depends on the smoothness of the approximated object.

Table 2.6 summarizes computational times for different finite element runs as well as reduced-order simulations using the polynomial and the relaxed double obstacle free energy, respectively. In addition, the approximation quality is compared. The computational times are rounded averages from various test runs. It turns out that the finite element simulation (row 1) using the smooth potential is around two times faster than using the nonsmooth potential. This is due to the fact that in the smooth case, two to three Newton steps are needed for convergence in each time step, whereas in the nonsmooth case six to eight iterations are needed in the semi-smooth Newton method.

Table 2.6: Run 2.5. Computational times, speedup factors, and approximation quality for different POD basis lengths and using different free energy potentials.

FE	W^p 1,644 s		W_s^{rel} 3,129 s	
	$\ell_c = 3$ $\ell_w = 4$	$\ell_c = 19$ $\ell_w = 26$	$\ell_c = 3$ $\ell_w = 4$	$\ell_c = 19$ $\ell_w = 26$
POD offline	355 s	355 s	350 s	349 s
DEIM offline	8 s	8 s	9 s	10 s
ROM	183 s	191 s	2,616 s	3,388 s
ROM-DEIM	0.05 s	0.1 s	0.04 s	no conv.
ROM-proj	0.008 s	0.03 s	0.01 s	0.03 s
speedup FE-ROM	8.9	8.6	1.1	none
speedup FE-ROM-DEIM	32,880	16,440	78,225	–
speedup FE-ROM-proj	205,500	54,800	312,900	104,300
rel $L^2(Q)$ error ROM	$5.46 \cdot 10^{-3}$	$3.23 \cdot 10^{-4}$	$8.44 \cdot 10^{-3}$	$1.57 \cdot 10^{-3}$
rel $L^2(Q)$ error ROM-DEIM	$1.46 \cdot 10^{-2}$	$3.83 \cdot 10^{-4}$	$8.84 \cdot 10^{-3}$	–
rel $L^2(Q)$ error ROM-proj	$4.70 \cdot 10^{-2}$	$4.18 \cdot 10^{-2}$	$8.72 \cdot 10^{-3}$	$9.80 \cdot 10^{-3}$

Using the smooth polynomial free energy, the reduced-order simulation is eight to nine times faster than the finite element simulation, whereas using the relaxed double obstacle free energy it only delivers a very small speedup. The inclusion of DEIM (we use $\ell_{deim} = \ell_c$) in the reduced-order model leads to immense speedup factors for both free energy functions (row 8). This is due to the fact that the evaluation of the nonlinearity in the reduced-order model is still dependent on the full spatial dimension and hyperreduction methods are necessary for useful speedup factors. Note that the speedup factors are of particular interest in the context of optimal control problems. At the same time, the relative $L^2(0, T; L^2(\Omega))$-error between the finite element solution and the ROM-DEIM solution is close to the quality of the reduced-order model solution (rows 10 and 11).

However, in the case of the nonsmooth free energy function using $\ell_c = 19$ POD modes for the phase field and $\ell_w = 26$ POD modes for the chemical potential, the inclusion of DEIM has the effect that the semi-smooth Newton method does not converge. For this reason, we treat the nonlinearity by applying the technique explained in Section 2.3.1, i. e., we project the finite element snapshots for $W'_+(c)$ (which are interpolated onto the finest mesh) onto the POD space. Since this leads to linear systems, the computational times are very small (row 6). The error between the finite element solution and the reduced-order solution using projection of the nonlinearity is of the magnitude $10^{-2}/10^{-3}$. Depending on the motivation, this approximation quality might be sufficient. Nevertheless, we note that for large numbers of POD modes, using the projection of the nonlinearity onto the POD space leads to a large increase of the error.

To summarize, a POD-ROM construction approach is proposed which can be set up and solved for snapshots originating from arbitrary finite element (and also other) spaces. The method is applicable for h-, p-, and r-adaptive finite elements. It is motivated from an infinite-dimensional perspective. Using the method of snapshots we are able to set up the correlation matrix K from (2.30) by evaluating the inner products of snapshots which live in different finite element spaces. For nonnested meshes, this requires the detection of cell collision and integration over cut finite elements. A numerical strategy how to implement this practically is elaborated and numerically tested. Using the eigenvalues and eigenvectors of this correlation matrix, we are able to set up and solve a POD surrogate model that does not need the expression of the snapshots with respect to the basis of a common finite element space or the interpolation onto a common reference mesh. Moreover, an error bound for the error between the true solution and the solution to the POD-ROM using spatially adapted snapshots is available in [39, Theorem 5.1]. The numerical tests show that the POD projection error decreases if the number of utilized POD basis functions is increased. However, the error between the POD solution and the true solution stagnates when the spatial discretization error dominates. Moreover, the numerics show that using the correlation matrix calculated explicitly without interpolation in order to build a POD-ROM gives the same results as the approach where the snapshots are interpolated onto the finest mesh. From a computational point of view, sufficient hardware should be available in order to compute the correlation matrix in parallel and make the offline computational time competitive. For semi-linear evolution problems, the nonlinearity is treated by linearization. This is of interest in view of optimal control problems, in which a linearized state equation has to be solved in each SQP iteration level. An appropriate treatment of the nonlinearity in our applications gains significant speedup of the reduced-order model with respect to computational time when compared to the full simulations. This makes POD-ROM with adaptive finite elements an ideal approach for the construction of surrogate models in, e. g., optimal control with nonlinear PDE systems, as they arise, e. g., in the context of multiphase flow control problems.

2.4 Certification with a priori and a posteriori error estimates

As we have seen in Section 2.3, POD provides a method for deriving low-order models of dynamical systems. It can be thought of as a Galerkin approximation in the spatial variable, built from functions corresponding to the solution of the physical system at prespecified time instances. After carrying out SVD, the leading ℓ generalized eigenfunctions are chosen as the POD basis $\{\Psi_j\}_{j=1}^{\ell}$ of rank ℓ. As soon as one uses POD, questions concerning the quality of the approximation properties, convergence, and rate of convergence become relevant. Let us refer, e. g., to the literature

[22, 42, 56, 58, 57, 85, 88, 89, 80] for a priori error analyses of POD-Galerkin approximations. It turns out that the error depends on the decay of the sum $\sum_{i>\ell} \lambda_i$, the error Δt^β (with an appropriate $\beta \geq 1$) due to the used time integration method, the used Galerkin spaces $\{V^{h_j}\}_{j=1}^n$, and the choice $X = H$ or $X = V$. In particular, best approximation properties hold provided the time differences $\dot{y}^h(t_j)$ (or the finite difference discretizations) are included in the snapshot ensembles; cf. [56, 58, 89].

Let us recall numerical test examples from [42, Section 1.5]. The programs are written in MATLAB using the PARTIAL DIFFERENTIAL EQUATION TOOLBOX for the computation of the piecewise linear finite element discretization. For the temporal integration the implicit Euler method is applied based on the equidistant time grid $t_j = (j-1)\Delta t$, $j = 1, \ldots, n$, and $\Delta t = T/(n-1)$.

Run 2.6 (POD for the heat equation; cf. [42, Run 1]). We choose the final time $T = 3$, the spatial domain $\Omega = (0,2) \subset \mathbb{R}$, the Hilbert spaces $H = L^2(\Omega)$, $V = H_0^1(\Omega)$, the source term $f(t, \boldsymbol{x}) = t^3 - \boldsymbol{x}^2$ for $(t, \boldsymbol{x}) \in Q = (0,T) \times \Omega$, and the discontinuous initial value $y_\circ(\boldsymbol{x}) = \chi_{(0.5,1.0)} - \chi_{(1,1.5)}$ for $\boldsymbol{x} \in \Omega$, where, e. g., $\chi_{(0.5,1)}$ denotes the characteristic function on the subdomain $(0.5,1) \subset \Omega$, $\chi_{(0.5,1)}(\boldsymbol{x}) = 1$ for $\boldsymbol{x} \in (0.5,1)$ and $\chi_{(0.5,1)}(\boldsymbol{x}) = 0$ otherwise. We consider a discretization of the linear heat equation (compare (2.17) with $c \equiv 0$)

$$
\begin{aligned}
y_t(t, \boldsymbol{x}) - \Delta y(t, \boldsymbol{x}) &= f(t, \boldsymbol{x}) && \text{for } (t, \boldsymbol{x}) \in Q, \\
y(t, \boldsymbol{x}) &= 0 && \text{for } (t, \boldsymbol{x}) \in \Sigma = (0,T) \times \partial\Omega, \\
y(0, \boldsymbol{x}) &= y_\circ(\boldsymbol{x}) && \text{for } \boldsymbol{x} \in \Omega.
\end{aligned}
\tag{2.43}
$$

To obtain an accurate approximation of the exact solution we choose $n = 4{,}000$ so that $\Delta t \approx 7.5 \cdot 10^{-4}$ holds. For the finite element discretization we choose $m = 500$ spatial grid points and the equidistant mesh size $h = 2/(m+1) \approx 4 \cdot 10^{-3}$. Thus, the finite element error – measured in the H-norm – is of the order 10^{-4}. In the left graphic of Figure 2.13, the finite element solution y^h to the state equation (2.43) is visualized. To compute a POD basis $\{\Psi_i\}_{i=1}^\ell$ of rank ℓ we utilize the multiple discrete snapshots $y_j^1 = y^h(t_j)$ for $1 \leq j \leq n_t$ as well $y_1^2 = 0$ and $y_j^2 = (y^h(t_j) - y^h(t_{j-1}))/\Delta t$, $j = 2, \ldots, n_t$, i. e., we include the temporal difference quotients in the snapshot ensemble and $K = 2$, $n_1 = n_2 = n_t$. We choose $X = H$ and utilize the (stable) SVD to determine the POD basis of rank ℓ; compare Section 2.2.2. We address this issue in a more detail in Run 2.9. Since the snapshots are finite element functions, the POD basis elements are also finite element functions. In the right plot of Figure 2.13, the projection and reduced-order error given by

$$
\text{PROJ Error}(\ell) = \left(\sum_{j=1}^{n_t} \alpha_j \left\| y^h(t_j) - \sum_{i=1}^{\ell} \langle y^h(t_j), \psi_i \rangle_H \psi_i \right\|_H^2 \right)^{1/2},
$$

$$
\text{ROM Error}(\ell) = \left(\sum_{j=1}^{n_t} \alpha_j \left\| y^h(t_j) - y^\ell(t_j) \right\|_H^2 \right)^{1/2}
$$
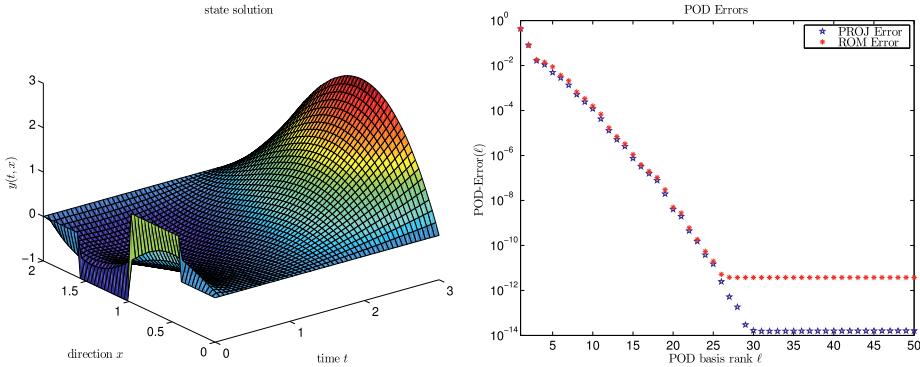
Figure 2.13: Run 2.6 (cf. [42, Figure 1.1]). The FE solution y^h (left) and the residuals corresponding to the POD basis rank ℓ (right).

are plotted for different POD basis ranks ℓ. The chosen trapezoidal weights α_j have been introduced in (2.23). We observe that both errors decay rapidly and coincide until the accuracy 10^{-12}, which is already significantly smaller than the finite element discretization error. These numerical results reflect the a priori error estimates presented in [42, Theorem 1.29].

Run 2.7 (POD for a convection dominated heat equation; cf. [42, Run 2]). Now we consider a more challenging example. We study a convection-reaction-diffusion equation with a source term which is close to being singular: Let T, Ω, $y_\circ$, H, and V be given as in Run 2.6. The parabolic problem reads as follows:

$$y_t(t,\boldsymbol{x}) - cy_{\boldsymbol{xx}}(t,\boldsymbol{x}) + \beta y_x(t,\boldsymbol{x}) + ay(t,\boldsymbol{x}) = f(t,\boldsymbol{x}) \qquad \text{for } (t,\boldsymbol{x}) \in Q,$$
$$y(t,\boldsymbol{x}) = 0 \qquad \text{for } (t,\boldsymbol{x}) \in \Sigma,$$
$$y(0,\boldsymbol{x}) = y_\circ(\boldsymbol{x}) \qquad \text{for } \boldsymbol{x} \in \Omega.$$

We choose the diffusivity $c = 0.025$, the velocity $\beta = 1.0$ that determines the speed in which the initial profile $y_\circ$ is shifted to the boundary, and the reaction rate $a = -0.001$. Finally, $f(t,\boldsymbol{x}) = \mathbb{P}(\frac{1}{1-t})\cos(\pi\boldsymbol{x})$ for $(t,\boldsymbol{x}) \in Q$, where $(\mathbb{P}z)(t) = \min(+l, \max(-l, z(t)))$ restricts the image of z on a bounded interval. In this situation, the state solution y develops a jump at $t = 1$ for $l \to \infty$; see the left plot of Figure 2.14. The right plot of Figure 2.14 demonstrates that in this case, the decay of the reconstruction residuals and the decay of the errors are much slower than in the right plot of Figure 2.13. The manifold dynamics of the state solution require an inconveniently large number of POD basis elements. Since the supports of these ansatz functions in general cover the whole domain Ω, the corresponding system matrices of the reduced model are not sparse. This is different for the matrices arising in the finite element Galerkin framework. Model order reduction is not effective for this example if a good accuracy of the solution function y^ℓ is required. Strategies to improve the accuracy and robustness of the POD-ROM in those situations are discussed in, e. g., [18, 99]
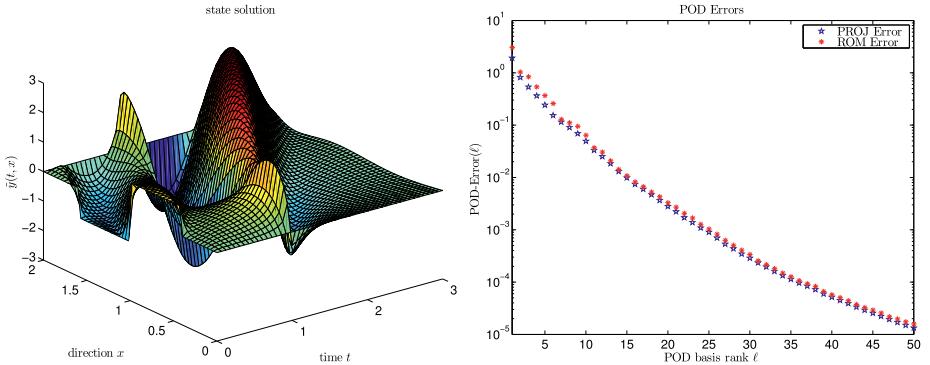
Figure 2.14: Run 2.7 (cf. [42, Figure 1.2]). The FE solution y^h (left) and the residuals corresponding to the POD basis rank ℓ (right).

Run 2.8 (True and exact approximation error; cf. [42, Run 3]). We consider the setting introduced in Run 2.6 again. The exact solution to (2.43) does not possess a representation by elementary functions. Hence, the presented reconstruction and reduction errors actually are the residuals with respect to a high-order finite element solution y^h. To compute an approximation y of the exact solution y_{ex} we apply a Crank–Nicolson method (with Rannacher smoothing [77]) ensuring $\|y - y_{ex}\|_{L^2(0,T;H)} = \mathcal{O}(\Delta t^2 + h^2) \approx 10^{-5}$. In the context of model reduction, such a state is sometimes called the "true" solution. To compute the finite element state y^h we apply the Euler method. In the left plot of Figure 2.15 we compare the true solution y_{ex} with the associated POD approximation for different values $n_t \in \{64, 128, 256, \ldots, 8192\}$ of the time integration and for the spatial mesh size $h = 4 \cdot 10^{-3}$. For the norm we apply a discrete $L^2(0, T; H)$-norm as in Run 2.6. Let us mention that we compute for every n_t a corresponding finite element solution y^h. We observe that the residuals ignore the errors arising by the application
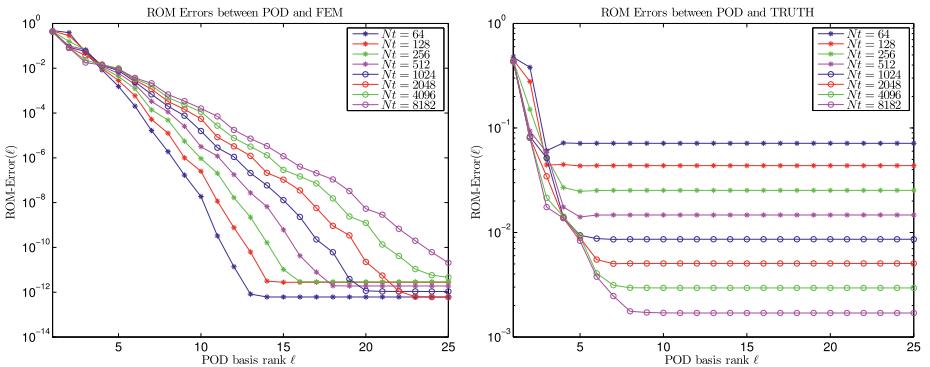

Figure 2.15: Run 2.8 (cf. [42, Figure 1.3]). The reduced-order model errors with respect to the true solution (left) and the exact one (right).

of time and space discretization schemes for the full-order model. The errors decay below the discretization error 10^{-5}. If these discretization errors are taken into account, the residuals stagnate at the level of the full-order model accuracy instead of decaying to zero; cf. the right plot of Figure 2.15. Due to the implicit Euler method we have $\|y^h - y_{\text{ex}}\|_{L^2(0,T;H)} = \mathcal{O}(\Delta t + h^2)$ with the mesh size $h = 4 \cdot 10^{-3}$. In particular, from $n_t \in \{64, 128, 256, \ldots, 8192\}$ it follows that $\Delta t > 3 \cdot 10^{-4} > h^2 = 1.6 \cdot 10^{-5}$. Therefore, the spatial error is dominated by the time error for all values of n_t. We can observe that the exact residuals do not decay below a limit of the order Δt. One can observe that for fixed POD basis rank ℓ, the residuals with respect to the true solution increase if the high-order accuracy is improved by enlarging n_t, since the reduced-order model has to approximate a more complex system in this case, where the residuals with respect to the exact solution decrease due to the lower limit of stagnation $\Delta t = 3/(n_t - 1)$.

Run 2.9 (Different strategies for a POD basis computation; cf. [42, Run 4]). As explained in Section 2.2.2, let $Y \in \mathbb{R}^{m \times n}$ denote the matrix of snapshots with rank r, let $W \in \mathbb{R}^{m \times m}$ be the (sparse) spatial weighting matrix consisting of the elements $\langle \varphi_j, \varphi_i \rangle_X$ (introduced in Section 2.2.3.3), and let $D \in \mathbb{R}^{n \times n}$ be the diagonal matrix containing the nonnegative weighting parameters α_j^k. As explained in Section 2.2.2, the POD basis $\{\Psi_i\}_{i=1}^{\ell}$ of rank $\ell \le$ r can be determined by providing an eigenvalue decomposition of the matrix $\bar{Y}\bar{Y}^\top = W^{1/2}YDY^\top W^{1/2} \in \mathbb{R}^{m \times m}$, one of $\bar{Y}^\top \bar{Y} = D^{1/2}Y^\top WYD^{1/2} \in \mathbb{R}^{n \times n}$, or an SVD of $\bar{Y} = W^{1/2}YD^{1/2} \in \mathbb{R}^{m \times n}$. Since $n \gg m$ in Runs 2.6–2.8, the first variant is the cheapest one from a computational point of view. In case of multiple space dimensions or if a second-order time integration scheme such as some Crank–Nicolson technique is applied, the situation is converse. On the other hand, an SVD is more accurate and stable than an eigenvalue decomposition if the POD elements corresponding to eigenvalues/singular values which are close to zero are taken into account: Since $\lambda_i = \sigma_i^2$ holds for all eigenvalues λ_i and singular values σ_i, the singular values are able to decay to machine precision, where the eigenvalues stagnate significantly above. This is illustrated in the left graphic of Figure 2.16. Indeed, for $\ell > 20$ the EIG-ROM system matrices become singular due to the numerical errors in the eigenfunctions and the reduced-order system is ill-posed in this case, while the SVD-ROM model remains stable. In the right plot of Figure 2.16 POD elements are constructed with respect to different scalar products and the resulting reduced-order model errors are compared: $\| \cdot \|_H$-residuals for $X = H$ (denoted by POD(H)), $\| \cdot \|_V$-residuals for $X = V$ (denoted by POD(V)), and $\| \cdot \|_V$-residuals for $X = H$ (denoted by POD(H,V)), which also works quite well, the consideration of time derivatives in the snapshot sample (denoted by POD(H,dt)), which allows to apply the a priori error estimate given in [42, Theorem 1.29-2)], and the corresponding sums of singular values (denoted by SV(H,dt)) corresponding to the unused eigenfunctions in the latter case which indeed nearly coincide with the reduced-order model errors.
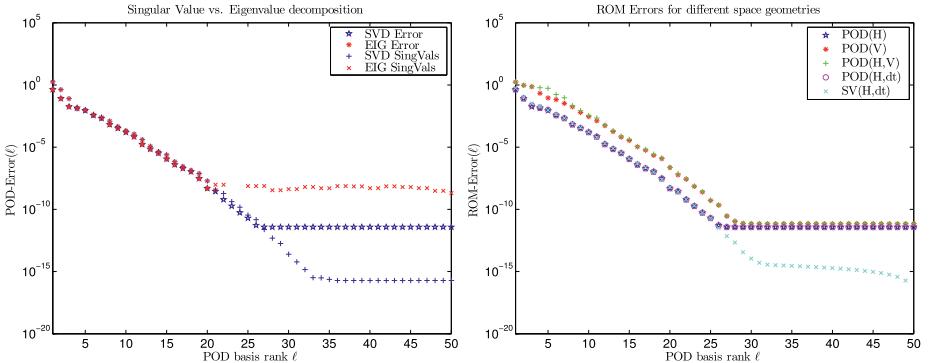
Figure 2.16: Run 2.9 (cf. [42, Figure 1.4]). Singular values σ_i using the SVD (SVD Vals) or the eigenvalue decomposition (EIG Vals) and the associated reduced-order model errors (SVD Error and EIG Error, respectively) (left); reduced-order model errors for different choices for X, the error norm, and the snapshot ensembles (right).

Note that in many applications, the quality of the reduced-order model does not vary significantly if the weights matrix W refers to the space $X = H$ or $X = V$ and if time derivatives of the used snapshots are taken into account or not. Especially, the reduced-order model residual decays with the same order as the sum over the remaining singular values, independent of the chosen geometrical framework.

2.5 Optimal snapshot location for computing POD basis functions

The construction of reduced-order models for nonlinear dynamical systems using POD is based on the information carried of the so-called snapshots. These provide the spatial distribution of the nonlinear system at discrete time instances. Thus, we are interested in optimizing the choice of these time instances in such a manner that the error between the POD solution and the trajectory of the dynamical system is minimized. This approach was suggested in [59] and was extended in [64] to parameterized elliptic problems. Let us briefly mention some related issues of interest. In [26, 32] the situation of missing snapshot data is investigated and gappy POD is introduced for their reconstruction. An important alternative to POD model reduction is given by reduced basis approximations; we refer to [72] and references given there. In [37] a reduced model is constructed for a parameter-dependent family of large-scale problems by an iterative procedure that adds new basis variables on the basis of a greedy algorithm. In the PhD thesis [20] a model reduction is sought of a class for a family of models corresponding to different operating stages.

Suppose that we are given the n_t snapshots $\{y(t_j)\}_{j=1}^{n_t} \subset V \subset X$. The goal is to determine additional k snapshots at time instances $\tau = (\tau_1, \ldots, \tau_k)$ with $0 \le \tau_j \le T$, $j = 1, \ldots, k$. In [59] we propose to determine $\tau = (\tau_1, \ldots, \tau_k)$ by solving the optimization problem

$$\min_{0 \le \tau_1, \ldots, \tau_k \le T} \int_0^T \|y(t) - y^\ell(t)\|_V^2 \, dt, \tag{2.44}$$

where y and y^ℓ are the solutions to (2.16) and its POD-Galerkin approximation, respectively. Clearly, the definition of the operator $\mathcal{R}$ given in (2.6) has to be modified as follows:

$$\mathcal{R}^\tau \Psi = \sum_{j=1}^{n_t} \alpha_j^\tau \langle y(t_j), \Psi \rangle_X y(t_j) + \sum_{j=1}^k \alpha_{n_t+j}^\tau \langle y(\tau_j), \Psi \rangle_X y(\tau_j)$$

with appropriately modified (trapezoidal) weights α_j^τ, $j = 1, \ldots, k + n_t$. Consequently, (2.44) becomes an optimization problem subject to the equality constraints

$$\mathcal{R}^\tau \Psi_i = \lambda_i \Psi_i, \quad i = 1, \ldots, \ell.$$

Note that no precautions are made in (2.44) to avoid multiple appearance of a snapshot. In fact, this would simply imply that a specific snapshot location should be given a higher weight than others. While the presented approach shows how to choose optimal snapshots in evolution equations, a similar strategy is applicable in the context of parameter-dependent systems.

It turns out in our numerical tests carried out in [59] that the proposed criterion is sensitive with respect to the choice of the time instances. Moreover, the tests demonstrate the feasibility of the method in determining optimal snapshot locations for concrete diffusion equations.

Run 2.10 (cf. [59, Run 1]). For $T = 1$ let $Q = (0, T) \times \Omega$ and $\Omega = (0, 1) \times (0, 1) \subset \mathbb{R}^2$. For the finite element triangulation we choose a uniform grid with mesh size $h = 1/40$, i. e., we have 900 degrees of freedom for the spatial discretization. Then, we consider

$$y_t(t, \boldsymbol{x}) - c\Delta y(t, \boldsymbol{x}) + \beta \cdot \nabla y(t, \boldsymbol{x}) + y(t, \boldsymbol{x}) = f(\boldsymbol{x}) \qquad \text{for } (t, \boldsymbol{x}) \in Q,$$

$$c\frac{\partial y}{\partial \boldsymbol{x}}(t, \boldsymbol{x}) + q(\boldsymbol{x})y(t, \boldsymbol{x}) = g(\boldsymbol{x}) \qquad \text{for } (t, \boldsymbol{x}) \in \Sigma,$$

$$y(0, \boldsymbol{x}) = y_\circ(\underline{x}) \qquad \text{for } \boldsymbol{x} \in \Omega,$$

where $c = 0.1$, $\beta = (0.1, -10)^\top \in \mathbb{R}^2$,

$$f(\boldsymbol{x}) = \begin{cases} 4 & \text{for all } \boldsymbol{x} = (x_1, x_2) \text{ with } (x_1 - 0.25)^2 + (x_2 - 0.65)^2 \le 0.05, \\ 0 & \text{otherwise,} \end{cases}$$
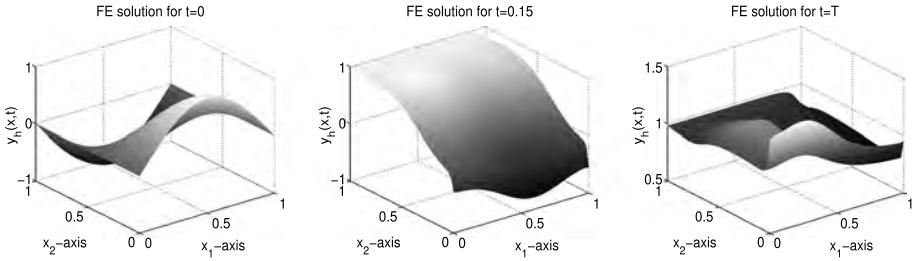
Figure 2.17: Run 2.10 (cf. [59, Figures 3 and 4]). Initial condition $y_\circ$ (left plot) and FE solution y^h for $t = 0.3$ (middle) and $t = T$ (right plot).

and $y_\circ(\boldsymbol{x}) = \sin(\pi x_1) \cos(\pi x_2)$ for $\boldsymbol{x} = (x_1, x_2) \in \Omega$ (Figure 2.17, left plot). Furthermore, we have

$$q(\boldsymbol{x}) = \begin{cases} 1 & \text{for } \boldsymbol{x} = (x_1, 1) \text{ with } 0 < x_1 < 1, \\ x_2 & \text{for } \boldsymbol{x} = (1, x_2) \text{ with } 0 < x_2 < 1, \\ -2 & \text{for } \boldsymbol{x} = (x_1, 0) \text{ with } 0 < x_1 < 1, \\ 0 & \text{for } \boldsymbol{x} = (0, x_2) \text{ with } 0 < x_2 < 1, \end{cases}$$

$$g(\boldsymbol{x}) = \begin{cases} 1 & \text{for } \boldsymbol{x} = (x_1, 1) \text{ with } 0 < x_1 < 1, \\ 0 & \text{for } \boldsymbol{x} = (1, x_2) \text{ with } 0 < x_2 < 1, \text{ for } \boldsymbol{x} = (0, x_2) \text{ with } 0 < x_2 < 1, \\ -1 & \text{for } \boldsymbol{x} = (x_1, 0) \text{ with } 0 < x_1 < 1. \end{cases}$$

We utilize piecewise linear finite element functions. The finite element solutions $y^h = y^h(t, \boldsymbol{x})$ for $t = 0.15$ and $t = T$ are shown in Figure 2.17. Next we take snapshots on the fixed uniform time grid $t_j = (j - 1)\Delta t$, $1 \leq j \leq n_t$, with $n_t = 10$ and $\Delta t = T/n_t = 0.1$. The goal is to determine four additional time instances $\bar{t} = (\bar{t}_1, \ldots, \bar{t}_4) \in [0, T]$ based on a finite element approximation for (2.44). Since the behavior of the solution exhibits more change during the initial time interval $[0, 0.3]$ than later on, we initialize our quasi-Newton method by the starting value $\tau^0 = (0.05, 0.15, 0.25, 0.35) \in [0, T]$. The number of POD ansatz functions is fixed to be $\ell = 3$. The corresponding value of the reduced-order model error is approximately 0.1093. The optimal solution is given as $\bar{\tau} = (0.0092, 0.0076, 0.1336, 0.2882) \in [0, T]$, while the associated reduced-order model error is approximately 0.0165, which is a reduction of about 85 %. In Figure 2.18 we can see that the shapes of the three POD bases change significantly from the initial time instances $\tau^0 \in \mathbb{R}^4$ to the optimal ones $\bar{\tau} \in \mathbb{R}^4$.

2.6 Optimal control with POD surrogate models

Reduced-order models are used in PDE-constrained optimization in various ways; see, e. g., [50, 86] for a survey. In optimal control problems it is sometimes necessary to
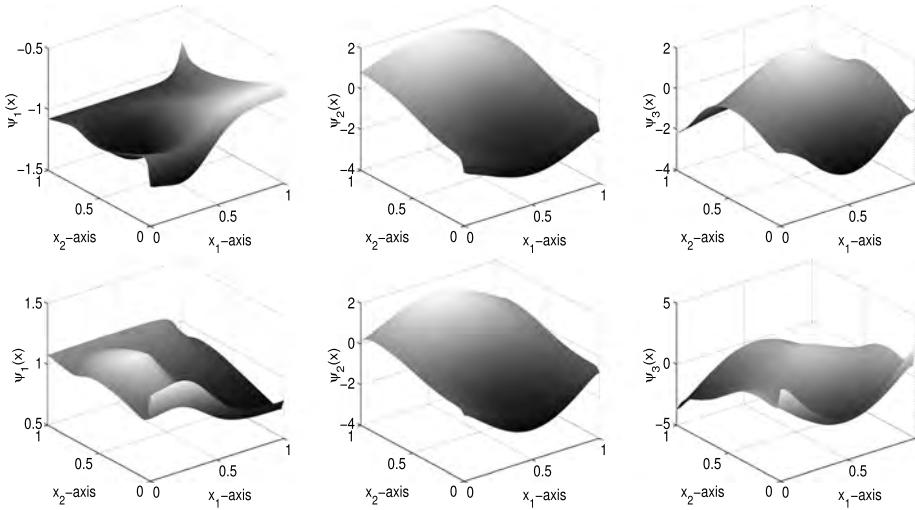
Figure 2.18: Run 2.10 (cf. [59, Figures 5 and 7]). POD basis Ψ_1, Ψ_2, Ψ_3 for the initial additional time instances $\tau^0 \in \mathbb{R}^4$ (upper three plots) and for the optimal additional time instances $\bar{\tau} \in \mathbb{R}^4$ (lower three plots).

compute a feedback control law instead of a fixed optimal control. In the implementation of these feedback laws models of reduced-order can play an important and very useful role; see [11, 40, 60, 65, 68, 79]. Another useful application is the use in optimization problems, where a PDE solver is part of the function evaluation. Obviously, thinking of a gradient evaluation or even a step size rule in the optimization algorithm, an expensive function evaluation leads to an enormous amount of computing time. Here, the reduced-order model can replace the system given by a PDE in the objective function. It is quite common that a PDE can be replaced by a five- or ten-dimensional system of ordinary differential equations. This results computationally in a very fast method for optimization compared to the effort for the computation of a single solution of a PDE. There is a large amount of literature in engineering applications in this regard; we mention only the papers [67, 71]. Recent applications can also be found in finance using the reduced models generated with the reduced basis method [76] and the POD model [85, 88] in the context of calibration for models in option pricing.

We refer to the survey article [42], where a linear quadratic optimal control problem in an abstract setting is considered. Error estimates for the POD-Galerkin approximations of the optimal control are proved. This is achieved by combining techniques from [28, 29, 44] and [56, 58]. For nonlinear problems we refer the reader to [50, 75, 86]. However, unless the snapshots are generating a sufficiently rich state space or are computed from the exact (unknown) optimal controls, it is not clear a priori how far the optimal solution of the POD problem is from the exact one. On the other hand, the POD method is a universal tool that is applicable also to problems with time-dependent coefficients or to nonlinear equations. Moreover, by generating snapshots from the

real (large) model, a space is constructed that inhibits the main and relevant physical properties of the state system. This, and its ease of use, makes POD very competitive in practical use, despite a certain heuristic flavor. In this context results for a POD a posteriori analysis are important; see, e. g., [93] and [41, 54, 55, 91, 92, 95, 97]. Using a fairly standard perturbation method it is deduced how far the suboptimal control, computed on the basis of the POD model, is from the (unknown) exact one. This idea turned out to be very efficient in our examples. It is able to compensate for the lack of a priori analysis for POD methods. Let us also refer to the papers [30, 36, 69], where a posteriori error bounds are computed for linear quadratic optimal control problems approximated by the reduced basis method.

Data- and/or simulation-based POD models depend on the data (e. g., initial values, right-hand sides, boundary conditions, observations, etc.) which are used to generate the snapshots. If those models are used as surrogates in, e. g., optimization problems with PDE constraints, the algorithmical framework has to account for this fact by providing mechanisms for accordingly updating the surrogate model during the solution process. Strategies proposed in this context for optimal flow control can be found in, e. g., [3, 4, 9, 34, 17]. One of the most mature methods developed in this context is trust-region POD, proposed in [9], which since then has successfully been applied in many applications. We also refer to the work [38], where strategies for updating the POD bases are compared.

The quality of the surrogate model highly depends on its information basis, which for snapshot-based methods is given by the snapshot set; compare Section 2.5. The location of snapshots and also the choice of the initial control in surrogate-based optimal control are discussed in [5]. There, techniques from time-adaptive schemes for optimality systems of parabolic optimal control problems are adjusted to compute optimal time locations for snapshots generation in POD surrogate modeling for parabolic optimal control problems.

Concepts for the construction and use of POD surrogate modeling in robust optimal control of electrical machines are presented in [63, 6]. Those problems are governed by nonlinear partial differential equations with uncertain parameters, so that robustness can be achieved by considering a worst case formulation. The resulting optimization problem then is of bilevel structure and POD-ROMs in combination with a posteriori error estimators are used to speed up the numerical computations.

2.7 Miscellaneous

POD model order reduction (POD-MOR) can also be applied to provide surrogate models for high-fidelity components in networks. The general perspective is discussed in, e. g., [48]. Related research for model order reduction of electrical networks is reported in, e. g., [16, 46, 47]. The basic idea here consists in a decoupling of MOR approaches

for the network and high-fidelity components which in general are modeled by PDE systems. For the latter, simulation-based POD-MOR techniques are used to construct surrogate models which then are stamped back into the (reduced) electrical network. Details and performance tests are reported, e. g., in [45, 47]. A short lecture series with related topics is presented under Hinze-Pilsen.[2] Further contributions to this topic can be found in [15].

Recent trends in data-driven and nonlinear MOR methods are discussed within a YouTube lecture series under Carlberg-YouTube.[3]

Bibliography

[1]   H. Abels, *Diffuse Interface Models for Two-Phase flows of Viscous Incompressible Fluids*, Lecture Note, vol. 36, Max-Planck Institut für Mathematik in den Naturwissenschaften, Leipzig, 2007.

[2]   H. Abels, H. Garcke, and G. Grün, Thermodynamically consistent, frame indifferent diffuse interface models for incompressible two-phase flows with different densities, *Mathematical Models and Methods in Applied Sciences*, **22** (3) (2012).

[3]   K. Afanasiev and M. Hinze, Adaptive control of a wake flow using proper orthogonal decomposition. Preprint No. 648/1999, Fachbereich Mathematik, TU Berlin, 1999.

[4]   K. Afanasiev and M. Hinze, Adaptive control of a wake flow using proper orthogonal decomposition, *Lecture Notes in Pure and Applied Mathematics*, **216** (2001), 317–332.

[5]   A. Alla, C. Gräßle, and M. Hinze, A-posteriori snapshot location for POD in optimal control of linear parabolic equations, *ESAIM: Mathematical Modelling and Numerical Analysis (M2AN)*, **52** (5) (2018), 1847–1873.

[6]   A. Alla, M. Hinze, P. Kolvenbach, O. Lass, and S. Ulbrich, A certified model reduction approach for robust parameter optimization with PDE constraints, *Advances in Computational Mathematics*, **45** (2019), 1221–1250.

[7]   A. Alla and J. N. Kutz, Nonlinear model order reduction via dynamic mode decomposition, *SIAM Journal on Scientific Computing*, **39** (2017), B778–B796.

[8]   M. S. Alnaes, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells, The FEniCS Project Version 1.5, *Archive of Numerical Software*, **100** (2015), 9–23.

[9]   E. Arian, M. Fahl, and E. W. Sachs, *Trust-region proper orthogonal decomposition for flow control*, Technical Report 2000-25, ICASE, 2000.

[10]   P. Astrid, S. Weiland, K. Willcox, and T. Backx, Missing point estimation in models described by proper orthogonal decomposition, *IEEE Transactions on Automatic Control*, **53** (2008), 2237–2251.

[11]   J. A. Atwell, J. T. Borggaard, and B. B. King, Reduced-order controllers for Burgers' equation with a nonlinear observer, *International Journal of Applied Mathematics and Computer Science*, **11** (2001), 1311–1330.

---

2 https://slideslive.com/38894790/mathematical-aspects-of-proper-orthogonal-decomposition-pod-iii

3 https://www.youtube.com/watch?v=KOHxCIx04Dg

[12] S. Banholzer, E. Makarov, and S. Volkwein, POD-based multiobjective optimal control of time-variant heat phenomena, *Lecture Notes in Computational Science and Engineering*, **126** (2019), 881–888.

[13] H. T. Banks, M. L. Joyner, B. Winchesky, and W. P. Winfree, Nondestructive evaluation using a reduced-order computational methodology, *Inverse Problems*, **16** (2000), 1–17.

[14] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, An "empirical interpolation" method: application to efficient reduced-basis discretization of partial differential equations, *Comptes Rendus de L'Académie des Sciences. Series 1, Mathematics*, **339** (2004), 667–672.

[15] P. Benner (ed.), *System Reduction for Nanoscale IC Design*, Springer International Publishing, Cham, Switzerland, 2017.

[16] P. Benner, M. Hinze, and E. J. W. ter Maten (eds.), *Model Reduction for Circuit Simulation*, Lecture Notes in Electrical Engineering, vol. 74, Springer, 2011.

[17] M. Bergmann and L. Cordier, Control of the cylinder wake in the laminar regime by Trust-Region methods and POD Reduced Order Models, *Journal of Computational Physics*, **227** (2009), 7813–7840.

[18] M. Bergmann, C.-H. Bruneau, and A. Iollo, Enablers for robust POD models, *Journal of Computational Physics*, **228** (2009), 516–538.

[19] J. F. Blowey and C. M. Elliott, The Cahn-Hilliard gradient theory for phase separation with non-smooth free energy. Part I: Mathematical analysis, *European Journal of Applied Mathematics*, **2** (1991), 233–280.

[20] T. Bui-Thanh, *Model-constrained optimization methods for reduction of parameterized systems*. PhD thesis, MIT, USA, 2007.

[21] J. W. Cahn and J. E. Hilliard, Free energy of a non-uniform system. I. Interfacial free energy, *Journal of Chemical Physics*, **28** (1958), 258–267.

[22] D. Chapelle, A. Gariah, and J. Saint-Marie, Galerkin approximation with proper orthogonal decomposition: new error estimates and illustrative examples, *ESAIM: Mathematical Modelling and Numerical Analysis*, **46** (2012), 731–757.

[23] A. Chatterjee, An introduction to the proper orthogonal decomposition, *Current Science*, **78** (2000), 539–575.

[24] S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM Journal on Scientific Computing*, **32** (2010), 2737–2764.

[25] Y. Chen, *Model order reduction for nonlinear systems*. Master's thesis, Massachusetts Institute of Technology, 1999.

[26] M. Damodoran, T. Bui-Thanh, and K. Willcox, Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition, *AIAA Journal*, **42** (2004), 1505–1516.

[27] R. Dautray and J.-L. Lions, *Mathematical Analysis and Numerical Methods for Science and Technology. Volume 5: Evolution Problems I*, Springer, Berlin, Heidelberg, New York, 2000.

[28] K. Deckelnick and M. Hinze, Error estimates in space and time for tracking-type control of the instationary Stokes system, *International Series of Numerical Mathematics*, **143** (2002), 87–103.

[29] K. Deckelnick and M. Hinze, Semidiscretization and error estimates for distributed control of the instationary Navier-Stokes equations, *Numerische Mathematik*, **97** (2004), 297–320.

[30] L. Dede, Reduced basis method and a posteriori error estimation for parametrized linear-quadratic optimal control problems, *SIAM Journal on Scientific Computing*, **32** (2010), 997–1019.

[31] Z. Drmac and S. Gugercin, A new selection operator for the discrete empirical interpolation method – improved a priori error bound and extension, *SIAM Journal on Scientific Computing*, **38** (2) (2016), A631–A648.

[32] R. Everson and L. Sirovich, The Karhunen-Loève precedure for gappy data, *Journal of the Optical Society of America*, **12** (1995), 1657–1664.

[33] D. J. Eyre, *Unconditionally Gradient Stable Time Marching the Cahn-Hilliard Equation*, MRS Proceedings, vol. 529, 1998.

[34] M. Fahl, Trust region methods for flow control based reduced order modeling. Dissertation, Fachbereich IV, Universität Trier, 2000.

[35] L. Feng, X. Zeng, C. Chiang, D. Zhou, and Q. Fang, Direct nonlinear order reduction with variational analysis, in *Proc. Design, Automation and Test in Europe*, pp. 1530–1591, 2004.

[36] M. Grepl and M. Kärcher, A posteriori error estimation for reduced order solutions of parametrized parabolic optimal control problems, *Mathematical Modelling and Numerical Analysis*, **48** (2014), 1615–1638.

[37] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, *ESAIM: Mathematical Modelling and Numerical Analysis*, **41** (2007), 575–605.

[38] C. Gräßle, M. Gubisch, S. Metzdorf, S. Rogg, and S. Volkwein, POD basis updates for nonlinear PDE control, *at – Automatisierungstechnik*, **65** (2017), 298–307.

[39] C. Gräßle and M. Hinze, POD reduced-order modeling for evolution equations utilizing arbitrary finite element discretizations, *Advances in Computational Mathematics*, **44** (2018), 1941–1978.

[40] L. Grüne, L. Mechelli, S. Pirkelmann, and S. Volkwein, Performance estimates for economic model predictive control and their application in POD-based implementations. To appear in *System Modeling and Optimization, Proceedings of the 28th IFIP TC 7 Conference 2018 on System Modelling and Optimization*, 2020.

[41] M. Gubisch and S. Volkwein, POD a-posteriori error analysis for optimal control problems with mixed control-state constraints, *Computational Optimization and Applications*, **58** (2014), 619–644.

[42] M. Gubisch and S. Volkwein, Proper orthogonal decomposition for linear-quadratic optimal control, in P. Benner, A. Cohen, M. Ohlberger, and K. Willcox (eds.), *Model Reduction and Approximation: Theory and Algorithms*, pp. 5–66, SIAM, Philadelphia, PA, 2017.

[43] M. Hintermüller, M. Hinze, and M. H. Tber, An adaptive finite element Moreau-Yosida-based solver for a non-smooth Cahn-Hilliard problem, *Optimization Methods & Software*, **26** (2011), 777–811.

[44] M. Hinze, A variational discretization concept in control constrained optimization: the linear-quadratic case, *Computational Optimization and Applications*, **30** (2005), 45–61.

[45] M. Hinze and M. Kunkel, Residual based sampling in POD model order reduction of drift-diffusion equations in parametrized electrical networks, *Zeitschrift für Angewandte Mathematik und Mechanik*, **92** (2012), 91–104.

[46] M. Hinze, M. Kunkel, U. Matthes, and M. Vierling, Model order reduction of integrated circuits in electrical networks, in P. Benner (ed.), *System Reduction for Nanoscale IC Design Hamburger*. Mathematics in Industry, vol. 20, 2014.

[47] M. Hinze, M. Kunkel, A. Steinbrecher, and T. Stykel, Model order reduction of coupled circuit-device systems, *International Journal of Numerical Modelling*, **25** (2012), 362–377.

[48] M. Hinze and U. Matthes, Model order reduction for networks of ODE and PDE systems, in D. Hömberg and F. Tröltzsch (eds.), *CSMO 2011*, IFIP AICT, vol. 391, pp. 92–101, 2013.

[49] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, Springer-Verlag, Berlin, 2009.

[50] M. Hinze and S. Volkwein, Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: error estimates and suboptimal control, *Lecture Notes in Computational Science and Engineering*, **45** (2005), 261–306.

[51]   M. Hinze and S. Volkwein, Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition, *Computational Optimization and Applications*, **39** (2008), 319–345.

[52]   P. C. Hohenberg and B. I. Halperin, Theory of dynamic critical phenomena, *Reviews of Modern Physics*, **49** (1977), 435–479.

[53]   P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge Monographs on Mechanics, 2nd ed., Cambridge University Press, Cambridge, 2012.

[54]   M. Kahlbacher and S. Volkwein, POD a-posteriori error based inexact SQP method for bilinear elliptic optimal control problems, *Mathematical Modelling and Numerical Analysis*, **46** (2012), 491–511.

[55]   E. Kammann, F. Tröltzsch, and S. Volkwein, A method of a-posteriori error estimation with application to proper orthogonal decomposition, *Mathematical Modelling and Numerical Analysis*, **47** (2013), 555–581.

[56]   K. Kunisch and S. Volkwein, Galerkin proper orthogonal decomposition methods for parabolic problems, *Numerische Mathematik*, **90** (2001), 117–148.

[57]   K. Kunisch and S. Volkwein, Crank-Nicolson Galerkin proper orthogonal decomposition approximations for a general equation in fluid dynamics, in *Proceedings of the 18th GAMM Seminar on Multigrid and Related Methods for Optimization Problems*, pp. 97–114, 2002.

[58]   K. Kunisch and S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, *SIAM Journal on Numerical Analysis*, **40** (2002), 492–515.

[59]   K. Kunisch and S. Volkwein, Optimal snapshot location for computing POD basis functions, *ESAIM: Mathematical Modelling and Numerical Analysis*, **44** (2010), 509–529.

[60]   K. Kunisch, S. Volkwein, and L. Xie, HJB-POD based feedback design for the optimal control of evolution problems, *SIAM Journal on Applied Dynamical Systems*, **3** (2004), 701–722.

[61]   S. Lall, J. E. Marsden, and S. Glavaski, A subspace approach to balanced truncation for model reduction of nonlinear control systems, *International Journal of Robust and Nonlinear Control*, **12** (2002), 519–535.

[62]   J. Lang, S. Ullmann, and M. Rotkvic, POD-Galerkin reduced-order modeling with adaptive finite element snapshots, *Journal of Computational Physics*, **325** (2016), 244–258.

[63]   O. Lass and S. Ulbrich, Model order reduction techniques with a posteriori error control for nonlinear robust optimization governed by partial differential equations, *SIAM Journal on Scientific Computing*, **39** (2016), S112–S139.

[64]   O. Lass and S. Volkwein, Adaptive POD basis computation for parametrized nonlinear systems using optimal snapshot location, *Computational Optimization and Applications*, **58** (2014), 645–677.

[65]   F. Leibfritz and S. Volkwein, Reduced order output feedback control design for PDE systems using proper orthogonal decomposition and nonlinear semidefinite programming, *Linear Algebra and Its Applications*, **415** (2006), 542–575.

[66]   A. Logg, K. A. Mardal, and G. Wells (eds.), Automated solution of differential equations by the finite element method, in *The FEniCS Book*. Lecture Notes in Computational Science and Engineering, vol. 84, Springer, 2012.

[67]   H. V. Ly and H. T. Tran, Modeling and control of physical processes using proper orthogonal decomposition, *Mathematical and Computer Modelling*, **33** (2001), 223–236.

[68]   L. Mechelli and S. Volkwein, POD-based economic model predictive control for heat-convection phenomena, *Lecture Notes in Computational Science and Engineering*, **126** (2019), 663–670.

[69]   F. Negri, G. Rozza, A. Manzoni, and A. Quateroni, Reduced basis method for parametrized elliptic optimal control problems, *SIAM Journal on Scientific Computing*, **35** (2013), A2316–A2340.

[70]   N. C. Nguyen, A. T. Patera, and J. Peraire, A "best point" interpolation method for efficient approximation of parametrized functions, *International Journal for Numerical Methods in Engineering*, **73** (2008), 521–543.

[71]   B. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, *Journal of Fluid Mechanics*, **497** (2003), 335–363.

[72]   A. T. Patera and G. Rozza, *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations*, MIT-Pappalardo Graduate Monographs in Mechanical Engineering, Massachusetts Institute of Technology, (C) MIT, Cambridge, MA, US, 2007.

[73]   J. R. Phillips, Automated extraction of nonlinear circuit macromodels, in *Proc. Custom Integrated Circuit Conf.*, pp. 451–454, 2000.

[74]   J. R. Phillips, Projection-based approaches for model reduction of weakly nonlinear, time-varying systems, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **22** (2003), 171–187.

[75]   R. Pinnau, Model reduction via proper orthogonal decomposition, in W. H. A. Schilder, H. A. van der Vorst, and J. Rommes (eds.), *Model Order Reduction: Theory, Research Aspects and Applications*. Mathematics in Industry, vol. 13, pp. 95–109, Springer, Berlin, Heidelberg, 2008.

[76]   O. Pironneau, Calibration of options on a reduced basis, *Journal of Computational and Applied Mathematics*, **232** (2006), 139–147.

[77]   R. Rannacher, Finite element solution of diffusion problems with irregular data, *Numerische Mathematik*, **43** (1984), 309–327.

[78]   M. Rathinam and L. Petzold, Dynamic iteration using reduced order models: a method for simulation of large scale modular systems, *SIAM Journal on Numerical Analysis*, **40** (2002), 1446–1474.

[79]   S. S. Ravindran, Reduced-order adaptive controllers for fluid flows using POD, *SIAM Journal on Scientific Computing*, **15** (2000), 457–478.

[80]   S. S. Ravindran, Error analysis for Galerkin POD approximation of the nonstationary Boussinesq equations, *Numerical Methods for Partial Differential Equations*, **27** (2011), 1639–1665.

[81]   M. Reed and B. Simon, *Methods of Modern Mathematical Physics I: Functional Analysis*, Academic Press, New York, 1980.

[82]   C. W. Rowley, Model reduction for fluids, using balanced proper orthogonal decomposition, *International Journal of Bifurcation and Chaos*, **15** (2005), 997–1013.

[83]   J. P. Raymond and H. Zidani, Hamiltonian Pontryagin's Principles for control problems governed by semilinear parabolic equations, *Applied Mathematics & Optimization*, **39** (1999), 143–177.

[84]   M. Rewiewski and J. White, A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **22** (2003), 155–170.

[85]   E. W. Sachs and M. Schu, A priori error estimates for reduced order models in finance, *ESAIM: M2AN*, **47** (2013), 449–469.

[86]   E. W. Sachs and S. Volkwein, POD Galerkin approximations in PDE-constrained optimization, *GAMM-Mitteilungen*, **33** (2010), 194–208.

[87]   A. Schmidt and K. G. Siebert, *Design of Adaptive Finite Element Software: The Finite Element Toolbox ALBERTA*, Lecture Notes in Computational Science and Engineering, vol. 42, Springer, 2005.

[88]   M. Schu, *Adaptive Trust-Region POD Methods and their Application in Finance*. PhD thesis, University of Trier, 2013.

[89]   J. R. Singler, New POD expressions, error bounds, and asymptotic results for reduced order models of parabolic PDEs, *SIAM Journal on Numerical Analysis*, **52** (2014), 852–876.

[90]   L. Sirovich, Turbulence and the dynamics of coherent structures. Parts I–II, *Quarterly of Applied Mathematics*, **XVL** (1987), 561–590.

[91]   A. Studinger and S. Volkwein, Numerical analysis of POD a-posteriori error estimation for optimal control, *International Series of Numerical Mathematics*, **164** (2013), 137–158.

[92]   T. Tonn, K. Urban, and S. Volkwein, Comparison of the reduced-basis and POD a-posteriori error estimators for an elliptic linear quadratic optimal control problem, *Mathematical and Computer Modelling of Dynamical Systems*, **17** (2011), 355–369.

[93]   F. Tröltzsch and S. Volkwein, POD a-posteriori error estimates for linear-quadratic optimal control problems, *Computational Optimization and Applications*, **44** (2009), 83–115.

[94]   S. Ullmann, M. Rotkvic, and J. Lang, POD-Galerkin reduced-order modeling with adaptive finite element snapshots, *Journal of Computational Physics*, **325** (2016), 244–258.

[95]   S. Volkwein, Optimality system POD and a-posteriori error analysis for linear-quadratic problems, *Control and Cybernetics*, **40** (2011), 1109–1125.

[96]   S. Volkwein, Optimal control of a phase-field model using proper orthogonal decomposition, *Zeitschrift für Angewandte Mathematik und Mechanik*, **81** (2001), 83–97.

[97]   G. Vossen and S. Volkwein, Model reduction techniques with a-posteriori error analysis for linear-quadratic optimal control problems, *Numerical Algebra, Control and Optimization*, **2** (2012), 465–485.

[98]   Z. Whang, Nonlinear model reduction based on the finite element method with interpolated coefficients: semilinear parabolic equations, *Numerical Methods for Partial Differential Equations*, **31** (2015), 1713–1741.

[99]   D. Wells, Z. Wang, X. Xie, and T. Iliescu, An Evolve-Then-Filter Regularized Reduced Order Model For Convection-Dominated Flows. arXiv:1506.07555v2, 2018.

[100]   K. Willcox and J. Peraire, Balanced model reduction via the proper orthogonal decomposition, *AIAA Journal*, **40** (2002), 2323–2330.

Francisco Chinesta and Pierre Ladevèze

3 Proper generalized decomposition

Abstract: The so-called "reduced" models have always been very popular and often essential in engineering to analyze the behavior of structures and materials, especially in dynamics. They highlight the relevant information and lead, moreover, to less expensive and more robust calculations. In addition to conventional reduction methods, a generation of reduction strategies is now being developed, such as proper generalized decomposition (PGD), which is the subject of this chapter. The primary feature of these strategies is to be very general and to offer enormous potential for solving problems beyond the reach of industrial computing codes. It is typically the case when trying to take into account the uncertainties or the variations of parameters or nonlinear problems with very large number of degrees of freedom, in the presence of several scales or interactions between several physics. These methods, along with the notions of "offline" and "online" calculations, also open the way to new approaches where simulation and analysis can be carried out almost in real-time. What distinguishes PGD from proper orthogonal decomposition (POD) and reduced basis is the calculation procedure that does not differentiate between the different variables parameters/time/space. In other terms, we can say that we minimize or make stationary a residual defined over the parameters-time-space domain. PGD with time/space separation and the classical greedy computation technique were introduced in the 1980s as part of the LATIN solver [66, 67] for solving nonlinear time-dependent problems with the terminology "time/space radial approximation." The corpus of literature devoted to this method is vast [68, 77] but remained in the form of time/space separations for many years. A more general separated representation was more recently employed in [5, 6] for approximating the solution of multidimensional partial differential equations. In [93], such separated representations are also considered for solving stochastic equations. PGD is the common name coined in 2010 by the authors of this chapter for these techniques because it can be viewed as an extension of the classical POD. Today, many works use and develop the PGD in extremely varied fields. In this chapter we revisit the fundamentals, variants, and applications of PGD, covering different kinds of separated representations of the involved unknown fields as well as different constructors able to address a variety of linear and nonlinear models, elliptic, parabolic, and hyperbolic.

Francisco Chinesta, Arts et Metiers Institute of Technology, Paris, France
Pierre Ladevèze, ENS Paris-Saclay, Paris, France

In this chapter we use different notations to be consistent with the referred publications. In any case, notation will be appropriately defined before being used to avoid any possible confusion.

Keywords: MOR, PGD, separated representations, real-time, parametric models, plane-out-of-plane separated representations

MSC 2010: 35C99, 35Q99, 39A99, 4104, 65L99, 65M99, 65N99, 65Z05, 74C99, 74K99, 74S99

3.1 PGD: fundamentals

3.1.1 Principles

The proper generalized decomposition (PGD) method belongs together with reduced basis and proper orthogonal decomposition (POD) methods to the last generation of reduced-order model (ROM) computational methods characterized by a very large scope of applications. They are all based on the same concepts. The first and main idea in these methods is that the shape functions are not a priori given as usual. They are computed simultaneously with the solution itself thanks to an iterative procedure. For a problem defined over a parameter-time-space domain, the solution is then written as

$$s_m(\mu, t, X) = \sum_{i=1}^{m} a_i \Psi_i(\mu, t, X) \quad \text{over } \Sigma_\mu \times [0, T] \times \Omega, \tag{3.1}$$

where Ψ_i are the normalized shape functions.

The second idea is to introduce a variable separation hypothesis or something equivalent:

$$\Psi_i(\mu, t, X) = \gamma_i(\mu)\lambda_i(t)\Lambda_i(X), \tag{3.2}$$

where the scalar time functions λ_i, the scalar parameter functions γ_i, and the space functions Λ_i are arbitrary. That is a deflection from other approximation methods for which shape functions are a priori given or partially given. Very often, a low-rank canonical format approximation is used.

PGD is characterized by a global residual defined over the parameter-time-space domain which should be minimized. Let R be this residual, given as

$$R(s) = \int_{\Sigma_\mu} \int_0^T \int_\Omega d\mu_1 \dots d\mu_n dt d\Omega \, r(s) \tag{3.3}$$

with

$$r \geq 0; \quad r = 0 \quad \text{over } \Sigma_\mu \times [0, T] \times \Omega \iff s = s_{\text{exact}}. \tag{3.4}$$

The problem to solve is then

$$\min R\left(\sum_{i=1}^{m} \gamma_i^1(\mu_1) \dots \gamma_i^n(\mu_n) \lambda_i(t) \Lambda_i(X) \right), \tag{3.5}$$

which is twice nonlinear. From a mechanical point of view, it could be nonlinear. Furthermore, the computation of a PGD approximation is always a nonlinear problem. To solve this minimization problem, the common technique is a "greedy" one. Such a technique has been introduced for PGD at the end of the 1980s [68]. It is an iterative technique in which, at iteration $m + 1$, the correction $\Delta s = s_{m+1} - s_m$ minimizes the residual:

$$\Delta s = \gamma^1(\mu_1) \dots \gamma^n(\mu_n) \lambda(t) \Lambda(X); \quad \min_{\gamma^1, \dots, \gamma^n, \lambda, \Gamma} R(s_m + \Delta s). \tag{3.6}$$

The residual is minimized alternatively over the scalar parameter functions, the scalar time function, and the space function. Practically, few iteration loops are performed to get the new PGD mode. It has also been numerically observed that better performance could be obtained by updating the time and parameter functions of the previous PGD modes before computing a new PGD mode by minimizing the global residual.

The classical PGD format has a tensorial equivalent: the canonical polyadic (CANDECOMP/PARAFAC, CP) format. The greedy procedure used in the PGD method is very close to tensorial tools such as the greedy CP alternating least-squares (ALS) algorithm [61], one major difference being the norm used: This algorithm minimizes the Frobenius norm of the error, while any norm, such as an energetic one, can be used in the PGD procedure. Other classical tensor formats can be used. The Tucker format, computed from an initial full tensor through higher-order singular value decomposition (SVD), is a compressed representation which gives a tensorial equivalent of the matrix concept of SVD. More recent structured representations have been developed such as tensor train or the hierarchical Tucker format. The former is a generic and simple format which can have a better compression ratio than the CP decomposition. The latter is the more general structured format and includes all the previously presented ones. Its use can be delicate as the potentially complex structure must be chosen before any computation. These two formats have both been adapted to the resolution of mechanical partial differential equations in [99].

Remark. The simplest case is the situation where one computes the PGD of a given function defined over the time-space domain. It was proved in this case that convergence properties could be obtained and that the time functions are the eigenfunctions of a certain eigenvalue problem [68]. For the L^2-norm and the discretized problem, the so-called PGD corresponds exactly to the classical SVD. It follows that PGD modes can be seen as "eigenmodes" and that the PGD can be seen as an extension of the SVD to partial differential equations.

3.1.2 Different types of separated representations

Most of the existing model reduction techniques proceed by extracting a suitable reduced basis and then projecting on it the problem solution. Thus, the reduced basis construction precedes its use in the solution procedure, and one must be careful on the suitability of a particular reduced basis when employed for representing the solution of a particular problem. This issue disappears if the approximation basis is constructed at the same time as the problem is solved. Thus, each problem has its associated basis in which its solution is expressed. One could consider few terms in its approximation, leading to a reduced representation, or all the terms needed for approximating the solution up to a certain accuracy level.

When calculating the transient solution of a generic problem $u(x, t)$ we usually consider a given basis of space functions $N_i(x)$, $i = 1, \ldots, N_n$, the so-called shape functions within the finite element framework, and approximate the problem solution as

$$u(x, t) \approx \sum_{i=1}^{N_n} a_i(t) N_i(x), \tag{3.7}$$

which implies a space-time separated representation where the time-dependent coefficients $a_i(t)$ are unknown at each time (when proceeding incrementally) and the space functions $N_i(x)$ are given "a priori," e. g., by means of a polynomial basis.

POD and reduced basis methodologies consider a reduced basis $\phi_i(x)$ for approximating the solution instead of using the generic functions $N_i(x)$. The former are expected to be more suitable for approximating the problem at hand. Thus, it results that

$$u(x, t) \approx \sum_{i=1}^{R} b_i(t) \phi_i(x), \tag{3.8}$$

where in general $R \ll N_n$. Again (3.8) represents a space-time separated representation where the time-dependent coefficient must be calculated at each time during the incremental solution procedure.

Inspired by these results one could consider the general space-time separated representation

$$u(x, t) \approx \sum_{i=1}^{N} X_i(x) T_i(t), \tag{3.9}$$

where now neither the time-dependent functions $T_i(t)$ nor the space functions $X_i(x)$ are "a priori" known. Both will be computed on-the-fly when solving the problem.

As soon as one postulates that the solution of a transient problem can be expressed in the separated form (3.9) whose approximation functions $X_i(x)$ and $T_i(t)$ will be determined during the problem solution, one could make a step forward and as-

sume that the solution of a multidimensional problem $u(x_1, \ldots, x_d)$ could be found in the separated form

$$u(x_1, x_2, \ldots, x_d) \approx \sum_{i=1}^{N} X_i^1(x_1) X_i^2(x_1) \ldots X_i^d(x_d). \tag{3.10}$$

Consider a problem defined in a high-dimensional space of dimension d for the unknown field $u(x_1, \ldots, x_d)$. Here, the coordinates x_i denote any usual coordinate (scalar or vectorial) related to space, time, and/or any conformational coordinate.

We seek a solution for $u(x_1, \ldots, x_d) \in \Omega_1 \times \cdots \times \Omega_d$. PGD yields an approximate solution in the separated form

$$u(x_1, \ldots, x_d) \approx \sum_{i=1}^{N} X_i^1(x_1) \cdots X_i^d(x_d) = \sum_{i=1}^{N} \prod_{j=1}^{d} X_i^j(x_j). \tag{3.11}$$

If N_n nodes are used to discretize each coordinate, the total number of PGD unknowns is $N \cdot N_n \cdot d$ instead of the $(N_n)^d$ degrees of freedom involved in standard mesh-based discretizations. Thus, the high-dimensional solution is computed by solving a number of low-dimensional problems alleviating the so-called curse of dimensionality involved in high-dimensional models.

Separated representations within the PGD framework were applied for solving the multidimensional Fokker–Planck equation describing complex fluids within the kinetic theory framework in [5, 6]. The solution procedure was extended to nonlinear kinetic theory descriptions of more complex molecular models in [86]. In [81] authors considered multibead-spring models but used a spectral approximation for representing all the functions involved in the finite sums decomposition. A deeper analysis of nonlinear and transient models was considered in [8]. Complex fluid models were coupled with complex flows in [105] and [87] opening very encouraging perspectives and pointing out the necessity of defining efficient stabilizations. A first tentative of convective stabilization was proposed in [56]. Finally, in [34] PGD was applied for solving the stochastic equation within the Brownian configuration field framework.

Multidimensional models encountered in the finer descriptions of matter (ranging from quantum chemistry to statistical mechanics descriptions) were revisited in [7]. The multidimensional chemical master equation was solved in [13] and the Langer equation governing phase transitions was solved in [79].

The solution of a parametric problem $u(\mathbf{x}, t, \mu_1, \ldots, \mu_P)$ (widely considered in the present chapter) can be expressed as

$$u(\mathbf{x}, t, \mu_1, \ldots, \mu_P) \approx \sum_{i=1}^{N} X_i(\mathbf{x}) T_i(t) \prod_{k=1}^{P} M_i^k(\mu_k), \tag{3.12}$$

where parameters are considered as model extra-coordinates.

Many times the spatial domain Ω, assumed three-dimensional, can be fully or partially separated, and consequently it can be expressed as $\Omega = \Omega_x \times \Omega_y \times \Omega_z$ or

$\Omega = \Omega_{xy} \times \Omega_z$, respectively. The first decomposition is related to hexahedral domains whereas the second one is related to plates, beams, or extruded domains. We consider both scenarios.

– The spatial domain Ω is partially separable. In this case the separated representation reads

$$u(\mathbf{x}, z, t) \approx \sum_{i=1}^{N} X_i(\mathbf{x}) Z_i(z) T_i(t), \tag{3.13}$$

where $\mathbf{x} = (x, y) \in \Omega_{xy}$, $z \in \Omega_z$ and $t \in \Omega_t$. Thus, iteration p of the alternated directions strategy at a given enrichment step n consists of:

1. solving in Ω_{xy} a two-dimensional boundary value problem (BVP) to obtain function X_n^p,
2. solving in Ω_z a one-dimensional BVP to obtain function Z_n^p,
3. solving in Ω_t a one-dimensional initial value problem (IVP) to obtain function T_n^p.

The complexity of the PGD simulation scales with the two-dimensional mesh used to solve the BVPs in Ω_{xy}, regardless of the mesh and the time step used in the solution of the BVP and the IVPs defined in Ω_z and Ω_t for calculating functions $Z_i(z)$ and $T_i(t)$.

– The spatial domain Ω is fully separable. In this case the separated representation reads

$$u(x, y, z, t) = \sum_{i=1}^{N} X_i(x) Y_i(y) Z_i(z) T_i(t). \tag{3.14}$$

Iteration p of the alternated directions strategy at a given enrichment step n consists of:

1. solving in Ω_x a one-dimensional BVP to obtain function X_n^p,
2. solving in Ω_y a one-dimensional BVP to obtain function Y_n^p,
3. solving in Ω_z a one-dimensional BVP to obtain function Z_n^p,
4. solving in Ω_t a one-dimensional IVP to obtain function T_n^p.

The cost savings provided by PGD are potentially phenomenal when the spatial domain is fully separable. Indeed, the complexity of the PGD simulation now scales with the one-dimensional meshes used to solve the BVPs in Ω_x, Ω_y, and Ω_z, regardless of the time step used in the solution of the decoupled IVPs in Ω_t.

Even when the domain is not fully separable, a fully separated representation could be considered by using appropriate geometrical mappings or by immersing the non-separable domain into a fully separable one. The interested reader can refer to [54] and [51].

In-plane-out-of-plane separated representations are particularly useful for addressing the solution of problems defined in plates [21], shells [22], or extruded domains [82]. A parametric three-dimensional elastic solution of beams involved in

frame structures was proposed in [23]. The same approach was extensively considered in structural plate and shell models in [48, 114–118, 107]. Space separated representations where enriched with discontinuous functions for representing cracks in [53], delamination in [84], and thermal contact resistances in [38]. Domain decomposition within the separated space representation was accomplished in [88] and localized behaviors were addressed by using superposition techniques in [12].

The in-plane-out-of-plane decomposition was then extended to many other physics: Thermal models were considered in [38]; squeeze flows of Newtonian and non-Newtonian fluids in laminates in [52]; flows in stratified porous media in [35], nonlinear viscoplastic flows in plate domains in [26], and electromagnetic problems in [112]. A full space decomposition was also efficiently applied for solving the Navier–Stokes equations in the lid-driven cavity problem in [44–46].

3.1.3 Illustrating the simplest separated representation constructor

In order to illustrate the simplest procedure for constructing the separated representation we consider the one-dimensional heat transfer equation involving the temperature field $u(x, t)$,

$$\frac{\partial u}{\partial t} - k\frac{\partial^2 u}{\partial x^2} = f, \tag{3.15}$$

defined in the space-time domain $\Omega = \Omega_x \times \Omega_t = (0, L) \times (0, \tau]$. The diffusivity k and source term f are assumed constant. We specify homogeneous initial and boundary conditions, i. e., $u(x, t = 0) = u(x = 0, t) = u(x = L, t) = 0$. More details and more complex scenarios can be found in [36].

The weighted residual form of (3.15) reads

$$\int_{\Omega_x \times \Omega_t} u^* \left(\frac{\partial u}{\partial t} - k\frac{\partial^2 u}{\partial x^2} - f \right) dx\, dt = 0, \tag{3.16}$$

for all suitable test functions u^*.

Our objective is to obtain a PGD approximate solution in the separated form

$$u(x, t) \approx \sum_{i=1}^{N} X_i(x) T_i(t). \tag{3.17}$$

We do so by computing each term of the expansion at each step of an enrichment process, until a suitable stopping criterion is met.

Thus, at enrichment step n, the $n - 1$ first terms of the PGD approximation (3.17) are known:

$$u^{n-1}(x, t) = \sum_{i=1}^{n-1} X_i(x) T_i(t). \tag{3.18}$$

We now wish to compute the next term $X_n(x)T_n(t)$ to get the enriched PGD solution

$$u^n(x,t) = u^{n-1}(x,t) + X_n(x)T_n(t) = \sum_{i=1}^{n-1} X_i(x)T_i(t) + X_n(x)T_n(t). \tag{3.19}$$

One must thus solve a nonlinear problem for the unknown functions $X_n(x)$ and $T_n(t)$ by means of a suitable iterative scheme. The simplest strategy consists of an alternated direction fixed point algorithm, which at iteration p reads

$$u^{n,p}(x,t) = u^{n-1}(x,t) + X_n^p(x)T_n^p(t). \tag{3.20}$$

Starting from an arbitrary initial guess $T_n^0(t)$, the alternating direction strategy computes $X_n^p(x)$ from $T_n^{p-1}(t)$, and then $T_n^p(t)$ from $X_n^p(x)$. These nonlinear iterations proceed until reaching a fixed point within a user-specified tolerance ϵ, i. e.,

$$\left\| X_n^p(x) \cdot Y_n^p(y) - X_n^{p-1}(x) \cdot Y_n^{p-1}(y) \right\| < \epsilon, \tag{3.21}$$

where $\| \cdot \|$ is a suitable norm. The enrichment step n thus ends with the assignments $X_n(x) \leftarrow X_n^p(x)$ and $T_n(t) \leftarrow T_n^p(t)$.

The enrichment process itself stops when an appropriate measure of error $\mathcal{E}(n)$ becomes small enough, i. e., $\mathcal{E}(n) < \tilde{\epsilon}$.

For additional details the interested reader can refer to [38], where the problems related to the calculation of functions X_n^p and T_n^p were defined, as well as more complex scenarios involving two-dimensional and high-dimensional problems.

3.1.4 Convergence properties

The convergence of the greedy technique is demonstrated for elliptic linear operators in classical separation cases (space/space, parameters/space) [10, 27, 62, 80]. However, the estimates of the convergence rate remain crude in the sense that they do not reflect what is observed in practice. For eigenvalue problems, convergence properties were given in [28]. For nonlinear problems, there are few results. However, the convergence of PGD is shown for convex problems in the sense that the set of PGD-like solutions is dense in the admissible space [63].

3.1.5 Verification

3.1.5.1 A posteriori error estimators and adaptive computational approaches

As any numerical method, PGD is associated with error sources which need to be effectively assessed and controlled, using a posteriori error estimation, in order to certify the accuracy of the results and then permit the transfer and intensive use of PGD-ROMs

in industrial activities for robust optimization and design. This is a main challenge in simulation-based engineering as identified in the report of the NSF Simulation-Based Engineering Science panel [100]. Moreover, one can resort to adaptive methods based on a posteriori error estimation for the construction of the reduced models in the hope of reducing the computational cost for a given accuracy.

Verification of PGD-ROMs has only been addressed in few works this last decade, in contrast to the vast literature dedicated to the control of the reduced basis method. A first attempt was considered in [9] using residual-based techniques for goal-oriented error estimation. In this work, mainly devoted to adaptivity, only the error coming from the truncation of the PGD modal representation was controlled and the error bounds were not guaranteed (the adjoint solution being approximated with a finer PGD decomposition). In [2], the approach of [9] was extended to the nonlinear context by using a linearized version of the problem to define the adjoint problem, before using a weighted residuals method with a higher number of PGD modes to represent the adjoint solution and catch the PGD truncation error. Even though this approach is cheap, it still cannot deliver guaranteed error bounds in which all error sources are taken into account.

In order to provide a general framework to obtain guaranteed, accurate, and fully computable bounds to effectively control the quality of PGD approximations, a robust a posteriori verification technique based on the constitutive relation error (CRE) concept [70, 31, 76] has been introduced in [73, 30, 74, 33, 113]. This was done in the context of parameterized linear elliptic or parabolic problems, the bounds being related to the global error (in the energy norm) or to specific quantities of interest (goal-oriented error estimation) using adjoint-based techniques [102, 104, 17, 75]. The error estimates involve all error sources including discretization error and truncation error in the PGD modal representation. They are based on the construction of perfectly equilibrated fields. The key technical point is to construct a finite element-equilibrated stress or flux vector from finite element equilibration properties of the computed PGD modes; after one uses similar tools as those used in the classical finite element analysis context. This is the mandatory procedure to recover strict bounds on discretization error which can be applied to other ROMs. A variant was also proposed in [85], even though equilibrated fields were here obtained using a dual PGD computational approach. Furthermore, the work in [33, 113] enables to split error sources by means of specific error indicators, which helps driving greedy adaptive algorithms to drive computations and optimize CPU time and memory space for a prescribed error tolerance; this comes down to defining a suitable PGD approximation in terms of the required number of terms in the modal representation of the solution, but also in terms of the discretization meshes used to compute modes. Consequently, the verification procedure based on CRE certifies the quality of the PGD approximation (globally or on specific outputs of interest) over the whole set of possible model parameters, and enables to adapt the PGD solution towards the specific goals of the computer simulation. The method was

illustrated in [33] with several numerical experiments on two-dimensional and three-dimensional mechanical problems; one of them is given at the end of this section. Let us note that advection problems were not considered in the previously mentioned works, even though these could be extended to such nonsymmetric problems with minor changes using ingredients given in [103, 60]. Extension to nonlinear problems could also be performed using [72] to get guaranteed bounds. Nevertheless, some CRE error indicators which are not guaranteed bounds are easy to compute from [68].

Eventually, we mention some other works in which PGD model reduction and verification methods are conjointly addressed. In [122], the effect of the separated approximation of input data in the accuracy of the resulting PGD solution was studied, from empirical and numerical considerations. In [31, 4], PGD was used to compute the CRE-based error estimate itself. In these latter works, the construction of equilibrated fields was facilitated using a PGD representation of the solution at the element level in the finite element mesh, parameterizing material properties and element shape. In addition to making the implementation of CRE into commercial finite element software easier, it was shown that the use of PGD enabled to optimize the verification procedure and to get both accurate and reasonably expensive upper bounds on the discretization error.

3.1.5.2 Error-driven PGD computation

Error analysis and computation can lead to efficient and robust PGD computation methods. In [65], the objective was to derive a reduced-order formulation such that the accuracy in given quantities of interest is increased when compared to a standard PGD method. Contrary to traditional goal-oriented methods that usually compute the solution of an adjoint problem following the calculation of the primal solution for error estimation and adaptation, it was proposed in this work to solve the adjoint problem first, based on a reduced approach, in order to extract estimates of the quantities of interest and use this information to constrain the reduced primal problem. This approach shares similarities with the work described in [20], where the authors define specific norms with additional weighting terms taking into account the error in the quantity of interest. The main idea in [20] is to minimize a norm weighted by a functional involving the adjoint solution, via a penalization approach, in order to obtain a goal-oriented PGD using the so-called ideal minimal residual approach.

Eventually, a new and promising PGD computational method based on the minimization of the CRE measure was proposed and analyzed in [3, 32]. In addition to enhancing the computation of PGD modes, it provides an improved, immediate, and robust reduction error estimation. This technique has been extended to solid mechanics nonlinear time-dependent problems [77].

3.1.5.3 Illustration

We present here a three-dimensional numerical experiment taken from [33], which illustrates the error estimation method and adaptive strategy. One considers a three-dimensional elasticity problem with three parameters for which the first PGD modes are computed; the associated errors as well as specific error indicators related to the discretization and the number of PGD modes are also given.

The domain is a cube of size $1\,\text{m} \times 1\,\text{m} \times 1\,\text{m}$ with three spherical inclusions for which Young's moduli $E_i \in [1, 10]$ ($1 \leq i \leq 3$) are parameters, so that the order m PGD representation reads $\mathbf{u}_m(\mathbf{x}, E_1, E_2, E_3)$. The three inclusions have the same radius $r = 0.1\,\text{m}$, and their centers are respectively located at points $c_1 = (0.2, 0.2, 0.2)$, $c_2 = (0.6, 0.3, 0.5)$, and $c_3 = (0.4, 0.7, 0.8)$ (Figure 3.1). The cube is clamped along the plane located at $x = 0$ and subjected to a unit traction force $\mathbf{F}_d = +\mathbf{x}$ applied on the plane located at $x = 1$. The initial finite element mesh contains 17,731 4-node tetrahedral elements and 3,622 nodes (10,866 degrees of freedom).
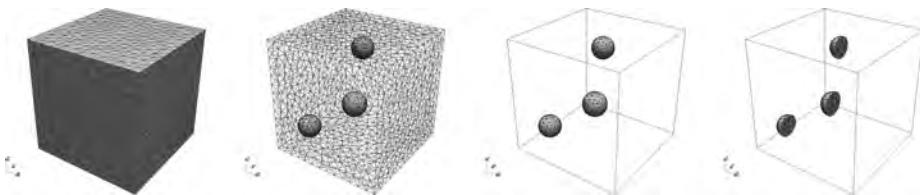

Figure 3.1: Three-dimensional elasticity problem: space domain with three inclusions and associated finite element mesh.

The first three PGD modes of the PGD approximate solution $\mathbf{u}_m^h$ are given in Figure 3.2 for space functions $\boldsymbol{\psi}_m(\mathbf{x})$ and in Figure 3.3 for parameter functions $\gamma_{1,m}(E_1)$, $\gamma_{2,m}(E_2)$, and $\gamma_{3,m}(E_3)$. Note that the first space function $\boldsymbol{\psi}_1(\mathbf{x})$ corresponds to a global mode, whereas the second and third space functions $\boldsymbol{\psi}_2(\mathbf{x})$ and $\boldsymbol{\psi}_3(\mathbf{x})$ are local modes mostly concentrated around the first and second inclusions, respectively.

The evolutions of the CRE-based error estimate E_{CRE} and associated error indicators η_{PGD} and η_{dis} with respect to the number m of PGD modes are shown in Figure 3.4 for $m = 1, \ldots, 6$ and for the maximal values obtained with triplets (E_1, E_2, E_3). This is represented along the adaptive strategy, and vertical evolutions indicate mesh refinements (which are performed each time the indicator associated with the discretization error is larger than the one associated with the PGD truncation error). Let us note that the error estimate converges quite fast toward the indicator associated to the discretization error (for a fixed mesh), while both error indicators decrease toward zero along the adaptive procedure. The computation cost associated with the error estimator and error indicators is of the same order as for the PGD computation.
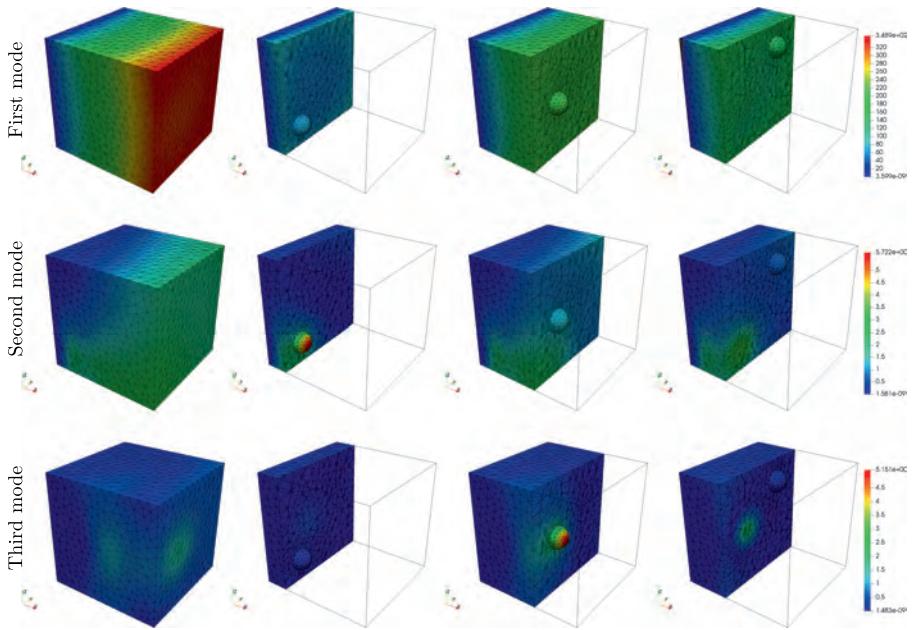
Figure 3.2: Magnitude of space functions $\psi_m(\mathbf{x})$ obtained for order $m = 1, \ldots, 3$ (from top to bottom).
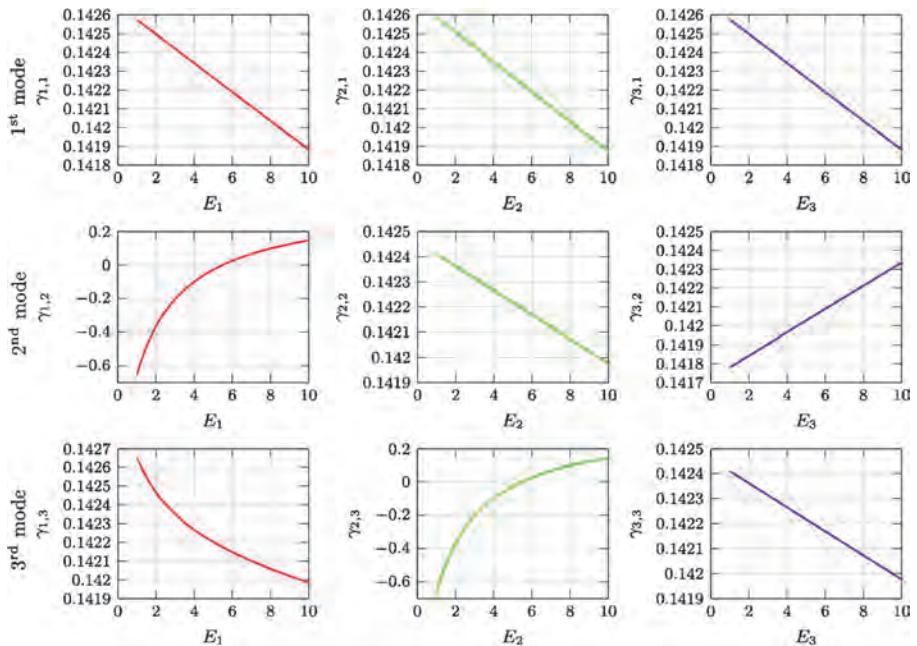

Figure 3.3: Parameter functions $\gamma_{1,m}(E_1)$, $\gamma_{2,m}(E_2)$, and $\gamma_{3,m}(E_3)$ (from left to right) obtained for order $m = 1, \ldots, 3$ (from top to bottom).
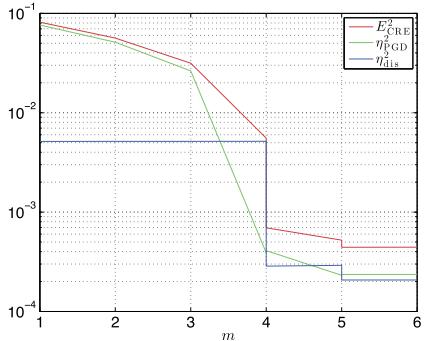
Figure 3.4: Evolutions of the error estimate E_{CRE}^2 and associated error indicators η_{PGD}^2 and η_{dis}^2 with respect to the number m of PGD modes.

3.1.6 Limits

PGD limits coincide with the limits of the variable separation hypothesis. Generally speaking, such limits could be reached by problems with moving loads. A thorough analysis was performed for a specific model problem in [3]; it follows that the quasi-stationary solution is not time-space separable with few modes. However, adding in the basis this particular solution (which is generally easy to compute), an accurate approximation can be obtained with few PGD modes. Transient dynamics also belongs to this class of problems where the time-space variable separation may not work well, in particular when high-frequency phenomena occur. That is clear, as considering a wave implies that the driven quantities are a combination of time and space variables. However, working over the frequency-space domain still enables to derive efficient PGD approximations [16].

3.2 PGD for nonlinear time-dependent problems

3.2.1 State of the art

Large time increment (LATIN)-PGD is a robust and effective tool for the construction of PGD in nonlinear solid mechanics, in the process of implementation in the industrial simulation codes; elsewhere, the approach it underlies should be extended to other parts of physics. The article [77] and the book [68] describe the state of the art. Until recent years, LATIN-PGD had no competitor except [110, 111], where POD is used in conjunction with a "hyperreduction" technique. Currently, many works using PGD or POD have appeared in the frame of the classical homogenization method for periodic media (in particular finite element square), among them [108, 121, 64, 59].

3.2.2 The LATIN-PGD computation method

The LATIN-PGD computation method has been derived to build ROMs in nonlinear solid mechanics but could be extended to other parts of physics.

3.2.2.1 Presentation of the problem

To present the method, with the assumption of small perturbations, let us consider the quasi-static and isothermal evolution of a structure defined over the time-space domain $[0, T] \times \Omega$. This structure is subjected to prescribed body forces $\boldsymbol{f}_d$, traction forces $\boldsymbol{F}_d$ over a part $\partial_2 \Omega$ of the boundary, and displacements $\boldsymbol{u}_d$ over the complementary part $\partial_1 \Omega$ (Figure 3.5).
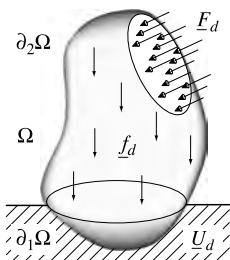

Figure 3.5: The reference problem.

The state of the structure is defined by the set of fields $\mathbf{s} = (\dot{\boldsymbol{\varepsilon}}_\mathrm{p}, \dot{\boldsymbol{X}}, \boldsymbol{\sigma}, \boldsymbol{Y})$ (where the dot notation $\dot{\square}$ denotes the time derivative), in which:

- $\boldsymbol{\varepsilon}_\mathrm{p}$ refers to the inelastic part of the strain field $\boldsymbol{\varepsilon}$ which corresponds to the displacement field $\boldsymbol{u}$, uncoupled into an elastic part $\boldsymbol{\varepsilon}_\mathrm{e}$ and an inelastic part $\boldsymbol{\varepsilon}_\mathrm{p} = \boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\mathrm{e}$; $\boldsymbol{X}$ refers to the remaining internal variables;
- $\boldsymbol{\sigma}$ refers to the Cauchy stress field and $\boldsymbol{Y}$ to the set of variables conjugate of $\boldsymbol{X}$ ($\boldsymbol{Y}$ and $\boldsymbol{X}$ have the same dimension); $\boldsymbol{X}$ could be hardening variables, damage variables, chemical variables, etc.

All these quantities are defined over the time-space domain $[0, T] \times \Omega$ and assumed to be sufficiently regular. For the sake of simplicity, the displacement $\boldsymbol{u}$ alone is assumed to have a nonzero initial value, denoted $\boldsymbol{u}_0$. Introducing the notations for the primal fields

$$\mathbf{e}_\mathrm{p} = \begin{bmatrix} \boldsymbol{\varepsilon}_\mathrm{p} \\ -\boldsymbol{X} \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} \boldsymbol{\varepsilon} \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{e}_\mathrm{e} = \begin{bmatrix} \boldsymbol{\varepsilon}_\mathrm{e} \\ \boldsymbol{X} \end{bmatrix} \quad \text{so that } \mathbf{e}_\mathrm{p} = \mathbf{e} - \mathbf{e}_\mathrm{e} \tag{3.22}$$

and for the dual fields

$$\mathbf{f} = \begin{bmatrix} \boldsymbol{\sigma} \\ \boldsymbol{Y} \end{bmatrix}, \tag{3.23}$$

the mechanical dissipation rate for the entire structure Ω is

$$\int_{\Omega} (\dot{\boldsymbol{\varepsilon}}_p : \boldsymbol{\sigma} - \dot{\boldsymbol{X}} \cdot \boldsymbol{Y}) d\Omega = \int_{\Omega} (\dot{\mathbf{e}}_p \circ \mathbf{f}) d\Omega, \tag{3.24}$$

where $\cdot$ denotes the contraction adapted to the tensorial nature of $\boldsymbol{X}$ and $\boldsymbol{Y}$. Notation $\circ$ denotes the contraction operator for generalized quantities. Let us now introduce the following fundamental bilinear "dissipation" form:

$$\langle \mathbf{s}, \mathbf{s}' \rangle = \int_{[0,T] \times \Omega} \left(1 - \frac{t}{T} \right) (\dot{\mathbf{e}}_p \circ \mathbf{f}' + \dot{\mathbf{e}}_p' \circ \mathbf{f}) d\Omega dt, \tag{3.25}$$

along with $\mathbf{E}$ and $\mathbf{F}$, the spaces of the fields $\dot{\mathbf{e}}_p$ and $\mathbf{f}$ which are compatible with (3.25). These spaces enable us to define $\mathbf{S} = \mathbf{E} \times \mathbf{F}$, the space in which the state $\mathbf{s} = (\dot{\mathbf{e}}_p, \mathbf{f})$ of the structure is being sought.

Following [67, 68], a normal formulation with internal state variables is used to represent the behavior of the material. If ρ denotes the mass density of the material, from the free energy $\rho\Psi(\boldsymbol{\varepsilon}_e, \boldsymbol{X})$ with the usual uncoupling assumptions, the state law yields

$$\boldsymbol{\sigma} = \rho \frac{\partial \psi}{\partial \boldsymbol{\varepsilon}_e} = \mathbf{K}\boldsymbol{\varepsilon}_e \quad \text{and} \quad \boldsymbol{Y} = \rho \frac{\partial \psi}{\partial \boldsymbol{X}} = \boldsymbol{\Lambda}\boldsymbol{X}, \tag{3.26}$$

where Hooke's tensor $\mathbf{K}$ and the constant, symmetric, and positive definite tensor $\boldsymbol{\Lambda}$ are material characteristics.

The state evolution laws can be written

$$\dot{\mathbf{e}}_p = \mathbf{B}(\mathbf{f}) \quad \text{with} \quad \mathbf{e}_{p|t=0} = 0, \tag{3.27}$$

where $\mathbf{B}$ is a positive operator which is also for most viscoplastic models maximal monotone. Let us introduce now the space $\mathcal{U}_{ad}^{[0,T]}$ of admissible displacement fields $\boldsymbol{u}$ defined over $[0, T] \times \Omega$ and $\mathcal{U}_{ad,0}^{[0,T]}$ the associated vectorial space. The compatibility equation can be written as follows:

Find $\boldsymbol{u} \in \mathcal{U}_{ad}^{[0,T]}$ such that $\forall \boldsymbol{u}^\star \in \mathcal{U}_{ad,0}^{[0,T]}$

$$\int_{[0,T] \times \Omega} \mathrm{Tr}[\boldsymbol{\varepsilon}(\boldsymbol{u})\mathbf{K}\boldsymbol{\varepsilon}(\boldsymbol{u}^\star)] d\Omega dt = \int_{[0,T] \times \Omega} \mathrm{Tr}[\boldsymbol{\varepsilon}_p \mathbf{K}\boldsymbol{\varepsilon}(\boldsymbol{u}^\star)] d\Omega dt$$

$$+ \int_{[0,T] \times \Omega} \boldsymbol{f}_d \cdot \boldsymbol{u}^\star d\Omega dt + \int_{[0,T] \times \partial_2\Omega} \boldsymbol{F}_d \cdot \boldsymbol{u}^\star dS dt. \tag{3.28}$$

It follows that the stress $\boldsymbol{\sigma} = \mathbf{K}(\boldsymbol{\varepsilon}(\boldsymbol{u}) - \boldsymbol{\varepsilon}_p)$ can be written

$$\boldsymbol{\sigma} = \boldsymbol{\Omega}\boldsymbol{\varepsilon}_p + \boldsymbol{r}_d, \tag{3.29}$$

where $\mathbf{\Omega}$ is a linear given operator and $\mathbf{r}_d$ is a prestress depending on the data. Introducing the generalized stress, the admissibility conditions can be written as

$$\mathbf{f} = \mathbf{Q}\mathbf{e}_\mathrm{p} + \mathbf{r}_d \tag{3.30}$$

with

$$\mathbf{Q} = \begin{bmatrix} \mathbf{\Omega} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda} \end{bmatrix} \quad \text{and} \quad \mathbf{r}_d = \begin{bmatrix} \mathbf{r}_d \\ \mathbf{0} \end{bmatrix}, \tag{3.31}$$

where $\mathbf{Q}$ is a linear symmetric positive operator. Finally, the problem to solve is

$$\begin{aligned} &\text{Find } \mathbf{s} = (\dot{\mathbf{e}}_\mathrm{p}, \mathbf{f}) \in \mathbf{S}^{[0,T]} \text{ such that} \\ &\mathbf{f} = \mathbf{Q}\mathbf{e}_\mathrm{p} + \mathbf{r}_d \quad \text{and} \quad \dot{\mathbf{e}}_\mathrm{p} = \mathbf{B}(\mathbf{f}) \quad \text{with} \quad \mathbf{e}_{\mathrm{p}|t=0} = 0. \end{aligned} \tag{3.32}$$

Consequently, one has to solve a first-order differential equation with an initial condition. The operators $\mathbf{Q}$ and $\mathbf{B}$ as well as the right-hand side member $\mathbf{r}_d$ could depend on the parameter $\boldsymbol{\mu}$ belonging to the parameter set $\Sigma_{\boldsymbol{\mu}}$.

3.2.2.2 The solver LATIN for ROM computation

Let us consider ROM computations based in time and space separation. A natural and general idea is to transform the reference problem into a succession of linear global problems over $[0, T] \times \Omega$ which could depend on parameters. Using reduced basis, POD, or PGD, an ROM can be built over $[0, T] \times \Omega$ for each linear problem. The final ROM is then obtained gathering all the previous ROMs. The LATIN method is an iterative strategy which differs from classical incremental or step-by-step techniques in that, at each iteration, it produces an approximation of the full structural response over the whole loading history being considered (see Figure 3.6). In other words, the name LATIN was not chosen very well because the method is essentially nonincremental.
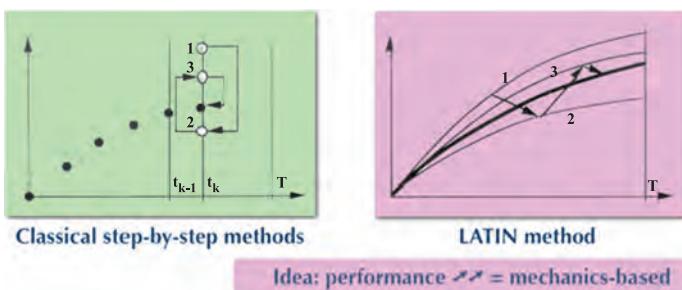

Figure 3.6: The LATIN method and classical step-by-step methods.

The LATIN method, which operates over the time-space domain $[0, T] \times \Omega$, is very convenient for solving the reformulation (3.32) of the reference problem. Its first principle consists in separating the difficulties. Thus, the equations are divided into:

- a set of linear equations which can be global in the space variables: the equilibrium and compatibility equations and the state equations;
- a set of equations which are local in the space variables but can be nonlinear: the state evolution laws.

Local stage at iteration $n + 1$ (see Figure 3.7)
Find $\hat{\mathbf{s}}_{n+1/2} = (\hat{\dot{\mathbf{e}}}_{p,n+1/2}, \hat{\mathbf{f}}_{n+1/2}) \in \mathbf{S}^{[0,T]}$ such that

$$
\begin{aligned}
\hat{\dot{\mathbf{e}}}_{p,n+1/2} &= \mathbf{B}(\hat{\mathbf{f}}_{n+1/2}) \quad \text{with} \quad \hat{\mathbf{e}}_{p,n+1/2} = 0 \text{ at } t = 0, \\
\hat{\dot{\mathbf{e}}}_{p,n+1/2} &- \dot{\mathbf{e}}_{p,n} + \mathbf{H}^{+}(\hat{\mathbf{f}}_{n+1/2} - \mathbf{f}_{n}) = 0.
\end{aligned}
\tag{3.33}
$$

The search direction $\mathbf{H}^{+}$ is a parameter. Practically, one takes a linear positive operator which is local in both time and space variables. It follows that the problem to solve is local in the space variable and then can be split into small independent problems associated to Gauss points. This local stage is very suitable for parallel computing.
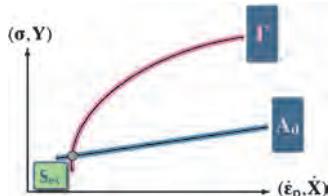
Figure 3.7: The geometric representation associated to the reformulation of the reference problem.
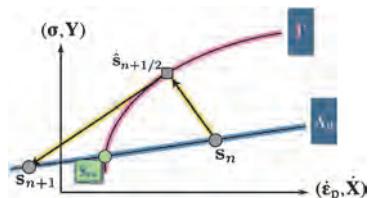
Figure 3.8: Iteration $n + 1$ of the LATIN method over $[0, T] \times \Omega$.

Linear stage at iteration $n + 1$ (see Figure 3.8)
Find $\mathbf{s}_{n+1} = (\dot{\mathbf{e}}_{p,n+1}, \mathbf{f}_{n+1}) \in \mathbf{S}^{[0,T]}$ such that

$$
\begin{aligned}
\mathbf{f}_{n+1} &= \mathbf{Q}\mathbf{e}_{p,n+1} + \mathbf{r}_{d}, \\
\dot{\mathbf{e}}_{p,n+1} &- \hat{\dot{\mathbf{e}}}_{p,n+1/2} - \mathbf{H}^{-}(\mathbf{f}_{n+1} - \hat{\mathbf{f}}_{n+1/2}) = 0 \quad \text{with} \quad \mathbf{e}_{p,n+1} = 0 \text{ at } t = 0.
\end{aligned}
\tag{3.34}
$$

The search direction $\mathbf{H}^{-}$ is a parameter. This is a linear positive operator which is local in both time and space variables. It is associated to the material operator $\mathbf{B}$. One

has to solve a first-order linear differential equation with an initial condition, the operator $\mathbf{Q}$ being nonexplicit. In practice, $\mathbf{H}^-$ is chosen close to the tangent to the manifold $\mathbf{\Gamma}$ at the point $\hat{\mathbf{s}}_{n+1/2} = (\hat{\dot{\mathbf{e}}}_{p,n+1/2}, \hat{\mathbf{f}}_{n+1/2})$. For $\mathbf{H}^+$, one takes $\mathbf{0}$ or $\mathbf{H}^-$. The convergence of the iterative process has been proved in the case of nonsoftening materials and contacts without friction [68]. Precisely, the iterative process converges if:

- the material operator $\mathbf{B}$ is maximal monotone;
- the material operator $\mathbf{\Lambda}$ is positive definite;
- the search directions $\mathbf{H}^-$ and $\mathbf{H}^+$ are positive definite and equal $\mathbf{H}^- = \mathbf{H}^-$.

The distance between two successive approximations gives a good and easily computable error indicator. Let us also note that one often uses an additional relaxation with a coefficient equal to 0.8.

Let us introduce corrections:

$$\Delta\dot{\mathbf{e}}_p = \dot{\mathbf{e}}_{p,n+1} - \dot{\mathbf{e}}_{p,n},$$
$$\Delta\mathbf{f} = \mathbf{f}_{n+1} - \mathbf{f}_n, \tag{3.35}$$

where $\mathbf{s}_{n+1} = (\dot{\mathbf{e}}_{p,n}, \mathbf{f}_n)$ has been computed at iteration n. The problem to solve over $[0, T] \times \Omega$ at iteration $n + 1$ is then:

Find $\Delta\mathbf{s} = (\Delta\dot{\mathbf{e}}_p, \Delta\mathbf{f}) \in \mathbf{S}^{[0,T]}$ such that
$$\Delta\mathbf{f} = \mathbf{Q}\Delta\dot{\mathbf{e}}_p \tag{3.36}$$
$$\Delta\dot{\mathbf{e}}_p - \mathbf{H}^-\Delta\mathbf{f} = \mathbf{R}_d \quad \text{with} \quad \Delta\mathbf{e}_p = 0 \text{ at } t = 0.$$

Problem (3.36) is interpreted as a linear constitutive relation, the operator $\mathbf{H}^-$ being local in both time and space variables and positive definite as the Hooke tensor. Consequently, one introduces the associated CRE which defines the global residual to minimize

$$r(\Delta\mathbf{s}, t) = \frac{1}{2} \int_\Omega [\Delta\dot{\mathbf{e}}_p - \mathbf{H}^-\Delta\mathbf{f} - \mathbf{R}_d](\mathbf{H}^-)^{-1}[\Delta\dot{\mathbf{e}}_p - \mathbf{H}^-\Delta\mathbf{f} - \mathbf{R}_d]d\Omega \tag{3.37}$$

and

$$R(\Delta\mathbf{s}) = \int_{[0,T]} \left(1 - \frac{t}{T}\right) r(\Delta\mathbf{s}, t)dt \tag{3.38}$$

with $\Delta\mathbf{s} = (\Delta\dot{\mathbf{e}}_p, \Delta\mathbf{f}) \in \mathbf{S}^{[0,T]}$. The problem (3.36) becomes

Find $\Delta\mathbf{s} \in \mathbf{S}^{[0,T]}$ miminizing
$$\Delta\mathbf{s} \in \mathbf{S}^{[0,T]} \mapsto R(\mathbf{s}) \in \mathbb{R} \tag{3.39}$$
with the constrains $\Delta\mathbf{f} = \mathbf{Q}\Delta\dot{\mathbf{e}}_p \quad \text{and} \quad \Delta\mathbf{e}_p = 0 \text{ at } t = 0.$

The time residual $r(\Delta \mathbf{s}, t)$ can be used to build ROMs with the reduced basis method. One only prescribes that

$$\Delta \mathbf{e}_{\mathrm{p}} = \sum_{i=1}^{m} \lambda_i(t) \mathbf{g}_i(\boldsymbol{x}) \tag{3.40}$$

with $\lambda_i(0) = 0$ (initial condition), $\mathbf{g}_i \in L^2(\Omega)$, and $\lambda_i(t) \in L^2[0, T]$. It follows, using admissibility conditions, that

$$\Delta \mathbf{f} = \sum_{i=1}^{m} \lambda_i(t) \mathbf{Q} \mathbf{g}_i(\boldsymbol{x}), \tag{3.41}$$

where $\mathbf{Q}\mathbf{g}_i(\boldsymbol{x})$ are computed solving several elasticity problems.

3.2.2.3 Minimization technique

Let us start with

$$\mathbf{s}_n = (\dot{\mathbf{e}}_{\mathrm{p}}^0, \mathbf{f}^0) + \sum_{i=1}^{m} (\dot{\lambda}_i \mathbf{E}_{\mathrm{p}}^i, \lambda_i \mathbf{F}^i). \tag{3.42}$$

The iteration $n + 1$ has two steps.

Step 1: Updating of the PGD time functions – This POD phase relies on the space PGD modes $(\mathbf{E}_{\mathrm{p}}^i, \mathbf{F}^i)$ for which the computation cost is relatively high. New time functions, still noted λ_i, are computed minimizing the residual R with the constraint $\lambda_i(0) = 0$. One gets a small system of differential equations over the time interval with conditions at both ends. The problem can be also solved globally over the time interval $[0, T]$.

Step 2: Addition of a new PGD mode – One computes following a "greedy" algorithm

$$\mathbf{s}_{n+1} = \mathbf{s}_n + (\dot{\lambda} \mathbf{E}_{\mathrm{p}}, \lambda \mathbf{F}) \tag{3.43}$$

with $\lambda(0) = 0$. The additional PGD mode is obtained through the minimization of the residual R, alternatively on the time function λ and on the space function $\mathbf{E}_{\mathrm{p}}$. The initialization of this iterative process is done taking as the first time function guess the root square of the time residual $r(\mathbf{0}, t)$. The minimization with respect to the space variables leads to the resolution of a time-independent spatial problem defined over Ω; that is, a classical finite element problem. The minimization with respect to the time variable leads to a scalar differential equation over $[0, T]$ with conditions at both ends whose resolution is quite inexpensive; the easier way is to solve the global time problem coming from the residual minimization. The iterative process is stopped after few iterations, practically two or three. Let us also note that this second step is canceled if the residual $R(\mathbf{0})$ is relatively small.

3.2.3 Illustration

The previous strategy has been applied in numerous cases. To illustrate the performances, we consider the engineering example presented in [90] and developed in collaboration with SAFRAN. This is a relatively small case with 151,600 degrees of freedom and 60 time steps. The geometry is freely inspired from a blade of the Vulcain engine (Figure 3.9) and the material behavior is the Marquis–Chaboche elastic-viscoplastic law (with kinematic hardening and Norton power law). The problem is not only strongly nonlinear and time-dependent. Two material parameters (power γ and yield stress R_0), as well as the loading amplitude, are not very well known. The range of variation of each parameter was discretized into 10 arbitrarily values, leading to 1,000 different nonlinear time-dependent problems.
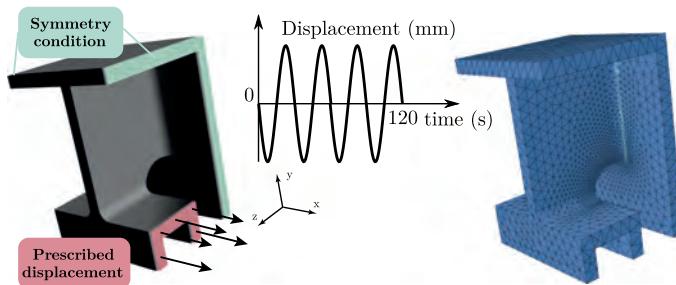
Figure 3.9: Geometry, boundary conditions, and mesh of the blade test case.

For nonlinear time-dependent problems involving especially viscoplasticity and damage, LATIN-PGD has been easily extended to take into account material or loading parameters which are seen as extra-coordinates. Reduced models are built on the parameters/time/space separation [119, 120]. For few parameters, it could be advantageous to describe point-by-point the parameter space using the remarkable property of the LATIN method: The initialization of the iterative process can be any function defined over $[0, T] \times \Sigma_\mu$ [90].

The computations have been carried out on an Intel bi-Xeon processor (total of 12 cores) at 2.8 GHz with 12 GB of RAM. Figure 3.10 gives the virtual chart related to the maximum value of the Von Mises stress. The computational time is about 25 days (estimated time) to complete the 1,000 resolutions with ABAQUS and less than 17 hours with the LATIN-PGD method, leading to a gain of more than 35.

In this example, material parameters (R_0, γ) were assumed to be stochastic, whereas the loading amplitude was assumed to be defined by its interval of variation. The use of the previous virtual chart allows to deal with uncertainties in an inexpensive manner. For example, one can build the interval of variation of the max-
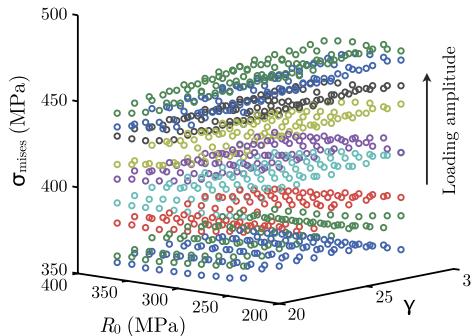
Figure 3.10: Virtual chart giving the maximum Von Mises stress as a function of the parameters.
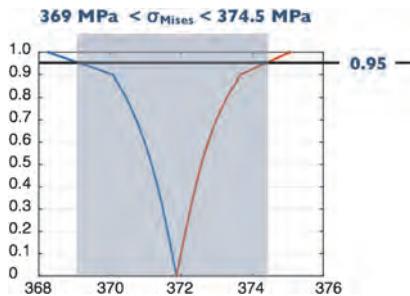
Figure 3.11: Interval of variation of the maximum value of the Von Mises stress with stochastic bounds.

imum value of the Von Mises stress with stochastic bounds, like the result presented in Figure 3.11.

3.2.4 Additional reduction or interpolation

Let us note that LATIN-PGD needs to compute numerous integrals as

$$I = \int_{[0,T]} \int_{\Omega} f(t, M)\mathbf{H}(t, M)g(t, M)d\Omega dt, \qquad (3.44)$$

where $\mathbf{H}$ changes along the iterations, as it can be time-dependent and also nonlinear in terms of the computed solution; f and g are not necessarily represented in the PGD framework. It follows that the computation requires to loop on all the time steps and all the space Gauss points and consequently its cost could be high. For ROM computations where such integral computations are performed online, this problem is a crucial one. This is not the case for LATIN-PGD, where all these calculations are done offline; however, it is always interesting to reduce the computation cost. Several additional reduction or interpolation methods have been proposed to overcome this difficulty and are described in this book, including the empirical interpolation method (EIM), the discrete EIM (DEIM), hyperreduction, etc. Here, we introduce another recent method, named "reference point method" (RPM) [39, Chapter 3], [29].

Let us divide the time interval $I = [0, T]$ being studied into m subintervals $\{I_i\}_{i=1,\dots,m}$ of lengths $\{\Delta t_i\}_{i=1,\dots,m}$ as shown in Figure 3.12. Introducing the centers $\{t_i\}_{i=1,\dots,m}$ of these subintervals, called "reference times," one has $I_i = [t_i - \Delta t_i/2, t_i + \Delta t_i/2]$. In the space domain, let us also introduce m' points $\{M_j\}_{j=1,\dots,m'}$ and partition Ω into $\{\Omega_j\}_{j=1,\dots,m'}$, as shown in Figure 3.12. These points are called "reference points" and the measures of the subdomains are denoted $\{\omega_j\}_{i=j,\dots,m'}$.
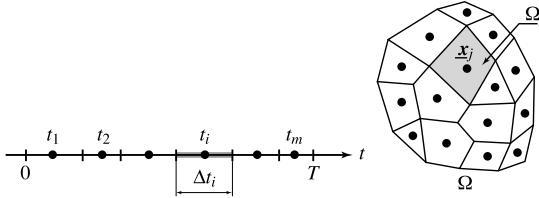
Figure 3.12: The reference times over $[0, T]$ and the reference points over Ω.

Let us consider that one needs 20 modes to describe the solution. One can take double space reference points, i. e., 40, and for the reference times the minimum between 40 and the number of time degrees of freedom. The choice of the reference times and reference points is unrelated to the classical discretizations of the time interval and space domain. Refined time and space discretizations should still be used for the calculation of the various quantities. Here, our purpose is to describe a field f over the time-space domain $[0, T] \times \Omega$ through

$$\hat{a}_i^j(t) = \begin{cases} f(t, M_j) & \text{if } t \in I_i, \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \hat{b}_i^j(M) = \begin{cases} f(t_i, M) & \text{if } M \in \Omega_j, \\ 0 & \text{otherwise}, \end{cases} \quad (3.45)$$

with $i = 1, \dots, m$ and $j = 1, \dots, m'$. The sets $\{(\hat{a}_i^j, \hat{b}_i^j)\}_{i=1,\dots,m}^{j=1,\dots,m'}$ are the generalized components of f. One should note that these quantities verify the following compatibility conditions: for $i = 1, \dots, m$ and $j = 1, \dots, m'$,

$$\hat{a}_i^j(t_i) = \hat{b}_i^j(M_j). \quad (3.46)$$

It could happen that the quantity f is not well represented over the time-space domain. Then, one adds if necessary a PGD description of the residual. The extension to parameter-dependent functions is easy. The great interest of such generalized components is that operations (addition, multiplication, derivation) are greatly facilitated. Then, the main question is: How can one build or rebuild a field from its components? We choose to define function f from its components using only one product per time-space subdomain $I_i \times \Omega_j$:

$$f(t, M) : a_i^j(t) b_i^j(M) \quad \forall (t, M) \in I_i \times \Omega_j. \quad (3.47)$$

The following very simple formula follows from [29]:

$$f(t,M) : a_i(t)b_i(M) = \frac{\sum_{k=1}^{m'} \omega_k \hat{a}_i^k(t)\hat{a}_i^k(t_i)}{\sum_{k=1}^{m'} \omega_k \hat{a}_i^k(t_i)\hat{a}_i^k(t_i)} \hat{b}_i^j(M), \tag{3.48}$$

where the quantities $\hat{a}$, f, and its approximation are equal for the time-space points (t_i, M_j) of the time-space domain. In [29], one will find a performance analysis as well as comparisons with the EIM.

3.2.5 Extensions

LATIN-PGD has been developed for most structural mechanics problems [68] and robust ROM computational methods are available for several important issues:

– **Cyclic viscoplasticity and fatigue for engineering structures** [40, 41, 19]. A PGD with two time scales has been introduced and developed for cyclic loadings. The applications deal with small displacement problems involving (visco)plastic and damageable materials.

– **Large displacement problems with instabilities** [68, 15, 24, 25]. PGD has been extended to large displacement problems for which the classical time/space separation hypothesis does not work or does not work well. The key was a new "material" reformulation of the structure problem; applications deal with large (visco)plastic deformation problems and elastic buckling problems.

– **Concurrent multiscale and multiphysics problems for nonlinear time-dependent problems** [89, 69, 71, 92, 47]. Nonlinear time-dependent problems are considered under the small displacement hypothesis. The key here is a mixed domain decomposition method with two scales over the time-space domain. Classical PGD is used to solve at each iteration the micro-problems. A further path in [43] is to build a PGD-ROM for the computation of the micro-problems over the complete time interval. For multiphysics problems, the idea consists in introducing an abstract interface between physics which plays the same role as the usual interface material.

For all these issues, the LATIN-PGD version described in this chapter should be a paradigm.

3.3 Parametric solutions

This section illustrates how parameters of different natures become coordinates. The problems considered are quite simplistic but the same rationale is considered for solving more complex problems reported at the end of the section. We consider three types

of parameters: (i) parameters related to the model; (ii) parameters related to initial and boundary conditions; and (iii) geometrical parameters defining the space-time domain in which the model is defined.

3.3.1 Model parameters as extra-coordinates

We consider the parametric heat transfer equation

$$\frac{\partial u}{\partial t} - k\Delta u - f = 0, \tag{3.49}$$

with homogeneous initial and boundary conditions. Here $(\mathbf{x}, t, k) \in \Omega_x \times \Omega_t \times \Omega_k$. The scalar conductivity k is here viewed as a new coordinate (called extra-coordinate) defined in the interval Ω_k. Thus, instead of solving the thermal model for different discrete values of the conductivity parameter, we wish to solve only once a more general problem. For that purpose we consider the weighted residual form related to equation (3.49):

$$\int_{\Omega \times \Omega_t \times \Omega_k} u^* \left(\frac{\partial u}{\partial t} - k\Delta u - f \right) d\mathbf{x}\, dt\, dk = 0. \tag{3.50}$$

The PGD solution is sought in the form

$$u(\mathbf{x}, t, k) \approx \sum_{i=1}^{N} X_i(\mathbf{x}) T_i(t) K_i(k), \tag{3.51}$$

constructed using rank-one updates and the alternate directions fixed point algorithm for addressing the nonlinearity, as discussed in the previous section.

3.3.2 Boundary conditions as extra-coordinates

For the sake of simplicity we first consider the steady-state heat equation

$$\nabla \cdot \left(\mathbf{K} \cdot \nabla u(\mathbf{x}) \right) + f(\mathbf{x}) = 0, \tag{3.52}$$

with $\mathbf{x} \in \Omega \subset \mathbb{R}^3$, subjected to the boundary conditions

$$\begin{cases} u(\mathbf{x} \in \Gamma_d) = u_g, \\ (-\mathbf{K} \cdot \nabla u)|_{\mathbf{x} \in \Gamma_n} \cdot \mathbf{n} = \mathbf{q}_g \cdot \mathbf{n} = q_g, \end{cases} \tag{3.53}$$

with $\mathbf{K}$ being the conductivity tensor and $\mathbf{n}$ the outwards unit vector defined in the domain boundary Γ_n, with $\partial\Omega \equiv \Gamma = \Gamma_d \cup \Gamma_n$ and $\Gamma_d \cap \Gamma_n = \emptyset$.

In what follows we consider the simplest scenario that consists of constant Neumann and Dirichlet boundary conditions. More complex and general situations were addressed in [37], where nonconstant boundary and initial conditions were addressed.

3.3.2.1 Neumann boundary condition as extra-coordinate

First, imagine that we are interested in the model solution for values of the heat flux $q_g \in \mathcal{I}_q = [q_g^-, q_g^+]$. We could consider the given heat flux as an extra-coordinate and then solve only once the resulting four-dimensional heat equation for calculating the general parametric solution $u(\mathbf{x}, q_g)$. For this purpose the solution is sought in the separated form

$$u(\mathbf{x}, q_g) \approx \sum_{i=1}^{N} X_i(\mathbf{x}) \cdot \mathcal{Q}_i(q_g). \tag{3.54}$$

In order to enforce the prescribed Dirichlet boundary condition $u(\mathbf{x} \in \Gamma_d) = u_g$, the simplest procedure consists of choosing the first functional couple $X_1(\mathbf{x}) \cdot \mathcal{Q}_1(q_g)$ in order to ensure that $u^1(\mathbf{x} \in \Gamma_d, q_g) = X_1(\mathbf{x} \in \Gamma_d) \cdot \mathcal{Q}_1(q_g) = u_g$. Thus, the remaining terms of the finite sum $X_i(\mathbf{x})$, $i > 1$, will be subjected to homogeneous essential boundary conditions, i. e., $X_i(\mathbf{x} \in \Gamma_d) = 0$.

In order to use the approximation (3.54) we start by considering the weak form related to equation (3.52), which reads as follows: Find $u(\mathbf{x})$ regular enough, verifying $u(\mathbf{x} \in \Gamma_d) = u_g$, such that

$$\int_{\Omega} \nabla u^* \cdot (\mathbf{K} \cdot \nabla u) \, d\mathbf{x} = \int_{\Gamma_n} u^* \, (\mathbf{K} \cdot \nabla u) \cdot \mathbf{n} \, d\mathbf{x} + \int_{\Omega} u^* \, f(\mathbf{x}) \, d\mathbf{x} \tag{3.55}$$

is verified $\forall u^*$, with $u^*(\mathbf{x} \in \Gamma_d) = 0$.

By introducing the Neumann condition (3.53) into (3.55) we obtain

$$\int_{\Omega} \nabla u^* \cdot (\mathbf{K} \cdot \nabla u) \, d\mathbf{x} = - \int_{\Gamma_n} u^* \, q_g \, d\mathbf{x} + \int_{\Omega} u^* \, f(\mathbf{x}) \, d\mathbf{x}, \tag{3.56}$$

which allows constructing the separated form (3.54) using the rank-one updates and the alternate directions fixed point algorithm for addressing the nonlinearity.

3.3.2.2 Dirichlet boundary condition as extra-coordinate

Now we consider the solution of model (3.52) for any value of u_g in (3.53) in a certain interval $\mathcal{I}_u = [u_g^-, u_g^+]$. For this purpose we consider the function $\varphi(\mathbf{x})$ continuous in $\overline{\Omega}$ such that $\Delta \varphi \in L_2(\Omega)$ and $\varphi(\mathbf{x} \in \Gamma_d) = 1$. Thus, we can define the change of variable [54]. We have

$$u(\mathbf{x}) = v(\mathbf{x}) + u_g \, \varphi(\mathbf{x}), \tag{3.57}$$

which allows rewriting equations (3.52) and (3.53) as

$$\nabla \cdot (\mathbf{K} \cdot \nabla v(\mathbf{x})) + u_g \, (\mathbf{K} \cdot \nabla \varphi(\mathbf{x})) + f(\mathbf{x}) = 0, \tag{3.58}$$

which proceeding from its weak form allows again constructing the separated representation of the solution

$$v(\mathbf{x}, u_g) \approx \sum_{i=1}^{N} X_i(\mathbf{x})\mathcal{U}_i(u_g). \tag{3.59}$$

3.3.3 Parametric domains

For the sake of clarity and without loss of generality we address in this section the transient one-dimensional heat equation

$$\frac{\partial u}{\partial t} = k\frac{\partial^2 u}{\partial x^2} + f, \tag{3.60}$$

with $t \in \Omega_t = (0, \Theta]$, $x \in \Omega_x = (0, L)$, constant conductivity k and source term f, and homogeneous initial and boundary conditions, i. e., $u(x = 0, t) = u(x = L, t) = u(x, t = 0) = 0$.

The associated space-time weak form reads

$$\int_{\Omega_x \times \Omega_t} u^* \frac{\partial u}{\partial t}\, dx\, dt = -\int_{\Omega_x \times \Omega_t} k\frac{\partial u^*}{\partial x}\frac{\partial u}{\partial x}\, dx\, dt + \int_{\Omega_x \times \Omega_t} u^* f\, dx\, dt. \tag{3.61}$$

If we are interested in computing the solution $u(x, t)$ in many domains of length $L \in \Omega_L = [L^-, L^+]$ and for many time intervals of length $\Theta \in \Omega_\Theta = [\Theta^-, \Theta^+]$, more than solving the model for many possible choices, it is preferable to compute the parametric solution by considering L and Θ as extra-coordinates. However, equation (3.61) does not involve an explicit dependence on the extra-coordinates L and Θ, both defining the domain of integration. In order to make this dependence explicit, we consider the coordinate transformation

$$\begin{cases} t = \tau\,\Theta, & \tau \in \mathcal{I} = [0, 1], \\ x = \lambda\,L, & \lambda \in \mathcal{I} = [0, 1]. \end{cases} \tag{3.62}$$

In this case the weak form (3.61) reads

$$\int_{\mathcal{I} \times \mathcal{I}} u^* \frac{\partial u}{\partial \tau} L\, d\lambda\, d\tau = -\int_{\mathcal{I} \times \mathcal{I}} k\frac{\partial u^*}{\partial \lambda}\frac{\partial u}{\partial \lambda}\frac{\Theta}{L}\, d\lambda\, d\tau + \int_{\mathcal{I} \times \mathcal{I}} u^* f L \Theta\, d\lambda\, d\tau, \tag{3.63}$$

which allows calculating the parametric solution $u(\tau, \lambda, L, \Theta)$ by considering the separated representation

$$u(\lambda, \tau, L, \Theta) \approx \sum_{i=1}^{N} X_i(\lambda) T_i(\tau) \mathcal{L}_i(L) \mathcal{T}_i(\Theta). \tag{3.64}$$

3.3.4 Related works

This kind of parametric modeling was widely addressed in a panoply of applications, where material parameters [106, 11, 78, 21, 1, 18], initial conditions [55, 57], boundary conditions [49, 50, 91, 58], and parameters defining the geometry [98, 14, 122] were considered as extra-coordinates within the PGD framework. All these parametric solutions were successfully employed for performing real-time simulations (e. g., surgical simulation involving haptic devices addressing contact, cutting, etc.), material homogenization, real-time process optimization, inverse analysis, and simulation-based control. They were also employed in dynamic data-driven application systems. In [119] authors adopted the just referred space-time-parameter separated representation for constructing parametric solutions. Other applications of optimization and dynamic data-driven application systems are found in [42, 83, 109].

For the treatment of the nonlinearities involved in the works just referred, the separated representation constructors were combined with numerous nonlinear solvers ranging from the most standard ones (fixed point, Newton, etc.) to less standard approaches based on LATIN, the asymptotic numerical method (e. g., [91, 82], among many others), or the DEIM [37].

In the context of stochastic modeling, PGD was introduced in [93] for the uncertainty quantification and propagation. The interpretation of the separated representation constructor as a generalized eigenproblem allowed to define dedicated algorithms inspired from solution techniques for classical eigenproblems [94]. In this context deterministic and stochastic contributions were separated, making PGD a promising alternative to traditional methods for uncertainty propagation, as discussed in [95]. PGD was also extended to stochastic nonlinear problems in [96]. More recently, the PGD was successfully applied to the solution of high-dimensional stochastic parametric problems, with the introduction of suitable hierarchical tensor representations and associated algorithms [97].

In engineering problems, the classical separated variable representation is used for up to around 20 parameters. To go further, a parameter-multiscale PGD has been devised to overcome this major limitation of the classical ROM computational techniques [101]. It is based on Saint-Venant's principle, which highlights two different levels of parametric influence.

3.4 Space separated representations

Plates and shells are very common in nature and thus they inspired engineers to use both of them from the very beginning of structural mechanics. Nowadays, plate and shell parts are massively present in most engineering applications.

This type of structural elements involves homogeneous and heterogeneous materials, isotropic and anisotropic, linear and nonlinear. The appropriate design of such parts consists not only in the structural analysis of the parts for accommodating the design loads, but also in the analysis of the associated manufacturing processes because many properties of the final parts depend on the formation process itself (e. g., flow-induced microstructures). Thus, fine analyses concern both the structural parts and their associated formation processes.

In general the whole design requires the solution of some mathematical models governing the evolution of the quantities of interest. These models consist of a set of partial differential equations combining general balance equations (mass, energy, and momentum) and some specific constitutive equations depending on the considered physics, the last involving different material parameters. These complex equations (in general nonlinear and strongly coupled) must be solved in the domain of interest.

When addressing plate or shell geometries the domains in which the mathematical models must be solved become degenerated because one of its characteristic dimensions (the thickness in the present case) is much lower than the other characteristic dimensions. We will understand the consequences of such degeneracy later. When analytical solutions are neither available nor possible because of the geometrical or behavioral complexities, the solution must be calculated by invoking any of the available numerical techniques (finite elements, finite differences, finite volumes, methods of particles, etc.).

In the numerical framework the solution is only obtained in a discrete number of points, usually called nodes, distributed in the domain. From the solution at those points, it can be interpolated at any other point in the domain. In general, regular nodal distributions are preferred because they offer better accuracies. In the case of degenerated plate or shell domains one could expect that if the solution evolves significantly in the thickness direction, a large enough number of nodes must be distributed along the thickness direction to ensure the accurate representation of the field evolution in that direction. In that case, a regular nodal distribution in the whole domain will imply the use of an extremely large number of nodes, with the consequent impact on the numerical solution efficiency.

When simple behaviors and domains were considered, plate and shell theories were developed in the structural mechanics framework allowing, through the introduction of some hypotheses, reducing the three-dimensional complexity to a two-dimensional one related to the problem now formulated by considering the in-plane coordinates.

In the case of fluid flows this dimensionality reduction is known as lubrication theory and it allows efficient solutions of fluid flows taking place in plate or shell geometries for many type of fluids, linear (Newtonian) and nonlinear. The interest of this type of flows is not only due to the fact that it is involved in the manufacturing processes of plate and shell parts, but also due to the fact that many tests for characterizing material behaviors involve it.

However, as soon as richer physics are included in the models and the considered geometries differ from those ensuring the validity of the different reduction hypotheses, simplified simulations are compromised and they fail in their predictions.

In these circumstances the reduction from the three-dimensional model to a two-dimensional simplified one is not obvious, and three-dimensional simulations appear many times as the only valid route for addressing such models, which despite the fact that they are defined in degenerated geometries (plates or shells), they seem to require a fully three-dimensional solution. However, in order to integrate such a calculation (fully three-dimensional and implying an impressive number of degrees of freedom) in usual design procedures, a new efficient (fast and accurate) solution procedure is needed. The in-plane-out-of-plane separated representations represent a valuable route able to compute the different unknown three-dimensional fields without the necessity of introducing any hypothesis. The most outstanding advantage is that three-dimensional solutions can be obtained with a computational cost characteristic of standard two-dimensional solutions, as previously described. In what follows we formulate different physics within such a separated representation framework.

3.4.1 Heat transfer in laminates

In this section we illustrate the construction of the PGD of a generic model defined in a plate domain $\Xi = \Omega \times \mathcal{I}$ with $\Omega \subset \mathbb{R}^2$ and $\mathcal{I} = [0, H] \subset \mathbb{R}$. For the sake of simplicity we consider the model related to the steady-state heat conduction equation

$$\nabla \cdot (\mathbf{K} \cdot \nabla u) = 0, \tag{3.65}$$

in a plate geometry that contains P plies in the plate thickness. Each ply is characterized by its conductivity tensor $\mathbf{K}_i(x, y)$ which is assumed constant through the ply thickness. Moreover, without any loss of generality, we assume the same thickness h for the different plies constituting the laminate. Thus, we can define a characteristic function representing the position of each ply $i = 1, \ldots, P$:

$$\chi_i(z) = \begin{cases} 1 & z_i \leqslant z \leqslant z_{i+1}, \\ 0 & \text{otherwise}, \end{cases} \tag{3.66}$$

where $z_i = (i - 1)h$ defines the location of the i-th ply in the laminate thickness. Now, the laminate conductivity can be given in the following separated form:

$$\mathbf{K}(x, y, z) = \sum_{i=1}^{i=P} \mathbf{K}_i(\mathbf{x}) \cdot \chi_i(z), \tag{3.67}$$

where $\mathbf{x}$ denotes the in-plane coordinates, i. e., $\mathbf{x} = (x, y) \in \Omega$.

The weak form of equation (3.65), with appropriate boundary conditions, reads

$$\int_\Xi \nabla u^* \cdot (\mathbf{K} \cdot \nabla u)\, d\Xi = 0, \tag{3.68}$$

with the test function u^* defined in an appropriate functional space. The solution $u(x, y, z)$ is then searched under the separated form:

$$u(\mathbf{x}, z) \approx \sum_{j=1}^{j=N} X_j(\mathbf{x}) Z_j(z). \tag{3.69}$$

3.4.2 Three-dimensional Resin Transfer Moulding

We now summarize the application of PGD to the modeling of resin transfer moulding processes. We consider the flow within a porous medium in a plate domain $\Xi = \Omega \times \mathcal{I}$ with $\Omega \subset \mathbb{R}^2$ and $\mathcal{I} = [0, H] \subset \mathbb{R}$. The governing equation is obtained by combining Darcy's law, which relates the fluid velocity to the pressure gradient,

$$\mathbf{v} = -\mathbf{K} \cdot \nabla p, \tag{3.70}$$

and the incompressibility constraint,

$$\nabla \cdot \mathbf{v} = 0. \tag{3.71}$$

Introduction of equation (3.70) into equation (3.71) yields a single equation for the pressure field:

$$\nabla \cdot (\mathbf{K} \cdot \nabla p) = 0. \tag{3.72}$$

The mould contains a laminate preform composed of P different anisotropic plies of thickness h, each one characterized by a permeability tensor $\mathbf{K}_i(x, y)$ that is assumed constant through the ply thickness. We define a characteristic function

$$\chi_i(z) = \begin{cases} 1 & z_i \leqslant z \leqslant z_{i+1}, \\ 0 & \text{otherwise,} \end{cases} \tag{3.73}$$

where $z_i = (i-1)h$ is the location of the i-th ply in the plate thickness. The laminate permeability is thus given in separated form as follows:

$$\mathbf{K}(x, y, z) = \sum_{i=1}^{P} \mathbf{K}_i(\mathbf{x}) \cdot \chi_i(z), \tag{3.74}$$

where $\mathbf{x}$ denotes the in-plane coordinates, i. e., $\mathbf{x} = (x, y) \in \Omega$.

The weak form of equation (3.72) reads

$$\int_\Xi \nabla p^* \cdot (\mathbf{K} \cdot \nabla p) \, d\Xi = 0, \tag{3.75}$$

for all test functions p^* selected in an appropriate functional space. Dirichlet boundary conditions are imposed for the pressure at the inlet and outlet of the flow domain $p(\mathbf{x} \in \Gamma_D) = p_g(\mathbf{x})$, while zero flux (i. e., no flow) $\nabla p \cdot \mathbf{n} = 0$ is imposed elsewhere ($\mathbf{n}$ being the unit outwards vector defined on the domain boundary) as a weak boundary condition. We seek an approximate solution $p(x, y, z)$ in the PGD form

$$p(\mathbf{x}, z) \approx \sum_{j=1}^{N} X_j(\mathbf{x}) \cdot Z_j(z), \tag{3.76}$$

which is constructed by using the standard procedure previously discussed.

3.4.3 The elastic problem defined in plate domains

We proposed in [21] and original in-plane-out-of-plane decomposition of the three-dimensional elastic solution in a plate geometry. The elastic problem was defined in a plate domain $\Xi = \Omega \times \mathcal{I}$ with $(x, y) \in \Omega$, $\Omega \subset \mathbb{R}^2$ and $z \in \mathcal{I}$, $\mathcal{I} = [0, H] \subset \mathbb{R}$, H being the plate thickness. The separated representation of the displacement field $\mathbf{u} = (u_1, u_2, u_3)$ reads

$$\mathbf{u}(x, y, z) = \begin{pmatrix} u_1(x, y, z) \\ u_2(x, y, z) \\ u_3(x, y, z) \end{pmatrix} \approx \sum_{i=1}^{N} \begin{pmatrix} P_1^i(x, y) \cdot T_1^i(z) \\ P_2^i(x, y) \cdot T_2^i(z) \\ P_3^i(x, y) \cdot T_3^i(z) \end{pmatrix}, \tag{3.77}$$

where P_k^i, $k = 1, 2, 3$, are functions of the in-plane coordinates (x, y), whereas T_k^i, $k = 1, 2, 3$, are functions involving the thickness coordinate z. In [21] we compared the first modes of such separated representations with the kinematic hypotheses usually considered in plate theories.

Expression (3.77) can be written in a more compact form by using the Hadamard (component-to-component) product:

$$\mathbf{u}(x, y, z) \approx \sum_{i=1}^{N} \mathbf{P}^i(x, y) \circ \mathbf{T}^i(z), \tag{3.78}$$

where vectors $\mathbf{P}^i$ and $\mathbf{T}^i$ contain functions P_k^i and T_k^i respectively.

Let us consider a linear elasticity problem on a plate domain $\Xi = \Omega \times \mathcal{I}$. The weak form using the so-called Voigt notation reads

$$\int_\Xi \boldsymbol{\epsilon}(\mathbf{u}^*)^T \cdot \mathbf{K} \cdot \boldsymbol{\epsilon}(\mathbf{u}) \, d\mathbf{x} = \int_\Xi \mathbf{u}^* \cdot \mathbf{f}_d \, d\mathbf{x} + \int_{\Gamma_N} \mathbf{u}^* \cdot \mathbf{F}_d \, d\mathbf{x}, \quad \forall \mathbf{u}^*, \tag{3.79}$$

where **K** is the generalized 6×6 Hooke tensor, $\mathbf{f}_d$ represents the volumetric body forces, and $\mathbf{F}_d$ represents the traction applied on the boundary Γ_N. The separation of variables introduced in equation (3.77) yields a separated representation for the derivatives of the displacement components u_i, $i = 1, 2, 3$, and from it the separated representation of the strain tensor $\boldsymbol{\epsilon}$:

$$\boldsymbol{\epsilon}(\mathbf{u}(x,y,z)) \approx \sum_{k=1}^{N} \begin{pmatrix} \frac{\partial P_1^k}{\partial x} \cdot T_1^k \\ \frac{\partial P_2^k}{\partial y} \cdot T_2^k \\ P_3^k \cdot \frac{\partial T_3^k}{\partial z} \\ \frac{\partial P_1^k}{\partial y} \cdot T_1^k + \frac{\partial P_2^k}{\partial x} \cdot T_2^k \\ \frac{\partial P_3^k}{\partial x} \cdot T_3^k + P_1^k \cdot \frac{\partial T_1^k}{\partial z} \\ \frac{\partial P_3^k}{\partial y} \cdot T_3^k + P_2^k \cdot \frac{\partial T_2^k}{\partial z} \end{pmatrix}, \tag{3.80}$$

which introduced into the weak form allows computing the separated form (3.78).

3.4.4 Three-dimensional elastic problem in a shell domain

In this section we consider a shell domain Ω^S, assumed with constant thickness and described from a reference surface $\mathbf{X}$. In what follows, that reference surface will be identified to the shell middle surface parameterized by the coordinates ξ, η, that is, $\mathbf{X}(\xi, \eta)$, where

$$\mathbf{X}(\xi, \eta) = \begin{pmatrix} X_1(\xi, \eta) \\ X_2(\xi, \eta) \\ X_3(\xi, \eta) \end{pmatrix}. \tag{3.81}$$

With $\mathbf{n}$ being the unit vector normal to the middle surface, the shell domain Ω^S can be parameterized from

$$\mathbf{x}(\xi, \eta, \zeta) = \mathbf{X}(\xi, \eta) + \zeta \cdot \mathbf{n}. \tag{3.82}$$

The geometrical transformation $(\xi, \eta, \zeta) \to (x_1, x_2, x_3)$, at its inverse, can be easily obtained and expressed into a separated form.

The weak form of the elastic problem defined in the shell domain Ω^S using again the Voigt notation reads

$$\int_{\Omega^S} \boldsymbol{\epsilon}(\mathbf{u}^*)^T \cdot \mathbf{K} \cdot \boldsymbol{\epsilon}(\mathbf{u}) \, d\mathbf{x} = \int_{\Omega^S} \mathbf{u}^* \cdot \mathbf{f}_d \, d\mathbf{x} + \int_{\Gamma_N^S} \mathbf{u}^* \cdot \mathbf{F}_d \, d\mathbf{x}. \tag{3.83}$$

Now we are considering the coordinate transformation introduced in the previous section mapping $\mathbf{x} \in \Omega^S$ into $(\xi, \eta, \zeta) \in \Xi = \Omega \times \mathcal{I}$, with $(\xi, \eta) \in \Omega \subset \mathbb{R}^2$ and $\zeta \in \mathcal{I} \subset \mathbb{R}$.

The geometric transformation requires to transform the differential operator as well as the different volume and surface elements, from which the standard procedure applies for computing the separated form of all the kinematics and static variables.

3.4.5 Squeeze flow in composite laminates

The in-plane-out-of-plane separated representation allows the solution of full three-dimensional flow models defined in plate geometries with a computational complexity characteristic of two-dimensional simulations. In the present case the three-dimensional velocity field reads

$$\mathbf{v}(\mathbf{x}, z) \approx \sum_{i=1}^{N} \mathbf{P}_i(\mathbf{x}) \circ \mathbf{T}_i(z). \tag{3.84}$$

The Stokes flow model is defined in $\Xi = \Omega \times \mathcal{I}$, $\Omega \subset \mathbb{R}^2$ and $\mathcal{I} \subset \mathbb{R}$, and for an incompressible fluid, in the absence of inertia and mass terms it reduces to

$$\begin{cases} \nabla \cdot \boldsymbol{\sigma} = \mathbf{0}, \\ \boldsymbol{\sigma} = -p\mathbf{I} + 2\eta\mathbf{D}, \\ \nabla \cdot \mathbf{v} = 0, \end{cases} \tag{3.85}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\mathbf{I}$ is the unit tensor, η is the fluid viscosity, p is the pressure (Lagrange multiplier associated with the incompressibility constraint), and the rate of strain tensor $\mathbf{D}$ is defined as

$$\mathbf{D} = \frac{\nabla\mathbf{v} + (\nabla\mathbf{v})^T}{2}. \tag{3.86}$$

When considering a laminate composed of P layers in which each layer involves a linear and isotropic viscous fluid of viscosity η_i, the extended Stokes flow problem in its weak form involves the dependence of the viscosity along the thickness direction.

If H is the total laminate thickness, and assuming for the sake of simplicity and without loss of generality that all plies have the same thickness h, it results that $h = \frac{H}{P}$. Now, from the characteristic function of each ply $\chi_i(z)$, $i = 1, \ldots, P$,

$$\chi_i(z) = \begin{cases} 1 & \text{if } (i-1)h \leqslant z < ih, \\ 0 & \text{elsewehere,} \end{cases} \tag{3.87}$$

the viscosity reads

$$\eta(\mathbf{x}, z) = \sum_{i=1}^{P} \eta_i \cdot \chi_i(z), \tag{3.88}$$

where it is assumed, again without loss of generality, that the viscosity does not evolve in the plane, i. e., $\eta_i(\mathbf{x}) = \eta_i$.

The Stokes model can be easily extended to power law fluids where the extra-stress tensor reads

$$\mathbf{T} = 2KD_{\text{eq}}^{n-1}\mathbf{D}, \tag{3.89}$$

with D_{eq} being the equivalent strain rate, as well as to more complex constitutive models as the ones involved in composite manufacturing processes.

3.4.6 Electromagnetic models in composite laminates

Conventional processing methods for producing polymer composite parts usually involve the application of heat to the material by convection or conductive heating through elements, which depend on surface heat transfer. Microwave (MW) technology relies on volumetric heating, that means thermal energy is transferred through electromagnetic fields to materials that can absorb it at specific frequencies. Volumetric heating enables better process temperature control and less overall energy use, which can result in shorter processing cycles. Furthermore, comparable mechanical properties are shown between parts made with the MW technology and parts made with a traditional curing system. These virtues of the MW technology have attracted interest in developing the method and adopting it for the production of thermoset as well as thermoplastic composite materials.

The double-curl formulation is derived from the Maxwell equations in the frequency space, which in the absence of current density in the laminate reads

$$\nabla \times \left(\frac{1}{\mu} \nabla \times \mathbf{E} \right) - \omega^2 \epsilon \mathbf{E} = 0, \quad \text{in } \Omega \subset \mathbb{R}^3 \tag{3.90}$$

with the complex permittivity ϵ given by

$$\epsilon = \epsilon_r - i \frac{\sigma}{\omega}, \tag{3.91}$$

and where μ, ϵ_r, and σ represent the usual magnetic permeability, the electric permittivity, and the conductivity, respectively.

The previous equation is complemented with adequate boundary conditions. Without loss of generality we are assuming in what follows Dirichlet boundary conditions in the whole domain boundary $\partial\Omega$,

$$\mathbf{n} \times \mathbf{E} = \mathbf{E}_g^t, \quad \text{in } \partial\Omega, \tag{3.92}$$

where $\mathbf{n}$ refers to the unit outwards vector defined on the domain boundary. In the previous expressions $\mathbf{E}_g^t$ is the prescribed electric field (assumed known) on the domain boundary, tangent to the boundary as equation (3.92) expresses.

The weighted residual weak form is obtained by multiplying (3.90) by the test function $\mathbf{E}^*$ (in fact by its conjugate, $\overline{\mathbf{E}}^*$, to define properly scalar products being the complex-valued electric field, i. e., $\mathbf{E} = \mathbf{E}_r + i\mathbf{E}_i$), and then introducing a stabilization to enforce the Gauss law,

$$\int_\Omega \frac{1}{\mu} (\nabla \times \mathbf{E}) \cdot (\nabla \times \overline{\mathbf{E}}^*) \, d\mathbf{x} - \omega^2 \int_\Omega \epsilon \mathbf{E} \cdot \overline{\mathbf{E}}^* \, d\mathbf{x}$$

$$+ \int_\Omega \frac{\tau}{\overline{\epsilon}\epsilon\mu} (\nabla \cdot (\epsilon\mathbf{E})) \left(\nabla \cdot (\overline{\epsilon}\overline{\mathbf{E}}^*) \right) d\mathbf{x}$$

$$-\int_{\Omega}\frac{\tau}{\overline{\epsilon}\epsilon\mu}\left(\nabla\cdot(\epsilon\mathbf{E})\right)\left(\nabla\cdot(\overline{\epsilon}\overline{\mathbf{E}}^{*})\right)d\mathbf{x}\; -\int_{\partial\Omega}\frac{\tau}{\overline{\epsilon}\epsilon\mu}\left(\nabla\cdot(\epsilon\mathbf{E})\left(\mathbf{n}\cdot(\overline{\epsilon}\overline{\mathbf{E}}^{*})\right)\right)d\mathbf{x}=0,\qquad(3.93)$$

where τ is the regularization coefficient.

To ensure a high enough resolution of the electric field along the component thickness to represent the multilayered structure, we consider an in-plane-out-of-plane separated representation

$$\mathbf{E}(x,y,z)\approx\sum_{i=1}^{N}\mathbf{P}_{i}(x,y)\circ\mathbf{T}_{i}(z),$$

where "∘" refers to the Hadamard product, and use the standard rank-one update constructor.

3.5 Conclusions

This chapter revisited the state of the art and the recent developments in the use of PGD for addressing engineering problems. In particular it addressed the pioneering works considering space-time separated representations, which were then extended for solving multidimensional models encountered in kinetic theory descriptions of complex fluids, quantum chemistry, etc. They were also considered for describing and solving stochastic models and any kind of parameterized partial differential equations whose solutions result in a sort of virtual chart or computational vademecum. These parametric solutions have been successfully employed with multiple purposes: simulation, optimization, inverse analysis, uncertainty propagation, and control, all of them under the stringent constraint of real-time feedbacks. Finally separated representations were extended for separating space and efficiently addressing the solution of problems in degenerated domains, as for example problems defined in plates, shells, laminates, etc.

Bibliography

[1]   M. S. Aghighi, A. Ammar, C. Metivier, and F. Chinesta, Parametric solution of the Rayleigh-Bénard convection model by Using the PGD: Application to nanofluids, *International Journal of Numerical Methods for Heat & Fluid Flow*, **25** (6) (2015), 1252–1281.
[2]   I. Alfaro, D. Gonzalez, S. Zlotnik, P. Diez, E. Cueto, and F. Chinesta, An error estimator for real-time simulators based on model order reduction, *Advanced Modeling and Simulation in Engineering Sciences*, **2** (1) (2015), 30.
[3]   P. E. Allier, L. Chamoin, and P. Ladevèze, Proper Generalized Decomposition computational methods on a benchmark problem: introducing a new strategy based on Constitutive Relation Error minimization, *Advanced Modeling and Simulation in Engineering Sciences*, **2** (2015), 17.

[4] P. E. Allier, L. Chamoin, and P. Ladevèze, Towards simplified and optimized a posteriori error estimation using PGD reduced models, *International Journal for Numerical Methods in Engineering*, **113** (6) (2018), 967–998.

[5] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids, *Journal of Non-Newtonian Fluid Mechanics*, **139** (2006), 153–176.

[6] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids: Part II: Transient simulation using space-time separated representations, *Journal of Non-Newtonian Fluid Mechanics*, **144** (2-3) (2007), 98–121.

[7] A. Ammar, F. Chinesta, and P. Joyot, The nanometric and micrometric scales of the structure and mechanics of materials revisited: An introduction to the challenges of fully deterministic numerical descriptions, *International Journal of Multiscale Computational Engineering*, **6** (3) (2008), 191–213.

[8] A. Ammar, M. Normandin, F. Daim, D. Gonzalez, E. Cueto, and F. Chinesta, Non-incremental strategies based on separated representations: Applications in computational rheology, *Communications in Mathematical Sciences*, **8** (3) (2010), 671–695.

[9] A. Ammar, F. Chinesta, P. Diez, and A. Huerta, An error estimator for separated representations of highly multidimensional models, *Computer Methods in Applied Mechanics and Engineering*, **199** (25-28) (2010), 1872–1880.

[10] A. Ammar, F. Chinesta, and A. Falco, On the convergence of a greedy rank-one update algorithm for a class of linear systems, *Archives of Computational Methods in Engineering*, **17** (4) (2010), 473–486.

[11] A. Ammar, M. Normandin, and F. Chinesta, Solving parametric complex fluids models in rheometric flows, *Journal of Non-Newtonian Fluid Mechanics*, **165** (2010), 1588–1601.

[12] A. Ammar, F. Chinesta, and E. Cueto, Coupling finite elements and Proper Generalized Decompositions, *International Journal of Multiscale Computational Engineering*, **9** (1) (2011), 17–33.

[13] A. Ammar, E. Cueto, and F. Chinesta, Reduction of the Chemical Master Equation for gene regulatory networks using Proper Generalized Decompositions, *International Journal for Numerical Methods in Biomedical Engineering*, **28** (9) (2012), 960–973.

[14] A. Ammar, A. Huerta, F. Chinesta, E. Cueto, and A. Leygue, Parametric solutions involving geometry: a step towards efficient shape optimization, *Computer Methods in Applied Mechanics and Engineering*, **268C** (2014), 178–193.

[15] X. Aubard, P. A. Boucard, P. Ladevèze, and S. Michel, Modeling and simulation of damage in elastomer structures at high strains, *Computers & Structures*, **80** (2002), 2289–2298.

[16] A. Barbarulo, P. Ladevèze, H. Riou, and L. Kovalevsky, Proper Generalized Decomposition applied to linear acoustic: A new tool for broad band calculation, *Journal of Sound and Vibration*, **333** (11) (2014), 2422–2431.

[17] R. Becker and R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, *Acta Numerica*, **10** (5-8) (2001), 1–120. A. Isereles (ed.), Cambridge University Press.

[18] M. Beringhier and M. Gigliotti, A novel methodology for the rapid identification of the water diffusion coefficients of composite materials, *Composites. Part A, Applied Science and Manufacturing*, **68** (2015), 212–218.

[19] M. Bhattacharyya, D. Néron, A. Fau, U. Nackenhorst, and P. Ladevèze, A multi-temporal scale model reduction approach for the computation of fatigue damage, *Computer Methods in Applied Mechanics and Engineering*, **340** (2018), 630–656.

[20] M. Billaud-Friess, A. Nouy, and O. Zahm, A tensor approximation method based on ideal minimal residual formulations for the solution of high-dimensional problems, *ESAIM: Mathematical Modelling and Numerical Analysis*, **48** (6) (2014), 1777–1806.

[21] B. Bognet, A. Leygue, F. Chinesta, A. Poitou, and F. Bordeu, Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity, *Computer Methods in Applied Mechanics and Engineering*, **201** (2012), 1–12.

[22] B. Bognet, A. Leygue, and F. Chinesta, Separated representations of 3D elastic solutions in shell geometries. *Advanced Modelling and Simulation in Engineering Sciences*, **1** (2014), 4, http://www.amses-journal.com/content/1/1/4.

[23] F. Bordeu, Ch. Ghnatios, D. Boulze, B. Carles, D. Sireude, A. Leygue, and F. Chinesta, Parametric 3D elastic solutions of beams involved in frame structures, *Advances in Aircraft and Spacecraft Science*, **2/3** (2015), 233–248.

[24] P. A. Boucard, P. Ladevèze, M. Poss, and P. Rougée, A nonincremental approach for large displacement problems, *Computers & Structures*, **64** (1997), 499–508.

[25] P. Bussy, P. Rougée, and P. Vauchez, The large time increment method for numerical simulation of metal forming processes, in *NUMETA*, Elsevier, pp. 102–109, 1990.

[26] D. Canales, A. Leygue, F. Chinesta, I. Alfaro, D. Gonzalez, E. Cueto, E. Feulvarch, and J. M. Bergheau, In-plane-out-of-plane separated representations of updated-Lagrangian descriptions of thermomechanical models defined in plate domains, *Comptes Rendus de l'Academie de Sciences*, **344** (4-5) (2016), 225–235.

[27] E. Cancès, V. Ehrlacher, and T. Lelièvre, Convergence of a greedy algorithm for high-dimensional convex nonlinear problems, *Mathematical Models and Methods in Applied Sciences*, **21** (12) (2011), 2433–2467.

[28] E. Cancès, V. Ehrlacher, and T. Lelievre, Greedy algorithms for high-dimensional eigenvalue problems, *Constructive Approximation*, **40** (3) (2014), 387–423.

[29] M. Capaldo, P. A. Guidault, D. Néron, and P. Ladevèze, The reference point method, a "hyperreduction" technique: Application to PGD-based nonlinear model reduction, *Computational Methods in Applied Mechanical Engineering*, **322** (2017), 483–514.

[30] L. Chamoin and P. Ladevèze, Robust control of PGD-based numerical simulations, *European Journal of Computational Mechanics*, **21** (3-6) (2012), 195–207.

[31] L. Chamoin and P. Diez, *Verifying Calculations – Forty Years on. SpringerBriefs in Applied Sciences and Technology*, Springer International Publishing, 2016.

[32] L. Chamoin, P. E. Allier, and B. Marchand, Synergies between the constitutive relation error concept and PGD model reduction for simplified V&V procedures, *Advanced Modeling and Simulation in Engineering Sciences*, **3** (2016), 18.

[33] L. Chamoin, F. Pled, P. E. Allier, and P. Ladevèze, A posteriori error estimation and adaptive strategy for PGD model reduction applied to parametrized linear parabolic problems, *Computer Methods in Applied Mechanics and Engineering*, **327** (2017), 118–146.

[34] F. Chinesta, A. Ammar, A. Falco, and M. Laso, On the reduction of stochastic kinetic theory models of complex fluids, *Modelling and Simulation in Materials Science and Engineering*, **15** (2007), 639–652.

[35] F. Chinesta, A. Ammar, A. Leygue, and R. Keunings, An overview of the Proper Generalized Decomposition with applications in computational rheology, *Journal of Non-Newtonian Fluid Mechanics*, **166** (2011), 578–592.

[36] F. Chinesta, R. Keunings, and A. Leygue, *The Proper Generalized Decomposition for Advanced Numerical Simulations. A Primer. Springerbriefs*, Springer, 2013.

[37] F. Chinesta, A. Leygue, F. Bordeu, J. V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar, and A. Huerta, Parametric PGD based computational vademecum for efficient design, optimization and control, *Archives of Computational Methods in Engineering*, **20/1** (2013), 31–59.

[38]   F. Chinesta, A. Leygue, B. Bognet, Ch. Ghnatios, F. Poulhaon, F. Bordeu, A. Barasinski, A. Poitou, S. Chatel, and S. Maison-Le-Poec, First steps towards an advanced simulation of composites manufacturing by automated tape placement, *International Journal of Material Forming*, **7** (1) (2014), 81–92.

[39]   F. Chinesta and P. Ladevèze (eds.), *Separated Representations and PGD-Based Model Reduction. CISM International Centre for Mechanical Sciences*, Springer-Verlag, Wien, 2014.

[40]   J. Y. Cognard and P. Ladevèze, A large time increment approach for cyclic viscoplasticity, *International Journal of Plasticity*, **9** (1993), 141–157.

[41]   J. Y. Cognard, P. Ladeveze, and P. Talbot, A large time increment approach for thermo-mechanical problems, *Advances in Engineering Software*, **30** (9-11) (1999), 583–593.

[42]   A. Courard, D. Neron, P. Ladeveze, and L. Ballere, Integration of pgd-virtual charts into an engineering design process, *Computational Mechanics*, **57** (4) (2016), 637–651.

[43]   M. Cremonesi, D. Néron, P. A. Guidault, and P. Ladevèze, A PGD-based homogenization technique for the resolution of nonlinear multiscale problems, *Computer Methods in Applied Mechanics and Engineering*, **267** (2013), 275–292.

[44]   A. Dumon, C. Allery, and A. Ammar, Proper general decomposition (PGD) for the resolution of Navier-Stokes equations, *Journal of Computational Physics*, **230** (4) (2011), 1387–1407.

[45]   A. Dumon, C. Allery, and A. Ammar, Proper Generalized Decomposition method for incompressible Navier-Stokes equations with a spectral discretization, *Applied Mathematics and Computation*, **219** (15) (2013), 8145–8162.

[46]   A. Dumon, C. Allery, and A. Ammar, Simulation of heat and mass transport in a square lid-driven cavity with Proper Generalized Decomposition, *Numerical Heat Transfer. Part B, Fundamentals*, **63** (1) (2013b), 18–43.

[47]   D. Dureisseix and D. Neron, A multiscale computational approach with field transfer dedicated to coupled problems, *International Journal of Multiscale Computational Engineering*, **6** (3) (2008), 233–250.

[48]   L. Gallimard, P. Vidal, and O. Polit, Coupling finite element and reliability analysis through proper generalized decomposition model reduction, *International Journal for Numerical Methods in Engineering*, **95** (13) (2013), 1079–1093.

[49]   Ch. Ghnatios, F. Chinesta, E. Cueto, A. Leygue, P. Breitkopf, and P. Villon, Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: Application to pultrusion, *Composites. Part A*, **42** (2011), 1169–1178.

[50]   Ch. Ghnatios, F. Masson, A. Huerta, E. Cueto, A. Leygue, and F. Chinesta, Proper Generalized Decomposition based dynamic data-driven control of thermal processes, *Computer Methods in Applied Mechanics and Engineering*, **213** (2012), 29–41.

[51]   Ch. Ghnatios, G. Xu, M. Visonneau, A. Leygue, F. Chinesta, and A. Cimetiere, On the space separated representation when addressing the solution of PDE in complex domains, *Discrete and Continuous Dynamical Systems*, **9** (2) (2016), 475–500.

[52]   Ch. Ghnatios, F. Chinesta, and Ch. Binetruy, The squeeze flow of composite laminates, *International Journal of Material Forming*, **8** (2015), 73–83.

[53]   E. Giner, B. Bognet, J. J. Rodenas, A. Leygue, J. Fuenmayor, and F. Chinesta, The Proper Generalized Decomposition (PGD) as a numerical procedure to solve 3D cracked plates in linear elastic fracture mechanics, *International Journal of Solids and Structures*, **50** (10) (2013), 1710–1720.

[54]   D. Gonzalez, A. Ammar, F. Chinesta, and E. Cueto, Recent advances in the use of separated representations, *International Journal for Numerical Methods in Engineering*, **81** (5) (2010), 637–659.

[55]   D. Gonzalez, F. Masson, F. Poulhaon, A. Leygue, E. Cueto, and F. Chinesta, Proper Generalized Decomposition based dynamic data-driven inverse identification, *Mathematics and Computers in Simulation*, **82** (9) (2012), 1677–1695.

[56] D. Gonzalez, E. Cueto, F. Chinesta, P. Diez, and A. Huerta, SUPG-based stabilization of Proper Generalized Decompositions for high-dimensional advection-diffusion equations, *International Journal for Numerical Methods in Engineering*, **94** (13) (2013), 1216–1232.

[57] D. Gonzalez, E. Cueto, and F. Chinesta, Real-time direct integration of reduced solid dynamics equations, *International Journal for Numerical Methods in Engineering*, **99** (9) (2014), 633–653.

[58] D. Gonzalez, I. Alfaro, C. Quesada, E. Cueto, and F. Chinesta, Computational vademecums for the real-time simulation of haptic collision between nonlinear solids, *Computer Methods in Applied Mechanics and Engineering*, **283** (2015), 210–223.

[59] F. El Halabi, D. Gonzalez, A. Chico, and M. Doblare, FE2 multiscale in linear elasticity based on parametrized microscale models using proper generalized decomposition, *Computer Methods in Applied Mechanics and Engineering*, **257** (2013), 183–202.

[60] A. Ern, A. F. Stephansen, and M. Vohralik, Guaranteed and robust discontinuous Galerkin a posteriori error estimates for convection-diffusion-reaction problems, *Journal of Computational and Applied Mathematics*, **234** (1) (2010), 114–130.

[61] N. K. M. Faber, R. Bro, and P. K. Hopke, Recent developments in candecomp/parafac algorithms: a critical review, *Chemometrics and Intelligent Laboratory Systems*, **65** (1) (2003), 119–137.

[62] A. Falco and A. Nouy, A proper generalized decomposition for the solution of elliptic problems in abstract form by using a functional Eckart-Young approach, *Journal of Mathematical Analysis and Applications*, **376** (2011), 469–480.

[63] A. Falco and A. Nouy, Proper generalized decomposition for nonlinear convex problems in tensor Banach spaces, *Numerische Mathematik*, **121** (3) (2012), 503–530.

[64] J. A. Hernández, J. Oliver, A. E. Huespe, M. A. Caicedo, and J. C. Cante, High-performance model reduction techniques in computational multiscale homogenization, *Computer Methods in Applied Mechanics and Engineering*, **276** (2014), 149–189.

[65] K. Kergrene, L. Chamoin, S. Prudhomme, and M. Laforest, On a goal-oriented version of the Proper Generalized Decomposition, *Journal of Scientific Computing*, **81** (2019), 92–111.

[66] P. Ladevèze, On a family of algorithms for structural mechanics (in french), *Comptes Rendus de l'Academie des Sciences*, **300** (2) (1985), 41–44.

[67] P. Ladevèze, The large time increment method for the analyse of structures with nonlinear constitutive relation described by internal variables, *Comptes Rendus de l'Académie des Sciences Paris*, **309** (II) (1989), 1095–1099.

[68] P. Ladevèze, *Nonlinear Computational Structural Mechanics – New Approaches and Non-Incremental Methods of Calculation*, Springer Verlag, 1999.

[69] P. Ladevèze and A. Nouy, On a multiscale computational strategy with time and space homogenization for structural mechanics, *Computer Methods in Applied Mechanics and Engineering*, **192** (2003), 3061–3087.

[70] P. Ladevèze and J. P. Pelle, *Mastering Calculations in Linear and Nonlinear Mechanics*, Springer, New York, 2004.

[71] P. Ladevèze, D. Néron, and P. Gosselet, On a mixed and multiscale domain decomposition method, *Computer Methods in Applied Mechanics and Engineering*, **196** (2007), 1526–1540.

[72] P. Ladevèze, Strict upper error bounds on computed outputs of interest in computational structural mechanics, *Computational Mechanics*, **42** (2) (2008), 271–286.

[73] P. Ladevèze and L. Chamoin, On the verification of model reduction methods based on the proper generalized decomposition, *Computer Methods in Applied Mechanics and Engineering*, **200** (23) (2011), 2032–2047.

[74] P. Ladevèze and L. Chamoin, Toward Guaranteed PGD-reduced Models, in G. Zavarise and D. P. Boso (eds.), CIMNE, pp. 143–154, 2012.

[75] P. Ladevèze, F. Pled, and L. Chamoin, New bounding techniques for goal-oriented error estimation applied to linear problems, *International Journal for Numerical Methods in Engineering*, **93** (13) (2013), 1345–1380.

[76] P. Ladevèze and L. Chamoin, The constitutive relation error method: a general verification tool, in *Verifying Calculations – Forty Years On*. Springer International Publishing, pp. 59–94, 2016.

[77] P. Ladevèze, On reduced models in nonlinear solid mechanics, *European Journal of Mechanics. A, Solids*, **60** (2016), 227–237.

[78] H. Lamari, A. Ammar, P. Cartraud, G. Legrain, F. Jacquemin, and F. Chinesta, Routes for efficient computational homogenization of non-linear materials using the proper generalized decomposition, *Archives of Computational Methods in Engineering*, **17** (4) (2010), 373–391.

[79] H. Lamari, A. Ammar, A. Leygue, and F. Chinesta, On the solution of the multidimensional Langer's equation by using the Proper Generalized Decomposition Method for modeling phase transitions, *Modelling and Simulation in Materials Science and Engineering*, **20** (1) (2012), 015007.

[80] C. Le Bris, T. Lelièvre, and Y. Maday, Results and questions on a nonlinear approximation approach for solving high-dimensional partial differential equations, *Constructive Approximation*, **30** (2009), 621–651.

[81] G. M. Leonenko and T. N. Phillips, On the solution of the Fokker-Planck equation using a high-order reduced basis approximation, *Computer Methods in Applied Mechanics and Engineering*, **199** (1-4) (2009), 158–168.

[82] A. Leygue, F. Chinesta, M. Beringhier, T. L. Nguyen, J. C. Grandidier, F. Pasavento, and B. Schrefler, Towards a framework for non-linear thermal models in shell domains, *International Journal of Numerical Methods for Heat & Fluid Flow*, **23** (1) (2013), 55–73.

[83] B. Marchand, L. Chamoin, and Ch. Rey, Real-time updating of structural mechanics models using Kalman filtering, modified constitutive relation error and proper generalized decomposition, *International Journal for Numerical Methods in Engineering*, **107** (9) (2016), 786–810.

[84] S. Metoui, E. Pruliere, A. Ammar, F. Dau, and I. Iordanoff, The proper generalized decomposition for the simulation of delamination using cohesive zone model, *International Journal for Numerical Methods in Engineering*, **99** (13) (2014), 1000–1022.

[85] J. P. Moitinho de Almeida, A basis for bounding the errors of proper generalized decomposition solutions in solid mechanics, *International Journal for Numerical Methods in Engineering*, **94** (10) (2013), 961–984.

[86] B. Mokdad, E. Pruliere, A. Ammar, and F. Chinesta, On the simulation of kinetic theory models of complex fluids using the Fokker-Planck approach, *Applied Rheology*, **17** (2) (2007), 26494-1–26494-14.

[87] B. Mokdad, A. Ammar, M. Normandin, F. Chinesta, and J. R. Clermont, A fully deterministic micro-macro simulation of complex flows involving reversible network fluid models, *Mathematics and Computers in Simulation*, **80** (2010), 1936–1961.

[88] M. Nazeer, F. Bordeu, A. Leygue, and F. Chinesta, Arlequin based PGD domain decomposition, *Computational Mechanics*, **54** (5) (2014), 1175–1190.

[89] D. Néron and P. Ladevèze, Proper Generalized Decomposition for multiscale and multiphysics problems, *Archives of Computational Methods in Engineering*, **17** (4) (2010), 351–372.

[90] D. Néron, P. A. Boucard, and N. Relun, Time-space PGD for the rapid solution of 3D nonlinear parametrized problems in the many-query context, *International Journal for Numerical Methods in Engineering*, **103** (4) (2015), 275–292.

[91] S. Niroomandi, D. Gonzalez, I. Alfaro, F. Bordeu, A. Leygue, E. Cueto, and F. Chinesta, Real time simulation of biological soft tissues: A PGD approach, *International Journal for Numerical Methods in Biomedical Engineering*, **29** (5) (2013), 586–600.

[92] A. Nouy and P. Ladevèze, Multiscale computational strategy with time and space homogenization: a radial type approximation technique for solving micro problems, *International Journal of Multiscale Computational Engineering*, **170** (2) (2004), 557–574.

[93] A. Nouy, A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations, *Computer Methods in Applied Mechanics and Engineering*, **196** (45-48) (2007), 4521–4537.

[94] A. Nouy, Generalized spectral decomposition method for solving stochastic finite element equations: invariant subspace problem and dedicated algorithms, *Computer Methods in Applied Mechanics and Engineering*, **197** (2008), 4718–4736.

[95] A. Nouy and O. Le Maitre, Generalized spectral decomposition method for stochastic non linear problems, *Journal of Computational Physics*, **228** (1) (2009), 202–235.

[96] A. Nouy, Recent developments in spectral stochastic methods for the numerical solution of stochastic partial differential equations, *Archives of Computational Methods in Engineering*, **16** (3) (2009), 251–285.

[97] A. Nouy, Proper Generalized Decompositions and separated representations for the numerical solution of high dimensional stochastic problems, *Archives of Computational Methods in Engineering – State of the Art Reviews*, **17** (2010), 403–434.

[98] A. Nouy, Fictitious domain method and separated representations for the solution of boundary value problems on uncertain parameterized domains, *Computer Methods in Applied Mechanics and Engineering*, **200** (45-46) (2011), 3066–3082.

[99] A. Nouy, *Low-Rank Tensor Methods for Model Order Reduction*, pp. 1-26, Springer International Publishing, Cham, 2016.

[100] J. T. Oden, J. Fish, C. Johnson, A. Laub, D. Srolovitz, T. Belytschko, T. J. R. Hughes, D. Keyes, and L. Petzold, Simulation-based engineering science. Report of the NSF SBES Panel to the NSF Engineering Advisory Committee, 2006.

[101] C. Paillet, D. Neron, and P. Ladeveze, A door to model reduction in high-dimensional parameter space, *Comptes Rendus. Mathématique*, **346** (7) (2018), 524–531.

[102] M. Paraschivoiu, J. Peraire, and A. T. Patera, A posteriori finite element bounds for linear-functional outputs of elliptic partial differential equations, *Computer Methods in Applied Mechanics and Engineering*, **150** (1-4) (1997), 289–312.

[103] N. Pares, P. Diez, and A. Huerta, Exact bounds for linear outputs of the advection-diffusion-reaction equation using flux-free error estimates, *SIAM Journal on Scientific Computing*, **31** (4) (2009), 3064–3089.

[104] S. Prudhomme and J. T. Oden, On goal-oriented error estimation for elliptic problems: application to the control of pointwise errors, *Computer Methods in Applied Mechanics and Engineering*, **176** (1) (1999), 313–331.

[105] E. Pruliere, A. Ammar, N. El Kissi, and F. Chinesta, Recirculating flows involving short fiber suspensions: Numerical difficulties and efficient advanced micro-macro solvers, *Archives of Computational Methods in Engineering, State of the Art Reviews*, **16** (2009), 1–30.

[106] E. Pruliere, F. Chinesta, and A. Ammar, On the deterministic solution of multidimensional parametric models by using the Proper Generalized Decomposition, *Mathematics and Computers in Simulation*, **81** (2010), 791–810.

[107] E. Pruliere, 3D simulation of laminated shell structures using the Proper Generalized Decomposition, *Composite Structures*, **117** (2014), 373–381.

[108] A. Radermacher and S. Reese, Pod-based model reduction with empirical interpolation applied to nonlinear elasticity, *International Journal for Numerical Methods in Engineering*, **107** (6) (2016), 477–495.

[109] P. Rubio, F. Louf, and L. Chamoin, Fast model updating coupling Bayesian inference and PGD model reduction, *Computational Mechanics*, **62** (6) (2018), 1485–1509.

[110] D. Ryckelynck, A priori hyperreduction method: an adaptive approach, *Journal of Computational Physics*, **202** (2005), 346–366.

[111] D. Ryckelynck, Hyper-reduction of mechanical models involving internal variables, *International Journal for Numerical Methods in Engineering*, **77** (1) (2009), 75–89.

[112] H. Tertrais, R. Ibanez, A. Barasinski, Ch. Ghnatios, and F. Chinesta, On the Proper Generalized Decomposition applied to microwave processes involving multilayered components, *Mathematics and Computers in Simulation*, **156** (2019), 347–363.

[113] L. Chamoin and H. P. Thai, Certified real-time shape optimization using isogeometric analysis, PGD model reduction, and a posteriori error estimation, *International Journal for Numerical Methods in Engineering*, **119** (3) (2019), 151–176.

[114] P. Vidal, L. Gallimard, and O. Polit, Composite beam finite element based on the Proper Generalized Decomposition, *Computers & Structures*, **102** (2012), 76–86.

[115] P. Vidal, L. Gallimard, and O. Polit, Proper Generalized Decomposition and layer-wise approach for the modeling of composite plate structures, *International Journal of Solids and Structures*, **50** (14-15) (2013), 2239–2250.

[116] P. Vidal, L. Gallimard, and O. Polit, Explicit solutions for the modeling of laminated composite plates with arbitrary stacking sequences, *Composites. Part B, Engineering*, **60** (2014), 697–706.

[117] P. Vidal, L. Gallimard, and O. Polit, Shell finite element based on the Proper Generalized Decomposition for the modeling of cylindrical composite structures, *Computers & Structures*, **132** (2014), 1–11.

[118] P. Vidal, L. Gallimard, and O. Polit, Assessment of variable separation for finite element modeling of free edge effect for composite plates, *Composite Structures*, **123** (2015), 19–29.

[119] M. Vitse, D. Néron, and P. A. Boucard, Virtual charts of solutions for parametrized nonlinear equations, *Computational Mechanics*, **54** (2014), 1529–1539.

[120] M. Vitse, D. Néron, and P. A. Boucard, Dealing with a nonlinear material behavior and its variability through PGD models: application to reinforced concrete structures, *Finite Elements in Analysis and Design*, **153** (2019), 22–37.

[121] J. Yvonnet, E. Monteiro, and Q. C. He, Computational homogenization method and reduced database model for hyperelastic heterogeneous structures, *International Journal of Multiscale Computational Engineering*, **11** (3) (2013), 201–225.

[122] S. Zlotnik, P. Diez, D. Gonzalez, E. Cueto, and A. Huerta, Effect of the separated approximation of input data in the accuracy of the resulting PGD solution, *Advanced Modeling and Simulation in Engineering Sciences*, **2** (1) (2015), 1–14.

Yvon Maday and Anthony T. Patera

4 Reduced basis methods

Abstract: In this chapter we describe the reduced basis (RB) method for parameterized partial differential equations (PDEs). We first describe the motivation for RB methods in the many-query and real-time contexts and the associated offline-online computational paradigm. We next introduce the framework for parameterized PDEs and the associated theoretical rationale for reduction. We then turn to projection techniques: formulation, a priori and a posteriori error estimation, and offline-online computational strategies. We next discuss techniques for identification of optimal approximation spaces, in particular the weak greedy approach. We emphasize linear elliptic PDEs, but we also consider nonlinear elliptic PDEs as well as linear parabolic PDEs.

Keywords: weak greedy sampling, empirical interpolation method, Galerkin projection, a posteriori error estimation, offline-online procedure

MSC 2010: 65M60, 65N30

4.1 Motivation

Parameterized partial differential equations (PDEs) are important in many scientific and engineering applications. The parameters typically characterize the spatial domain, the boundary conditions and initial conditions and sources, and the coefficients associated with the underlying constitutive relations. In general the solution of our (say, elliptic) PDE shall be a parameterized field: For given parameter value $\boldsymbol{\mu} \equiv (\mu_1, \dots, \mu_p) \in \mathcal{P}$, $u(\boldsymbol{\mu}) \in V$; here $\mathcal{P} \in \mathbb{R}^p$ is a compact parameter domain, and V is the appropriate function space associated with our PDE. In what follows, we assume that V is a Hilbert space. In the forward context we prescribe $\boldsymbol{\mu}$ to deduce $u(\boldsymbol{\mu})$; in the inverse context, such as parameter estimation, classification, and optimization, we deduce $\boldsymbol{\mu}$ from functionals applied to $u(\boldsymbol{\mu})$.

We introduce the parametric manifold $\mathcal{M} \equiv \{u(\boldsymbol{\mu}) \mid \boldsymbol{\mu} \in \mathcal{P}\}$. The premise for parameterized model order reduction is well established [28, 1]: For the approximation of the solution $u(\boldsymbol{\mu})$ for many $\boldsymbol{\mu} \in \mathcal{P}$, we need not necessarily consider a finite element (FE) approximation space $V_h \subset V$ which can well represent *any* function in V; we need only

Yvon Maday, Sorbonne Université and Université de Paris, CNRS, Laboratoire Jacques-Louis Lions (LJLL) and Institut Universitaire de France, F-75005 Paris, France
Anthony T. Patera, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, USA

consider a reduced basis (RB) approximation space $V_N \subset V_h$ which can well represent any function in $\mathcal{M} \subset V$ in the sense that dist$(\mathcal{M}, V_N)$ is small. We may thus anticipate that the dimension of V_N, N, will be much smaller than the dimension of V_h, N_h, with attendant reductions in computational cost: A query to the RB approximation, $\boldsymbol{\mu} \in \mathcal{P} \mapsto u_N(\boldsymbol{\mu}) \in V_N$, will be much less expensive than a query to the FE approximation, $\boldsymbol{\mu} \in \mathcal{P} \mapsto u_h(\boldsymbol{\mu}) \in V_h$. In actual fact, not all manifolds are "reducible," where reducible here is defined as a sufficiently rapid decrease of the Kolmogorov N-width [30, 23]; we shall provide some a priori and a posteriori tests to confirm the latter. Furthermore, given a reducible manifold, the identification of a good RB space V_N – such that dist$(\mathcal{M}, V_N)$ also decreases rapidly with N – requires considerable computational effort, and in particular many appeals to the FE approximation.

Parameterized model order reduction thus proceeds in two stages: in the offline stage, given our parameter domain $\mathcal{P}$ and parameterized PDE, we construct a sequence of parameter-independent spaces $\{V_N\}_{N=1,\dots,N_{\max}}$; in the online stage, for given N, we query the RB approximation, $\boldsymbol{\mu} \in \mathcal{P} \mapsto u_N(\boldsymbol{\mu}) \in V_N$. This offline-online paradigm is computationally relevant if (i) the offline effort to construct V_N can be justified either by a real-time or many-query context, and (ii) the online effort to evaluate $\boldsymbol{\mu} \mapsto u_N(\boldsymbol{\mu})$ is indeed much less than the online effort to evaluate $\boldsymbol{\mu} \mapsto u_h(\boldsymbol{\mu})$. We elaborate on (i) and (ii).

(i)   In the real-time context, we simply choose to "write off" the offline effort given the stated premium on rapid response in the online (deployed) stage. In the many-query context, we explicitly amortize the offline effort over many online RB queries.

(ii) We will typically require not only $N_{\max} \ll N_h$ but also special structural properties of the parameterized PDE and associated solution procedures: This special structure is often realized through an empirical interpolation method (EIM) [5] which introduces additional "variational-crime" errors; we shall denote the resulting FE and RB approximations by $\tilde{u}_h(\boldsymbol{\mu}) \in V_h$ and $\tilde{u}_N(\boldsymbol{\mu}) \in V_N$, respectively.

Note from an applications perspective we proceed not from model order reduction to context, but from context to model order reduction: A real-time or many-query application justifies an offline-online computational strategy which in turn can be realized through (among other strategies) model order reduction.

We briefly discuss the choice of the parameter dimensionality, p, and parameter domain, $\mathcal{P}$. We may consider as a first proposal all parameters of possible interest, p_0 large, and a parameter domain $\mathcal{P}_0$ which contains all values of $\boldsymbol{\mu}_0$ for which our PDE is well-posed (in the sense to be described below). However, in the context of model order reduction, the offline and online computational cost will depend on the number of parameters and also the extent and "shape" of the parameter domain, and hence we must typically accept $p < p_0$ and hence $\mathcal{P} \subset \mathcal{P}_0$: the parameterization and parameter domain must be chosen to anticipate the parameter values of ultimate interest in the online applications to be considered.

We provide a roadmap of the chapter. In general, we emphasize general (second-order) linear elliptic PDEs, but we also consider a nonlinear elliptic PDE as well as general linear parabolic PDEs; extension to nonlinear parabolic PDEs is then immediate. In Section 4.2 we formulate our parameterized PDEs and summarize the associated theoretical foundation for dimension reduction: conditions, or at least guidelines, under which a manifold is reducible. In Section 4.3 we develop the projection method, in fact simple Galerkin projection, by which we determine $\tilde{u}_N(\boldsymbol{\mu}) \in V_N$; we also provide a priori error estimates and a posteriori error estimators, and we describe the associated offline-online computational procedures. In Section 4.4 we describe methods for construction of the RB approximation space, V_N, with emphasis on the weak greedy procedure: The weak greedy procedure efficiently identifies a parameter sample $S_{N_{\max}} \equiv \{\boldsymbol{\mu}^j\}_{j=1,\ldots,N_{\max}}$ (from a rich train set $\Xi_{\mathrm{RB}} \subset \mathcal{P}$) to form hierarchical RB spaces $V_N \equiv \mathrm{span}\{\tilde{u}_h(\boldsymbol{\mu}^j)\}_{j=1,\ldots,N}$, $1 \le N \le N_{\max}$, which well represent the parametric manifold $\mathcal{M}$; the weak greedy procedure, and in particular the rate of decrease of $\mathrm{dist}(\mathcal{M}, V_N)$ with N, is a constructive test of reducibility.

The prerequisite for this chapter is experience in the formulation, elementary theory, and implementation of FE methods for PDEs, as well as some exposure to associated functional analysis. The intended audience is graduate students and professionals who wish to consider RB methods in their research or design efforts. The chapter emphasizes (i) the conditions and hypotheses under which RB methods may prove fruitful, (ii) the fundamental ingredients and procedures which must be incorporated in any RB formulation, and (iii) the underlying error analysis, a priori and a posteriori, which informs successful RB practice. We focus on the "inputs" – related to the particular PDE, parameter domain, and context – which must be provided by the prospective user, and on methods which can be generally and easily implemented given a standard FE foundation.

Finally, for readers who seek further details, a broader range of alternative techniques, more general classes of problems, and deeper coverage of both theory and implementation, we recommend two recent research monographs on RB methods [21, 32]. We hope our chapter here can serve as a portal to further study.

4.2 Parameterized PDEs

In Section 4.2.1 and Section 4.2.2 we consider linear elliptic PDEs. In Section 4.2.3 we consider parabolic PDEs as well as nonlinear elliptic PDEs.

4.2.1 Weak form

4.2.1.1 Formulation

We introduce a spatial domain $\Omega \subset \mathbb{R}^d$ with boundary $\partial\Omega$; we denote a point in Ω as $x \equiv (x_1, \ldots, x_d)$. We then define the space V as $V \equiv \{v \in H^1(\Omega) \,|\, v|_{\Gamma_D} = 0\}$ for Γ_D a nonempty portion of the boundary $\partial\Omega$; we denote the inner product and induced norm associated to V as $(\cdot, \cdot)_V$ and $\|\cdot\|_V$, respectively. Unless otherwise noted, we shall take for our inner product

$$(w, v)_V \equiv \int_\Omega \nabla w \cdot \nabla v + c_{L^2}\, w\, v, \tag{4.1}$$

for c_{L^2} a nonnegative real number. We further introduce the dual space to V, V', of linear functionals continuous with respect to $\|\cdot\|_V$; we equip V' with the usual dual norm,

$$\|g\|_{V'} = \sup_{v \in V} \frac{|g(v)|}{\|v\|_V}, \quad \forall g \in V'. \tag{4.2}$$

We also define the Riesz representation of any g in V', $\mathcal{R}g \in V$, by

$$(\mathcal{R}g, v)_V = g(v), \quad \forall v \in V. \tag{4.3}$$

Finally, we recall that

$$\|g\|_{V'} = \|\mathcal{R}g\|_V, \quad \forall g \in V', \tag{4.4}$$

which follows directly from (4.2), (4.3), and the Cauchy–Schwarz inequality.

We now introduce the parameterized linear forms $\boldsymbol{\mu} \in \mathcal{P} \mapsto f(\cdot; \boldsymbol{\mu})$ and $\boldsymbol{\mu} \in \mathcal{P} \mapsto \ell(\cdot; \boldsymbol{\mu})$. In fact, for simplicity of exposition, we shall assume that

$$f(v; \boldsymbol{\mu}) = \int_\Omega f_\Omega(\cdot; \boldsymbol{\mu})\, v + \int_{\Gamma_{N,R}} f_{\Gamma_{N,R}}(\cdot; \boldsymbol{\mu})\, v, \; \ell(v; \boldsymbol{\mu}) = \int_\Omega \ell_\Omega(\cdot; \boldsymbol{\mu})\, v + \int_{\Gamma_{N,R}} \ell_{\Gamma_{N,R}}(\cdot; \boldsymbol{\mu})\, v, \tag{4.5}$$

where $f_\Omega(\boldsymbol{\mu}) \in L^2(\Omega)$, $\ell_\Omega(\boldsymbol{\mu}) \in L^2(\Omega)$, $f_{\Gamma_{N,R}}(\boldsymbol{\mu}) \in L^2(\Gamma_{N,R})$, and $\ell_{\Gamma_{N,R}}(\boldsymbol{\mu}) \in L^2(\Gamma_{N,R})$. Here $\Gamma_{N,R} \equiv \partial\Omega \setminus \bar{\Gamma}_D$ is the portion of the boundary on which non-Dirichlet (Neumann or Robin) boundary conditions are applied. It follows from our assumptions that $f(\cdot; \boldsymbol{\mu})$ and $\ell(\cdot; \boldsymbol{\mu})$ are continuous for all $\boldsymbol{\mu} \in \mathcal{P}$.

We further introduce the parameterized bilinear form $\boldsymbol{\mu} \in \mathcal{P} \mapsto a(\cdot, \cdot; \boldsymbol{\mu}) : V \times V \to \mathbb{R}$. In general, a may take the form

$$a(w, v; \boldsymbol{\mu}) = \sum_{i,j=0}^{d} \int_\Omega \Upsilon_{ij}(\cdot; \boldsymbol{\mu}) \frac{\partial w}{\partial x_i} \frac{\partial v}{\partial x_j}, \tag{4.6}$$

for $\Upsilon_{ij} \in L^\infty(\mathcal{P}; L^\infty(\Omega))$ and (for convenience) $\partial./\partial x_0 \equiv \mathrm{Id}$ (the identity operator); note that a can also include a contribution such as $\int_{\Gamma_{N,R}} \overline{\Upsilon}(\cdot;\boldsymbol{\mu})\, wv$ with $\overline{\Upsilon} \in L^\infty(\mathcal{P}; L^\infty(\Gamma_{N,R}))$. We shall need several constants to characterize our bilinear form a: For any $\boldsymbol{\mu} \in \mathcal{P}$, the coercivity constant, $\alpha(\boldsymbol{\mu})$, continuity constant, $\gamma(\boldsymbol{\mu})$, and inf-sup constant, $\beta(\boldsymbol{\mu})$, are given respectively by

$$\alpha(\boldsymbol{\mu}) \equiv \inf_{w \in V} \frac{|a(w,w;\boldsymbol{\mu})|}{(w,w)_V}, \quad \gamma(\boldsymbol{\mu}) \equiv \sup_{w \in V} \sup_{v \in V} \frac{|a(w,v;\boldsymbol{\mu})|}{\|w\|_V \|v\|_V}, \quad \beta(\boldsymbol{\mu}) \equiv \inf_{w \in V} \sup_{v \in V} \frac{|a(w,v;\boldsymbol{\mu})|}{\|w\|_V \|v\|_V}; \tag{4.7}$$

we denote by $\alpha = \min_{\boldsymbol{\mu} \in \mathcal{P}} \alpha(\boldsymbol{\mu})$, $\gamma = \max_{\boldsymbol{\mu} \in \mathcal{P}} \gamma(\boldsymbol{\mu})$, and $\beta = \min_{\boldsymbol{\mu} \in \mathcal{P}} \beta(\boldsymbol{\mu})$ the corresponding worst case quantities over the entire parameter domain. For economy of presentation we define

$$c^{\mathrm{s}}(\boldsymbol{\mu}) \equiv \begin{cases} \alpha(\boldsymbol{\mu}) & \text{for the coercive case,} \\ \beta(\boldsymbol{\mu}) & \text{for the noncoercive case;} \end{cases} \tag{4.8}$$

we may also write $c^{\mathrm{s}}(\boldsymbol{\mu}) = \max(\alpha(\boldsymbol{\mu}), \beta(\boldsymbol{\mu}))$, however for computational purposes we prefer the more explicit definition (4.8). We can then state our hypotheses on a: in the coercive case, α is positive and γ is finite, and in the noncoercive case, β is positive and γ is finite; more succinctly, we require c^{s} positive and γ finite.

We now define the weak form of our parameterized PDE: Given $\boldsymbol{\mu} \in \mathcal{P}$, find a (or the) field $u(\boldsymbol{\mu}) \in V$ such that

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \quad \forall v \in V, \tag{4.9}$$

and evaluate the scalar output $s(\boldsymbol{\mu}) \in \mathbb{R}$ as $s(\boldsymbol{\mu}) = \ell(u(\boldsymbol{\mu}); \boldsymbol{\mu})$.[1] (We implicitly assume that all inhomogeneous essential boundary conditions $u_D(\boldsymbol{\mu}) \in H^{1/2}(\Gamma_D)$ are lifted and hence implicitly incorporated in $f(\cdot; \boldsymbol{\mu})$.) It follows from our hypotheses on a and f and the Lions–Lax–Milgram–Babuška theorem that (4.9) admits a unique solution for all $\boldsymbol{\mu} \in \mathcal{P}$; furthermore, from our hypothesis on ℓ, $s(\boldsymbol{\mu})$ is finite for all $\boldsymbol{\mu} \in \mathcal{P}$. (In fact, for the noncoercive case, we require a third condition on a, typically satisfied in our context.)

We provide a simple illustration, which we denote Example 1.0. We consider a connected open domain $\Omega \subset \mathbb{R}^{d=3}$ and further assume that Ω is decomposed as the union of two nonoverlapping open subdomains, $\Omega_{(1)}$ and $\Omega_{(2)}$: $\overline{\Omega} = \overline{\Omega}_{(1)} \cup \overline{\Omega}_{(2)}$ and $\Omega_{(1)} \cap \Omega_{(2)} = \emptyset$. We set $p = 3$ and introduce parameter $\boldsymbol{\mu} \equiv (\mu_1, \mu_2, \mu_3) \in \mathcal{P} \subset \{v \in \mathbb{R}^3 \mid v_1 > 0, v_2 \geq 0\}$. We then define

$$a(w, v; \boldsymbol{\mu}) = \int_{\Omega_{(1)}} \mu_1 \nabla w \cdot \nabla v + \int_{\Omega_{(2)}} \nabla w \cdot \nabla v + \int_{\Omega} \mu_2 wv, \quad \forall w, v \in V^2, \tag{4.10}$$

---

[1] We may associate to our output functional a dual problem and corresponding adjoint; the latter can serve (for example) in the development of improved RB output approximation and error estimation [25]. In the interest of space, we consider only simpler primal-only approximation.

as well as

$$f(v; \mu) = \int_\Omega (1 + \mu_3 x_1) v \quad \text{and} \quad \ell(v; \mu) = \int_{\Gamma_{N,R}} v. \tag{4.11}$$

We can readily demonstrate that a is continuous and coercive, and furthermore symmetric, and that f and ℓ are continuous. (We can also include a convection term: For a divergence-free convection velocity which is furthermore outward on $\Gamma_{N,R}$, the bilinear form a remains continuous and coercive but will no longer be symmetric.) In the case in which Ω is polyhedral we would expect $u(\mu) \in H^{1+\sigma_{\text{regularity}}}$ for $\sigma_{\text{regularity}} > 0$; the latter would limit the convergence rate of the FE approximation, but need not limit the convergence rate of the RB approximation.

We comment briefly on the treatment of parameter-dependent geometry. In our exposition we shall consider only the case in which Ω is independent of μ. In actual practice, we may treat problems in parameter-dependent geometry, $\mu \in \mathcal{P} \mapsto \Omega_{\text{orig}}(\mu)$. In that case we introduce $V_{\text{orig}}(\mu) = H^1(\Omega_{\text{orig}}(\mu))$ and bilinear and linear form $\mu \in \mathcal{P} \mapsto a_{\text{orig}}(\cdot, \cdot; \mu) : V_{\text{orig}}(\mu) \times V_{\text{orig}}(\mu) \to \mathbb{R}$ and $\mu \in \mathcal{P} \mapsto f_{\text{orig}}(\cdot; \mu) : V_{\text{orig}}(\mu) \to \mathbb{R}$, respectively; we then seek $u_{\text{orig}}(\mu) \in V_{\text{orig}}(\mu)$ solution of $a_{\text{orig}}(u_{\text{orig}}(\mu), v; \mu) = f_{\text{orig}}(v; \mu), \forall v \in V_{\text{orig}}(\mu)$. RB methods will rely on some similarity of solutions on the parametric manifold: We thus map Ω_{orig} to a parameter-independent reference domain Ω, $\mu \in \mathcal{P} \mapsto \mathcal{T}(\cdot; \mu) : \Omega \to \Omega_{\text{orig}}(\mu)$; a variety of RB-relevant mapping procedures are described in Chapter 1 of this volume (Volume II) of this handbook. We thereby arrive at the statement (4.9) for $u(\mu) = u_{\text{orig}}(\cdot; \mu) \circ \mathcal{T}(\cdot; \mu)$, which is then the point of departure for the RB formulation. Note for the case of parameter-dependent geometry the functions $\Upsilon_{ij}(\cdot; \mu), 0 \le i, j \le d$, of (4.6) will include the usual transformation terms associated with the Jacobian of the mapping function $\mathcal{T}(\cdot; \mu)$.

4.2.1.2 FE approximation

In general, $\tilde{u}_N(\mu)$ approximates $u(\mu)$, but typically we must construct the RB approximation through a computable intermediary or "surrogate"; in the context of this chapter, and quite often in practice, the latter takes the form of an underlying finite element (FE) approximation. Towards that end, we introduce a conforming FE space $V_h \subset V$ of dimension N_h. We shall choose V_h such that, for any $\mu \in \mathcal{P}$ (say), $\|u(\mu) - u_h(\mu)\|_V \le \text{tol}_V/2$, where tol_V is the prescribed error tolerance; we will then subsequently require $\|u_h(\mu) - \tilde{u}_N(\mu)\|_V \le \text{tol}_V/2$ to ensure $\|u(\mu) - \tilde{u}_N(\mu)\|_V \le \text{tol}_V$.

We shall require for purposes of our subsequent RB approximation that V_h is independent of μ. As we shall see, the RB online cost is largely independent of N_h, however the offline cost will indeed depend on N_h, and hence we may be conservative but not profligate in the design of the FE approximation space. We might construct V_h as follows: Consider a representative sequence of parameter values $\Xi_{\text{FE}} \equiv \{\mu^i_{\text{FE}} \in \mathcal{P}\}_{i=1,\dots,K_{\text{FE}}}$;

initialize the FE approximation for parameter value $\boldsymbol{\mu}_{\mathrm{FE}}^i$, V_h^i, as V_h^{i-1} (and for V_h^1 choose an initial uniform coarse mesh) and adaptively refine to the desired error tolerance $\mathrm{tol}_V/2$; set $V_h = V_h^{K_{\mathrm{FE}}}$. We further introduce the dual space to V_h, V_h', of linear functionals continuous with respect to $\|\cdot\|_V$ for all functions in V_h; we equip V_h' with dual norm

$$\|g_h\|_{V_h'} = \sup_{v \in V_h} \frac{|g_h(v)|}{\|v\|_{V_h}}. \tag{4.12}$$

We may then define the Riesz representation of any g_h in V_h', $\mathcal{R}_h g_h \in V_h$, by

$$(\mathcal{R}_h g_h, v)_V = g_h(v), \quad \forall v \in V_h, \tag{4.13}$$

in terms of which we can evaluate the dual norm from

$$\|g_h\|_{V_h'} = \|\mathcal{R}_h g_h\|_V. \tag{4.14}$$

We note that (4.13) is a finite-dimensional problem.

We now define the (continuous) Galerkin-FE approximation: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $u_h(\boldsymbol{\mu}) \in V_h$ such that

$$a(u_h(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \quad \forall v \in V_h, \tag{4.15}$$

and evaluate the scalar output $s_h(\boldsymbol{\mu}) \in \mathbb{R}$ as $s_h(\boldsymbol{\mu}) = \ell(u_h(\boldsymbol{\mu}); \boldsymbol{\mu})$. We denote the FE version of the constants in (4.7) – with V replaced by V_h – by subscript h: for all $\boldsymbol{\mu} \in \mathcal{P}$, $\alpha(\boldsymbol{\mu}) \leq \alpha_h(\boldsymbol{\mu}) \leq \gamma_h(\boldsymbol{\mu}) \leq \gamma(\boldsymbol{\mu})$; however, in general $\beta_h(\boldsymbol{\mu}) \not\gtrless \beta(\boldsymbol{\mu})$, and thus in the noncoercive case we include the additional hypothesis (on V_h) $\beta_h(\boldsymbol{\mu}) > 0$, in order to ensure well-posedness of the FE approximation. We may then also define the corresponding worst case constants (over $\mathcal{P}$) as α_h, γ_h, and β_h. Finally, we introduce $c_h^{\mathrm{s}}(\boldsymbol{\mu}) \equiv \max(\alpha_h(\boldsymbol{\mu}), \beta_h(\boldsymbol{\mu}))$ and corresponding worst case (minimum over $\mathcal{P}$) constant c_h^{s}.

We next represent V_h by a nodal basis $\{\varphi^j\}_{j=1}^{N_h}$ associated to nodes $\{x_j^{\mathrm{node}} \in \mathbb{R}^d\}_{j=1,\dots,N_h}$; given any function $v_h \in V_h$, we shall denote by $\mathbf{v}_h \in \mathbb{R}^{N_h}$ the corresponding vector of (nodal) basis coefficients. The FE discrete equations now directly follow from (4.15) and our basis for V_h: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $\mathbf{u}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ such that

$$\mathbb{A}_h(\boldsymbol{\mu}) \mathbf{u}_h(\boldsymbol{\mu}) = \mathbf{f}_h(\boldsymbol{\mu}), \tag{4.16}$$

and evaluate the scalar output $s_h(\boldsymbol{\mu}) \in \mathbb{R}$ as $s_h(\boldsymbol{\mu}) = \boldsymbol{\ell}_h^{\mathrm{T}}(\boldsymbol{\mu}) \mathbf{u}_h(\boldsymbol{\mu})$ (for $^{\mathrm{T}}$ the transpose operator). Here $\mathbb{A}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$, $\mathbf{f}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, and $\boldsymbol{\ell}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ are given by

$$(\mathbb{A}_h(\boldsymbol{\mu}))_{ij} = a(\varphi^j, \varphi^i; \boldsymbol{\mu}), \quad (\mathbf{f}_h(\boldsymbol{\mu}))_i = f(\varphi^i; \boldsymbol{\mu}), \quad (\boldsymbol{\ell}_h(\boldsymbol{\mu}))_i = \ell(\varphi^i; \boldsymbol{\mu}), \quad 1 \leq i, j \leq N_h; \tag{4.17}$$

note that $\mathbb{A}_h$ and $\mathbf{f}_h$ are the FE stiffness and load vector, respectively.

For future reference we also introduce the parameter-independent inner-product matrix $\mathbb{X}_h \in \mathbb{R}^{N_h \times N_h}$,

$$(\mathbb{X}_h)_{ij} = (\varphi^j, \varphi^i)_V, \quad 1 \le i,j \le N_h. \tag{4.18}$$

We note that for any $w_h \in V_h$, $v_h \in V_h$, $(w_h, v_h)_V = \mathbf{w}_h^T \mathbb{X}_h \mathbf{v_h}$. Furthermore, it follows from (4.13) that for any $g_h \in V_h'$ the FE basis coefficients of the Riesz representation $\mathcal{R}_h g_h$ are given by $\mathbb{X}_h^{-1} \mathbf{g}_h$ for $(\mathbf{g}_h)_i = g_h(\varphi^i)$, $1 \le i \le N_h$. It follows from (4.14) that the dual norm $\|g_h\|_{V_h'}$ may be evaluated as

$$\left(\mathbf{g}_h^T \mathbb{X}_h^{-1} \mathbf{g}_h\right)^{1/2}; \tag{4.19}$$

note that we require only the action of $\mathbb{X}_h^{-1}$, which might be effected (in the direct context) through Cholesky decomposition and subsequent forward/back substitution.

In actual practice the FE matrices and vectors are formed by numerical quadrature: The integral of (4.6) is replaced by a corresponding sum with quadrature points and weights

$$x_j^{\text{quad},\Omega} \in \Omega, \quad \rho_j^{\text{quad},\Omega} \in \mathbb{R}_+, \quad j = 1, \ldots, N_h^{\text{quad},\Omega}, \tag{4.20}$$

and the integrals of (4.5) are replaced by corresponding sums with quadrature points and weights

$$x_j^{\text{quad},\Gamma_{N,R}} \in \Gamma_{N,R}, \quad \rho_j^{\text{quad},\Gamma_{N,R}} \in \mathbb{R}_+, \quad j = 1, \ldots, N_h^{\text{quad},\Gamma_{N,R}}; \tag{4.21}$$

for simplicity of exposition, we shall presume that the error induced by quadrature is negligible relative to tol_V. The latter is plausible if h is sufficiently small and furthermore $\Upsilon_{i,j}$, $0 \le i,j \le d$, f_Ω, $f_{\Gamma_{N,R}}$, ℓ_Ω, and $\ell_{\Gamma_{N,R}}$ are sufficiently smooth. Note that in practice the FE quadrature is effected as a sum of elemental quadratures.

4.2.1.3 Affine parameter dependence

Affine dependence of the forms $\{a, f, \ell\}$ on the parameter $\boldsymbol{\mu}$ greatly reduces the computational complexity of the online stage of the RB method. We note that parameterized model order reduction can proceed without (appeal to) affine parameter dependence – but less effectively than if we can and do take advantage of affine parameter dependence. We thus wish either to confirm affine parameter dependence or alternatively, and more generally, to impose affine parameter dependence though approximate forms $\{\tilde{a}, \tilde{f}, \tilde{\ell}\} \approx \{a, f, \ell\}$; we pursue here the latter, which includes the former as a special case.

In particular, we introduce $\tilde{a}$, $\tilde{f}$, and $\tilde{\ell}$ which can be expressed, for all $\boldsymbol{\mu} \in \mathcal{P}$, as

$$\tilde{a}(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) a^q(w, v), \quad \tilde{f}(v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) f^q(v), \quad \tilde{\ell}(v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_\ell} \Theta_\ell^q(\boldsymbol{\mu}) \ell^q(v),$$

(4.22)

where $\Theta_a^q : \mathcal{P} \to \mathbb{R}, 1 \le q \le Q_a, \Theta_f^q : \mathcal{P} \to \mathbb{R}, 1 \le q \le Q_f$, and $\Theta_\ell^q : \mathcal{P} \to \mathbb{R}, 1 \le q \le Q_\ell$, are suitably smooth functions, $a^q : V \times V \to \mathbb{R}, 1 \le q \le Q_a$, are parameter-independent continuous bilinear forms, and $f^q : V \to \mathbb{R}, 1 \le q \le Q_f, \ell^q : V \to \mathbb{R}, 1 \le q \le Q_\ell$, are parameter-independent continuous linear forms; we assume the forms are linearly independent.

Given $\boldsymbol{\mu} \in \mathcal{P}$, we now seek $\tilde{u}(\boldsymbol{\mu})$ such that

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

(4.23)

and evaluate the scalar output $\tilde{s}(\boldsymbol{\mu}) \in \mathbb{R}$ as $\tilde{s}(\boldsymbol{\mu}) = \tilde{\ell}(\tilde{u}(\boldsymbol{\mu}); \boldsymbol{\mu})$. We also define the corresponding FE approximation: Find $\tilde{u}_h(\boldsymbol{\mu}) \in V_h$ such that

$$\tilde{a}(\tilde{u}_h(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = \tilde{f}(v; \boldsymbol{\mu}), \quad \forall v \in V_h,$$

(4.24)

and evaluate the scalar output $\tilde{s}_h(\boldsymbol{\mu}) \in \mathbb{R}$ as $\tilde{s}_h(\boldsymbol{\mu}) = \tilde{\ell}(\tilde{u}_h(\boldsymbol{\mu}); \boldsymbol{\mu})$. We denote the FE stability and continuity constants (hence over V_h) associated to $\tilde{a}$ by $\tilde{\alpha}_h(\boldsymbol{\mu})$, $\tilde{\gamma}_h(\boldsymbol{\mu})$, and $\tilde{\beta}_h(\boldsymbol{\mu})$ for any $\boldsymbol{\mu} \in \mathcal{P}$; we may then also define the corresponding worst case constants (over $\mathcal{P}$) as $\tilde{\alpha}_h$, $\tilde{\gamma}_h$, and $\tilde{\beta}_h$. Finally, we introduce $\tilde{c}_h^s(\boldsymbol{\mu}) \equiv \max(\tilde{\alpha}_h(\boldsymbol{\mu}), \tilde{\beta}_h(\boldsymbol{\mu}))$ and the corresponding worst case (minimum over $\mathcal{P}$) constant $\tilde{c}_h^s$. We shall shortly provide a perturbation result for $\tilde{c}_h^s$.

The discrete FE equations now read

$$\tilde{\mathbb{A}}_h(\boldsymbol{\mu}) \tilde{\boldsymbol{u}}_h(\boldsymbol{\mu}) = \tilde{\boldsymbol{f}}_h(\boldsymbol{\mu})$$

(4.25)

and $\tilde{s}_h(\boldsymbol{\mu}) = \tilde{\boldsymbol{\ell}}_h^{\mathrm{T}}(\boldsymbol{\mu}) \tilde{\boldsymbol{u}}_h(\boldsymbol{\mu})$, where $\tilde{\mathbb{A}}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}, \tilde{\boldsymbol{f}}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, and $\tilde{\boldsymbol{\ell}}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ are given by

$$(\tilde{\mathbb{A}}_h(\boldsymbol{\mu}))_{ij} = \tilde{a}(\varphi^j, \varphi^i; \boldsymbol{\mu}), \quad (\tilde{\boldsymbol{f}}_h(\boldsymbol{\mu}))_i = \tilde{f}(\varphi^i; \boldsymbol{\mu}), \quad (\tilde{\boldsymbol{\ell}}_h(\boldsymbol{\mu}))_i = \tilde{\ell}(\varphi^i; \boldsymbol{\mu}), \quad 1 \le i, j \le N_h.$$

(4.26)

We further note from (4.22) and (4.26) that

$$\tilde{\mathbb{A}}_h(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \mathbb{A}_h^q, \quad \tilde{\boldsymbol{f}}_h(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu}) \boldsymbol{f}_h^q, \quad \tilde{\boldsymbol{\ell}}_h(\boldsymbol{\mu}) = \sum_{q=1}^{Q_\ell} \Theta_\ell^q(\boldsymbol{\mu}) \boldsymbol{\ell}_h^q,$$

(4.27)

for parameter-independent $\mathbb{A}_h^q \in \mathbb{R}^{N_h \times N_h}, 1 \le q \le Q_a, \boldsymbol{f}_h^q \in \mathbb{R}^{N_h}, 1 \le q \le Q_f$, and $\boldsymbol{\ell}_h^q \in \mathbb{R}^{N_h}, 1 \le q \le Q_\ell$; for example, $(\mathbb{A}_h^q)_{ij} = a^q(\varphi^j, \varphi^i), 1 \le i, j \le N_h, 1 \le q \le Q_a$.

We must now seek affine approximations (4.22) such that $\{\tilde{a}, \tilde{f}, \tilde{\ell}\}$ is sufficiently close to $\{a, f, \ell\}$, and hence $\tilde{u}$ and $\tilde{u}_h$ are sufficiently close to u and u_h, respectively. Towards that end, we first introduce (a restricted form of) the EIM [5]. We are given integer $N_{\text{EIM}} \geq 1$ (in practice, large) and a function $\mathbf{g} : \mathcal{P} \to \mathbb{R}^{N_{\text{EIM}}}$; we further define a train parameter sample of size K_{EIM}, $\Xi_{\text{EIM}} \equiv \{\boldsymbol{\mu}^i \in \mathcal{P}\}_{i=1,\dots,K_{\text{EIM}}} (\subset \mathcal{P})$, and generate the associated snapshot set $G_{\text{EIM}} \equiv \{\mathbf{g}(\boldsymbol{\mu})\}_{\boldsymbol{\mu} \in \Xi_{\text{EIM}}}$. Lastly, we prescribe norm $\| \cdot \|_{\text{EIM}}$ and associated error tolerance tol_{EIM}.

- In the offline stage, we execute Algorithm 4.1 (presented in detail below) to obtain interpolation indices $\{i_m^* \in \{1, \dots, N_{\text{EIM}}\}\}_{m=1,\dots,M}$, associated interpolation vectors $\{\boldsymbol{\xi}^m \in \mathbb{R}^{N_{\text{EIM}}}\}_{m=1,\dots,M}$, and a nonsingular lower triangular interpolation matrix $B_M \in \mathbb{R}^{M \times M}$.

- In the online stage, given $\boldsymbol{\mu} \in \mathcal{P}$, we approximate $\mathbf{g}(\boldsymbol{\mu})$ as

$$\tilde{\mathbf{g}}(\boldsymbol{\mu}) = \sum_{m=1}^{M} b_m(\boldsymbol{\mu}) \, \boldsymbol{\xi}^m, \tag{4.28}$$

where $b(\boldsymbol{\mu}) \in \mathbb{R}^M$ is the solution of $B_M b(\boldsymbol{\mu}) = \mathbf{g}^*(\boldsymbol{\mu})$ for $\mathbf{g}^*(\boldsymbol{\mu}) \in \mathbb{R}^M$ given by $(\mathbf{g}^*(\boldsymbol{\mu}))_m = (\mathbf{g}(\boldsymbol{\mu}))_{i_m^*}$, $1 \leq m \leq M$. For succinctness in the description of Algorithm 4.1 we define $\mathcal{I}_M : \mathbb{R}^{N_{\text{EIM}}} \to \mathbb{R}^{N_{\text{EIM}}}$ such that (4.28) reads $\tilde{\mathbf{g}}(\boldsymbol{\mu}) = \mathcal{I}_M \mathbf{g}(\boldsymbol{\mu})$ for $M \geq 1$; for $M = 0$ we set $\mathcal{I}_M \equiv 0$.

Algorithm 4.1: Empirical interpolation method (EIM): The EIM algorithm is of the "greedy" variety (also invoked in the identification of V_N, as discussed in Section 4.4.1): on line 4 we choose for the next parameter value the point in Ξ_{EIM} for which the current EIM approximation is worst; within the EIM context, this strategy in conjunction with line 5 also ensures a stable interpolation procedure. We assume that the set G_{EIM} is not embedded in a small finite-dimensional space.

Data: $N_{\text{EIM}}, \Xi_{\text{EIM}}, \mathbf{g} : \mathcal{P} \to \mathbb{R}^{N_{\text{EIM}}}$ (in fact, G_{EIM} suffices), $\| \cdot \|_{\text{EIM}}, \text{tol}_{\text{EIM}}$
Result: M, $\{i_m^* \in \{1, \dots, N_{\text{EIM}}\}\}_{1 \leq m \leq M}$, $\{\boldsymbol{\xi}^m \in \mathbb{R}^{N_{\text{EIM}}}\}_{1 \leq m \leq M}$, $B_M \in \mathbb{R}^{M \times M}$

1 Set $M = 0$ and err $= \infty$;
2 **while** err $> \text{tol}_{\text{EIM}}$ **do**
3   Set $M \leftarrow M + 1$;
4   Find $\boldsymbol{\mu}^* = \arg\sup_{\boldsymbol{\mu} \in \Xi_{\text{EIM}}} \| \mathbf{g}(\boldsymbol{\mu}) - \mathcal{I}_{M-1} \mathbf{g}(\boldsymbol{\mu}) \|_{\text{EIM}}$;
5   Find $i_M^* = \arg\sup_{i \in \{1,\dots,N_{\text{EIM}}\}} |(\mathbf{g}(\boldsymbol{\mu}^*) - \mathcal{I}_{M-1} \mathbf{g}(\boldsymbol{\mu}^*))_i|$;
6   Define $\boldsymbol{\xi}^M = (\mathbf{g}(\boldsymbol{\mu}^*) - \mathcal{I}_{M-1} \mathbf{g}(\boldsymbol{\mu}^*)) / (\mathbf{g}(\boldsymbol{\mu}^*) - \mathcal{I}_{M-1} \mathbf{g}(\boldsymbol{\mu}^*))_{i_M^*}$;
7   Update $(B_M)_{jk} = (\boldsymbol{\xi}^k)_{i_j^*}, 1 \leq j, k \leq M$;
8   Set err $= \| \mathbf{g}(\boldsymbol{\mu}^*) - \mathcal{I}_{M-1} g(\boldsymbol{\mu}^*) \|_{\text{EIM}}$;
9 **end**

By construction, for any $\boldsymbol{\mu} \in \Xi_{\mathrm{EIM}}$, $\|\mathbf{g}(\boldsymbol{\mu}) - \tilde{\mathbf{g}}(\boldsymbol{\mu})\|_{\mathrm{EIM}} \leq \mathrm{tol}_{\mathrm{EIM}}$. However, the EIM approximation may (and must, in practice) be applied to values of $\boldsymbol{\mu} \in \mathcal{P}$ which do not appear in the train parameter sample Ξ_{EIM}.[2] We note that in infinite precision we obtain $M \leq \dim(\mathrm{span}\{G_{\mathrm{EIM}}\})$.

We now apply the EIM to develop approximate affine forms for our PDE. We consider the coefficient function Υ_{00} of (the numerical quadrature version of) (4.6). We set $N_{\mathrm{EIM}} \equiv N_h^{\mathrm{quad},\Omega}$ (or $N_h^{\mathrm{quad},\Gamma_{N,R}}$ for $f_{\Gamma_{N,R}}$, $\ell_{\Gamma_{N,R}}$), identify $(\mathbf{g}(\boldsymbol{\mu}))_i \equiv \Upsilon_{00}(x_i^{\mathrm{quad},\Omega}, \boldsymbol{\mu})$, $1 \leq i \leq N_h^{\mathrm{quad},\Omega}$, and choose $\|\cdot\|_{\mathrm{EIM}} \equiv \|\cdot\|_{\ell^\infty}$. We then apply Algorithm 4.1 to obtain M, the interpolation indices $\{i_m^*\}_{1 \leq m \leq M}$, interpolation vectors $\{\boldsymbol{\xi}^m\}_{1 \leq m \leq M}$, and interpolation matrix B_M associated to the Υ_{00} contribution to the affine sum for the bilinear form $\tilde{a}$. (Each of the other coefficient functions Υ_{ij}, $0 \leq i,j \leq d$, and $(i,j) \neq (0,0)$, f_Ω, and ℓ_Ω is treated [separately] in the same fashion, as well as $f_{\Gamma_{N,R}}$ and $\ell_{\Gamma_{N,R}}$.) Note that the corresponding $\Theta_a^{\cdot}(\boldsymbol{\mu})$ correspond to the $b_{\cdot}(\boldsymbol{\mu})$ of (4.28) and are thus defined implicitly in terms of interpolation indices, vectors, and matrices; the corresponding $a^{\cdot}(w,v)$ are given by

$$\sum_{k=1}^{N_h^{\mathrm{quad},\Omega}} (\boldsymbol{\xi}^{\cdot})_k \, \varphi^j(x_k^{\mathrm{quad},\Omega}) \, \varphi^i(x_k^{\mathrm{quad},\Omega}) \, \rho_k^{\mathrm{quad},\Omega} \tag{4.29}$$

for quadrature weights and points defined by (4.20).

Assuming that each of the coefficient functions resides on a low-dimensional parametric manifold, we may anticipate that we can obtain a corresponding EIM approximation with relatively few terms, M small. For any given $\mathcal{D} \subset \mathcal{P}$, we define the error induced by the EIM approximation in our bilinear and linear forms as $\epsilon_{\mathrm{EIM}}^{\mathcal{D}}$,

$$\epsilon_{\mathrm{EIM}}^{\mathcal{D}} \equiv \sup_{\boldsymbol{\mu} \in \mathcal{D}} \max\left(\sup_{w \in V, v \in V} \frac{|a(w,v;\boldsymbol{\mu}) - \tilde{a}(w,v;\boldsymbol{\mu})|}{\|w\|_V \|v\|_V}, \sup_{v \in V} \frac{|f(v;\boldsymbol{\mu}) - \tilde{f}(v;\boldsymbol{\mu})|}{\|v\|_V}, \right.$$
$$\left. \sup_{v \in V} \frac{|\ell(v;\boldsymbol{\mu}) - \tilde{\ell}(v;\boldsymbol{\mu})|}{\|v\|_V} \right). \tag{4.30}$$

For a of the form (4.6) application of the Cauchy–Schwarz inequality yields $\epsilon_{\mathrm{EIM}}^{\Xi_{\mathrm{EIM}}}$ as a function of $\mathrm{tol}_{\mathrm{EIM}}$; for example, in the absence of off-diagonal terms in (4.6), $\epsilon_{\mathrm{EIM}}^{\Xi_{\mathrm{EIM}}} = \max(1/c_{L^2}, 1) \, \mathrm{tol}_{\mathrm{EIM}}$. We can further argue that, for Ξ_{EIM} sufficiently rich, $\epsilon_{\mathrm{EIM}}^{\mathcal{P}} \approx \epsilon_{\mathrm{EIM}}^{\Xi_{\mathrm{EIM}}}$; an adaptive procedure has been proposed in [27] to support this argument. Finally, it can readily be demonstrated that

$$\tilde{c}_h^{\mathrm{s}} = c_h^{\mathrm{s}} - \epsilon_{\mathrm{EIM}}^{\mathcal{P}}, \tag{4.31}$$

---

2 In a similar fashion, in the most general EIM formulation, $\mathbf{g}(\boldsymbol{\mu})$ corresponds to the evaluation of a function $g : \Omega \times \mathcal{P} \to \mathbb{R}$ at a set of (here, quadrature) points in Ω; however, the resulting EIM approximation can then be applied for any $x \in \Omega$.

and also $\tilde{\gamma}_h \leq \gamma_h + \epsilon_{\mathrm{EIM}}^{\mathcal{P}}$; it follows that our EIM-perturbed FE problem is well-posed for $\epsilon_{\mathrm{EIM}}^{\mathcal{P}}$ sufficiently small.

There is an alternative EIM approach to the development of the affine forms: the operator EIM (OEIM) [15]. In this case we would apply the EIM method directly (for $N_{\mathrm{EIM}} = N_h$) to $\boldsymbol{f}_h(\boldsymbol{\mu})$, $\boldsymbol{\ell}_h(\boldsymbol{\mu})$ and (for $N_{\mathrm{EIM}} = N_h^2$) to $\mathbf{A}_h \in \mathbb{R}^{N_h^2}$; the latter is the single-index (vector) form of the stiffness matrix $\mathbb{A}_h$, which is then repacked in double-index form once the OEIM is complete. The OEIM has the important advantage of nonintrusiveness: the affine approximation may be deduced solely from the FE stiffness matrix and load vector without any knowledge of the associated formation processes, thus permitting a general interface between FE code and the RB code. It is important to choose a norm $\| \cdot \|_{\mathrm{EIM}}$ for the OEIM procedure to permit ultimate error control of the RB approximation in $\| \cdot \|_V$. For $\boldsymbol{f}_h$ and $\boldsymbol{\ell}_h$ we may choose $\|\boldsymbol{\delta g}\|_{\mathrm{EIM}} = (\boldsymbol{\delta g}^{\mathrm{T}} \mathbb{X}_h^{-1} \boldsymbol{\delta g})^{1/2}$, which can be reasonably rapidly estimated. For $\mathbf{A}_h$, the relevant norm can be evaluated as the square root of the maximum eigenvalue $\lambda_{\max}^{\boldsymbol{\delta A}_h}$ associated to the generalized SPD eigenproblem $\boldsymbol{\delta A}_h^{\mathrm{T}} \mathbb{X}_h^{-1} \boldsymbol{\delta A}_h \chi^{\boldsymbol{\delta A}_h} = \lambda^{\boldsymbol{\delta A}_h} \mathbb{X}_h \chi^{\boldsymbol{\delta A}_h}$, hence somewhat cumbersome. For these norm choices we directly obtain $\epsilon_{\mathrm{EIM}}^{\Xi_{\mathrm{EIM}}} = \mathrm{tol}_{\mathrm{EIM}}$. In summary, the OEIM offers a very easily implemented procedure for the construction of affine approximations.

We briefly revisit Example 1.0, described by equations (4.9)–(4.11). We first note from inspection that we can directly choose $\{\tilde{a}, \tilde{f}, \tilde{\ell}\} = \{a, f, \ell\}$ to obtain an affine representation with $Q_a = 3$, $Q_f = 2$, $Q_\ell = 1$. For our particular example Υ_{00}, Υ_{11}, Υ_{22}, and Υ_{33} are nonzero, and hence the EIM procedure described – which treats each term in the expansion (4.6) separately – would yield $Q_a = 7$; a concatenated EIM – in which we treat all the Υ_{ij}, $0 \leq i, j \leq d$, within a single EIM – would recover $Q_a = 3$. The OEIM procedure, which treats the entire form, would directly recover $Q_a = 3$, $Q_f = 2$, $Q_\ell = 1$. It is often the case for problems in which the geometry does not depend on the parameter that $\{a, f, \ell\}$ admits an exact affine representation, $\epsilon_{\mathrm{EIM}}^{\mathcal{P}} = 0$, with relatively few terms. However, in the presence of parameter-dependent geometry, and in particular nonaffine geometry transformations, $\{a, f, \ell\}$ will not admit an exact affine representation.

4.2.2 Justification for reduction

The fundamental hypothesis made on the parametric manifold $\mathcal{M} \equiv \{u(\boldsymbol{\mu}) \mid \boldsymbol{\mu} \in \mathcal{P}\}$ introduced in the first section is its "reducibility" in the sense that there supposedly exist(s) some (series of) finite-dimensional space(s) V_N that approximate well $\mathcal{M}$ in the sense that, denoting by $\mathrm{dist}(\mathcal{M}, V_N)$ the deviation of $\mathcal{M}$ from V_N, i. e.,

$$\mathrm{dist}(\mathcal{M}, V_N) = \sup_{u(\boldsymbol{\mu})} \inf_{v_N \in V_N} \|u(\boldsymbol{\mu}) - v_N\|_V,$$

$\mathrm{dist}(\mathcal{M}, V_N)$ is decreasing fast with N increasing. The question we want to raise here is: Why should it be so? And also, what is that (series of) finite-dimensional space(s) V_N?

This hypothesis is formally stated by going one step further in the definition of the deviation, i. e., introducing the quantity

$$d_N(\mathcal{M}, V) = \inf_{V_N, \dim V_N = N} \text{dist}(\mathcal{M}, V_N), \tag{4.32}$$

which is known as the Kolmogorov N-width [30, 23] and represents the ability of $\mathcal{M}$ to be approximated by some optimally chosen vectorial space of dimension N.

As has already been remarked above, the "optimal" choice V_N depends on $\mathcal{M}$, and the optimal choice for $\mathcal{M}$ will not be valid for another set of functions.

This notion is the right one, indeed, when, e. g., $d_N(\mathcal{M}, V)$ goes to 0, like ρ^N with $0 < \rho < 1$ or even like cN^{-p} with p large enough (say, $p \geq 6$), then we are in a good shape, expecting that, for any $\boldsymbol{\mu} \in \mathcal{P}$, very few (well-chosen) degrees of freedom (the coefficient in some appropriate basis of V_N) will be sufficient to approximate well any $u(\boldsymbol{\mu})$.

We know that in some cases, e. g., a linear structure of the PDE, by superposition it is possible to check that $\mathcal{M}$ is finite-dimensional. Of course this is neither the generic case nor the case we are interested in. Then typically, regularity of the solutions with respect to the spatial variable may lead to propose, for every $\boldsymbol{\mu}$, a high-order (say, polynomial or spectral) approximation that converges rapidly, even exponentially (when the degree of the polynomial increases) and thus, X_N can be chosen as the set of polynomials of degree $\leq cN^{1/d}$. This will not be the optimal space but the optimal choice is better than the polynomial choice and thus the best polynomial fit provides an upper bound for the Kolmogorov N-width. However, the fact that polynomials approximate well any regular function, regardless of the property of the whole set $\mathcal{M}$, i. e., its structure, shape, coherence, makes it understandable that this "generic" choice cannot be the optimal one for $\mathcal{M}$, and that it may be much, much better.

As noted in [12], if the mapping $\boldsymbol{\mu} \in \mathcal{P} \mapsto u(\boldsymbol{\mu})$ is linear continuous, then the Kolmogorov N-width of $\mathcal{M}$ is upper bounded by a constant times the Kolmogorov N-width of $\mathcal{P}$. The generalization of this statement that is proposed in [12] is that if the previous mapping is holomorphic (meaning that $u(\boldsymbol{\mu})$ has a Fréchet derivative at any parameter $\boldsymbol{\mu}$ belonging to a compact set $K \subset \mathcal{P}$), then, for any $s > 1$ and $t < s - 1$,

$$\sup_{n \geq 1} n^s d_n(K, \mathcal{P}) < \infty \implies \sup_{n \geq 1} n^t d_n(u(K), V) < \infty, \tag{4.33}$$

where $u(K) = \{u(\boldsymbol{\mu}), \boldsymbol{\mu} \in K\}$. The loss of 1 (with respect to the linear case) may be not optimal but this result provides extra reasons for $d_N(\mathcal{M}, V)$ to be small when $u(\boldsymbol{\mu})$ is the solution to some parameter-dependent PDE. Indeed, it is well known that, by differentiating the PDE, $\nabla_{\boldsymbol{\mu}} u(\boldsymbol{\mu})$ is the solution to a similar PDE as $u(\boldsymbol{\mu})$, and thus under a reasonable hypothesis, this set satisfies the holomorphic assumption.

In order to explain faster rates of convergence, the first analysis we know of is [26], where exponential convergence was proven for a simple one-dimensional parameter space elliptic PDE. More recently a general extension was performed in [3], where, by

using, in a constructive way, low-rank tensor approximations for families of elliptic diffusion PDEs parameterized by the diffusion coefficients, the authors have been able to derive exponential convergence rates in a much more general framework.

A typical case where the Kolmogorov N-width is not small (at least in a straightforward manner) is when the problem is convection-dominated. Indeed, the set of solutions to a simple pure linear convection problem, when the velocity is among the parameters, is the initial solution, properly translated: If it is not regular, then the set of all solutions is of very large Kolmogorov N-width. There are ways to circumvent this (see, e. g., [10, 7]) but it is out of the scope of this contribution. Another case where, a priori, the Kolmogorov N-width is not small (but it is simple to fix) is for elliptic problems where the right-hand side contains pointwise singularities, the position of which may vary and is one parameter of the problem. The "trace" of these singularities of the right-hand side can be clearly "seen" on the solution $u(\boldsymbol{\mu})$ itself and this may lead also to a large Kolmogorov N-width. A simple postprocessing of the solution's manifold through a change of variable that maps the singularities' positions to a fixed reference position allows to better compare the solutions and check that indeed, when the singularities are sort of "aligned," the set of all (postprocessed) solutions is of small Kolmogorov N-width.

4.2.3 Extensions

4.2.3.1 Generalization of linear elliptic problems

The extension of our parameterized PDE formulation to vector fields is very simple. In particular, and if we assume that Dirichlet conditions at any point on the boundary are always applied to all components of the vector, we need only redefine $V^d \rightarrow V$. Thus linear elasticity readily falls into the framework considered in this chapter. It is interesting to emphasize here that the manifold of all vector fields (solutions to the problem of interest when the parameter varies) can be considered as a whole, which leads us to approximate a vector solution as a linear combination of vector (reduced) basis functions with scalar coefficients, thereby further decreasing the complexity of the online procedure. Note that we shall not consider here saddle problems: These mixed formulations – such as the incompressible Stokes equations or the equations of linear elasticity for Poisson ratio approaching 1/2 – require special RB treatment [34].

The extension of our parameterized PDE formulation to complex fields – "complexification" for short – is also relatively simple. We must change $\mathbb{R}$ to $\mathbb{C}$, interpret $|\cdot|$ as complex modulus, consider spaces V of complex-valued functions, conjugate the argument of all linear (now anti-linear) forms and the second argument of all inner products and bilinear (now sesquilinear) forms, and in our discrete representations replace transpose $^{\mathrm{T}}$ with Hermitian $^{\mathrm{H}}$. We provide a simple illustration, the Helmholtz problem of acoustics, which we shall denote Example 2.0.

We consider $\Omega \subset \mathbb{R}^{d=3}$. We set $p = 2$ and introduce parameter $\boldsymbol{\mu} \equiv (\mu_1, \mu_2) \in \mathcal{P} \subset \{v \in \mathbb{R}^2 \mid v_1 \geq 0, v_2 \geq 0\}$. We then define

$$a(w, v; \boldsymbol{\mu}) = \int_\Omega (1 + \mathrm{i}\,\mu_2) \nabla w \cdot \nabla \bar{v} - \mu_1 w \bar{v}, \quad \forall w, v \in V^2, \tag{4.34}$$

as well as suitable $f(\cdot; \boldsymbol{\mu})$ and $\ell(\cdot; \boldsymbol{\mu})$. Here $\mathrm{i}^2 = -1$ and $\bar{v}$ denotes the complex conjugate of v. For $\mu_2 > 0$ (positive dissipation), a is coercive. For $\mu_2 = 0$ (no dissipation), a is inf-sup stable unless μ_1 is an eigenvalue λ of the associated "resonance" problem: Find $(\chi \in V, \lambda_{\mathrm{resonance}} \in \mathbb{R})$ such that $\int_\Omega \nabla \chi \cdot \nabla v = \lambda_{\mathrm{resonance}} \int_\Omega \chi v$, $\forall v \in V$, where here V is our standard real space. In some applications, such as the elastodynamics extension of (4.34), the actual dissipation can be substantial; in other applications, such as acoustics, μ_2 must often be interpreted as a numerical regularization parameter.

4.2.3.2 Evolution problems: parabolic PDEs

We shall consider here only parabolic PDEs. In fact, RB methods can also readily be applied to hyperbolic PDEs, for example the second-order wave equation, however in the absence of adequate dissipation the treatment of rough initial conditions and limited regularity remains an outstanding issue as indicated above. We shall assume for our parabolic PDEs that our bilinear form a is coercive; the noncoercive case is more difficult in particular as regards effective a posteriori error estimation [36].

We introduce the time variable t and temporal domain $(0, T]$, and the space $L^2(\Omega)$ and associated inner product $(\cdot, \cdot)_0$ and induced norm $\| \cdot \|_0$. We further define inner product $\boldsymbol{\mu} \in \mathcal{P} \mapsto m(\cdot, \cdot; \boldsymbol{\mu}) : V \times V \to \mathbb{R}$ which induces norm $m^{1/2}(\cdot, \cdot; \boldsymbol{\mu})$ equivalent to $\| \cdot \|_0$. We now state the weak form of our parabolic PDE: Given $\boldsymbol{\mu} \in \mathcal{P}$, we look for $u(\boldsymbol{\mu}) \in C^0((0, T]; L^2(\Omega)) \cap L^2((0, T]; V)$ such that, for any time t,

$$m\left(\frac{\partial u(t; \boldsymbol{\mu})}{\partial t}, v; \boldsymbol{\mu}\right) + a(u(t; \boldsymbol{\mu}), v; \boldsymbol{\mu}) = \tau(t) f(v; \boldsymbol{\mu}), \quad \forall v \in V, \tag{4.35}$$

where $\tau \in L^2((0, T])$ and $f(\cdot; \boldsymbol{\mu}) \in L^2(\Omega)$. We take for initial condition $u(t = 0; \boldsymbol{\mu}) = u_0 \in L^2(\Omega)$; in actual practice, we may also permit parameter-dependent initial conditions. Note that we shall not explicitly present the treatment of the linear functional output, as (in the absence of an adjoint) the latter differs little between the elliptic and parabolic cases.

We inherit the underlying FE approximation from the elliptic problem of Section 4.2.1.2. We further assume that m admits an (EIM-approximate) affine expansion,

$$\tilde{m}(w, v; \boldsymbol{\mu}) = \sum_{q=1}^{Q_m} \Theta_m^q(\boldsymbol{\mu}) m^q(w, v), \tag{4.36}$$

for $\Theta_m^q : \mathcal{P} \to \mathbb{R}$ and parameter-independent $m^q : V \times V \to \mathbb{R}, 1 \leq 1 \leq Q_m$. We may then further define our FE mass matrix, $\tilde{\mathbb{M}}_h(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$,

$$(\tilde{\mathbb{M}}_h(\boldsymbol{\mu}))_{kn} = \tilde{m}(\varphi^n, \varphi^k; \boldsymbol{\mu}), \quad 1 \leq k, n \leq N_h, \tag{4.37}$$

which we may form as

$$\tilde{\mathbb{M}}_h(\boldsymbol{\mu}) = \sum_{q=1}^{Q_m} \Theta_m^q(\boldsymbol{\mu}) \mathbb{M}_h^q, \tag{4.38}$$

for $\mathbb{M}_h^q = m^q(\varphi^n, \varphi^k), 1 \leq k, n \leq N_h, 1 \leq q \leq Q_m$. We also introduce a "truth" finite difference discretization in time: we choose $\Delta t = T/J$ and define $t^j = j\Delta t, 0 \leq j \leq J$.

We can now state the FE approximation: Given $\boldsymbol{\mu} \in \mathcal{P}$, we look for $\tilde{u}_{h,\Delta t}^j(\boldsymbol{\mu}) (\approx u(t^j, .; \boldsymbol{\mu})) \in V_h, j = 1, \ldots, J$, such that

$$\tilde{m}\left(\frac{\tilde{u}_{h,\Delta t}^j(\boldsymbol{\mu}) - \tilde{u}_{h,\Delta t}^{j-1}(\boldsymbol{\mu})}{\Delta t}, v; \boldsymbol{\mu}\right) + \tilde{a}(\tilde{u}_{h,\Delta t}^j(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = \tau(t^j)\tilde{f}(v; \boldsymbol{\mu}), \quad \forall v \in V_h; \tag{4.39}$$

we impose the initial condition $\tilde{u}_{h,\Delta t}^{j=0}(\boldsymbol{\mu}) = u_0$. We can then state the discrete equations to be solved at each time t^j:

$$\left(\tilde{\mathbb{A}}_h(\boldsymbol{\mu}) + \frac{1}{\Delta t}\tilde{\mathbb{M}}_h(\boldsymbol{\mu})\right)\tilde{\boldsymbol{u}}_{h,\Delta t}^j(\boldsymbol{\mu}) = \frac{1}{\Delta t}\tilde{\mathbb{M}}_h(\boldsymbol{\mu})\boldsymbol{u}_{h,\Delta t}^{j-1}(\boldsymbol{\mu}) + \tau(t^j)\tilde{\boldsymbol{f}}_h(\boldsymbol{\mu}). \tag{4.40}$$

Although we consider here Euler backward temporal treatment, the methodology readily extends to higher-order temporal discretizations.

4.2.3.3 A nonlinear elliptic problem

Although our presentation of linear elliptic PDEs and also linear parabolic PDEs is rather general, the scope of this article does not permit similar treatment of nonlinear problems. We thus focus on a particular nonlinear elliptic PDE with rather simple structure and underlying theory. In the linear part of this chapter our goal is to provide a complete picture of the state of the art, albeit with a balance between performance and simplicity, and indeed an emphasis on the latter. In this nonlinear thread our goal is less ambitious: We highlight the new difficulty introduced by nonlinearity and the corresponding new ingredient – hyperreduction – developed to address this difficulty. The hyperreduction treatment presented, as well as other hyperreduction approaches [35, 17, 40], is broadly applicable. However, the a posteriori error estimator takes advantage of our particular simple nonlinearity, and our offline computational procedure is rather inefficient.

We consider a particular nonlinear elliptic PDE with monotonic nondecreasing nonlinearity: Given $\boldsymbol{\mu} \in \mathcal{P} \subset \{v \in \mathbb{R} \mid v \geq 0\} \subset \mathbb{R}^{p \equiv 1}$, find $u(\boldsymbol{\mu}) \in V$ such that

$$\int_\Omega \nabla u(\boldsymbol{\mu}) \cdot \nabla v + \int_\Omega \eta(u(\boldsymbol{\mu}))\, v = \boldsymbol{\mu} \int_\Omega v, \quad \forall v \in V. \tag{4.41}$$

We require that $\eta \in C^1(\mathbb{R})$ and furthermore $\eta(z_2) - \eta(z_1) \geq 0$ for $z_2 > z_1$. Two examples are the classical smooth test problem given by $\eta(z) = z^3$ and the more relevant (and nonpolynomial) drag law $\eta(z) = |z|z$. It can be shown that (4.41) admits a unique solution. (Note for this nonlinear problem we prefer explicit rather than abstract representation of the weak form.)

We may then directly introduce the corresponding FE approximation: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $u_h(\boldsymbol{\mu}) \in V_h$ such that

$$\int_\Omega \nabla u_h(\boldsymbol{\mu}) \cdot \nabla v + \int_\Omega \eta(u_h(\boldsymbol{\mu}))\, v = \boldsymbol{\mu} \int_\Omega v, \quad \forall v \in V_h. \tag{4.42}$$

In actual practice, the integrals in (4.42) should be interpreted as quadrature sums, hence

$$\sum_{j=1}^{N^{\mathrm{quad},\Omega}} \rho_j^{\mathrm{quad},\Omega}\, \nabla u_h(x_j^{\mathrm{quad},\Omega}; \boldsymbol{\mu}) \cdot \nabla v(x_j^{\mathrm{quad},\Omega})$$

$$+ \sum_{j=1}^{N^{\mathrm{quad},\Omega}} \rho_j^{\mathrm{quad},\Omega}\, \eta(u_h(x_j^{\mathrm{quad},\Omega}; \boldsymbol{\mu}))\, v(x_j^{\mathrm{quad},\Omega})$$

$$= \boldsymbol{\mu} \sum_{j=1}^{N^{\mathrm{quad},\Omega}} \rho_j^{\mathrm{quad},\Omega}\, v(x_j^{\mathrm{quad},\Omega}), \quad \forall v \in V_h. \tag{4.43}$$

For future reference and for purposes of consistency with previous notation we define for this nonlinear problem $\alpha = \alpha(\boldsymbol{\mu}) = 1$ and $\alpha_h = \alpha_h(\boldsymbol{\mu}) = 1$.

We now prepare an affine version of our nonlinear problem. Towards that end we apply the EIM approach, in particular Algorithm 4.1, to $\mathbf{g}$ given by $(\mathbf{g}(\boldsymbol{\mu}))_i = \eta(u_h(x_i^{\mathrm{quad},\Omega}; \boldsymbol{\mu}))$, $1 \leq i \leq N^{\mathrm{quad},\Omega}$; we specify $N_{\mathrm{EIM}}^{\mathrm{NL}}$, $\Xi_{\mathrm{EIM}}^{\mathrm{NL}}$, $\|\cdot\|_{\mathrm{EIM}}^{\mathrm{NL}} \equiv \|\cdot\|_{\ell^\infty}$, and $\mathrm{tol}_{\mathrm{EIM}}^{\mathrm{NL}}$, and denote by $\mathcal{I}_M : V_h \to V_h$ the resulting interpolation operator as characterized by M, $\{i_m^*\}_{m=1,\dots,M}$, $\{\boldsymbol{\xi}^i\}_{i=1,\dots,M}$, and B_M. (We provide an NL superscript to the inputs, but context suffices to indicate the NL for the outputs.) We note that each evaluation of $\mathbf{g}(\boldsymbol{\mu})$ for $\boldsymbol{\mu} \in \Xi_{\mathrm{EIM}}^{\mathrm{NL}}$ is now expensive – solution of the FE approximation to our problem – and not simply evaluation of a coefficient function; more efficient alternatives are proposed in the literature [13]. Then, given $\boldsymbol{\mu} \in \mathcal{P}$, we look for $\tilde{u}_h(\boldsymbol{\mu}) \in V_h$ such that

$$\sum_{j=1}^{N^{\text{quad},\Omega}} \rho_j^{\text{quad},\Omega} \, \nabla \tilde{u}_h(x_j^{\text{quad},\Omega}; \boldsymbol{\mu}) \cdot \nabla v(x_j^{\text{quad},\Omega})$$

$$+ \sum_{m,m'=1}^{M} (B_M^{-1})_{mm'} \, \eta(\tilde{u}_h(x_{i_{m'}^*}^{\text{quad},\Omega}, \boldsymbol{\mu})) \left[\sum_{j=1}^{N^{\text{quad},\Omega}} (\boldsymbol{\xi}^m)_j \, v(x_j^{\text{quad},\Omega}) \rho_j^{\text{quad},\Omega} \right]$$

$$= \boldsymbol{\mu} \sum_{j=1}^{N^{\text{quad},\Omega}} \rho_j^{\text{quad},\Omega} v(x_j^{\text{quad},\Omega}), \quad \forall v \in V_h. \quad (4.44)$$

Under the assumption of exact quadrature we may write (4.44) as

$$\int_{\Omega} \nabla \tilde{u}_h(\boldsymbol{\mu}) \cdot \nabla v + \int_{\Omega} \mathcal{I}_M \big[\eta((\tilde{u}_h)(\boldsymbol{\mu})) \big] v = \boldsymbol{\mu} \int_{\Omega} v, \quad \forall v \in V_h, \quad (4.45)$$

where $\mathcal{I}_M$ is the EIM interpolant operator. We emphasize that $\mathcal{I}_M$ is developed through Algorithm 4.1 for $\eta(u_h(x_{i=1,\dots,N^{\text{quad},\Omega}}^{\text{quad},\Omega}; \boldsymbol{\mu}))$, but then applied in (4.45) to $\eta(\tilde{u}_h(x_{i=1,\dots,N^{\text{quad},\Omega}}^{\text{quad},\Omega}; \boldsymbol{\mu}))$. We say that (4.44) is "affine" in the sense that $\eta(\tilde{u}_h(\cdot; \boldsymbol{\mu}))$ no longer appears in the quadrature sum associated with the nonlinear term. The computational importance of this simplification will become clear when we consider RB projection.

4.3 Projection

4.3.1 Elliptic problems

4.3.1.1 Galerkin projection

We consider here the real case, but note that our "complexification" transformation may be directly applied. We are given a hierarchical set of RB spaces $\{V_N\}_{N=1,\dots,N_{\max}}$. In fact, this section is applicable to any RB spaces, but for purposes of concreteness we sketch here the particular space we shall propose in Section 4.4.1. We first introduce parameter sample $S_{N_{\max}} \equiv \{\boldsymbol{\mu}^j \in \mathcal{P}\}_{j=1,\dots,N_{\max}}$, the optimal choice of which shall be discussed in Section 4.4.1. The space V_N, for any given N, is then defined as $\text{span}\{\tilde{u}_h(\boldsymbol{\mu}^j), j = 1, \dots, N\}$. Note that, since $\boldsymbol{\mu}^j, 1 \le j \le N_{\max}$, are independent of N, our RB spaces are nested: $V_1 \subset V_2 \subset \cdots \subset V_{N_{\max}}$.

We now define the RB-Galerkin approximation for some given $N \in \{1, \dots, N_{\max}\}$: Given $\boldsymbol{\mu} \in \mathcal{P}$, find $\tilde{u}_N(\boldsymbol{\mu}) \in V_N$ such that

$$\tilde{a}(\tilde{u}_N(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = \tilde{f}(v; \boldsymbol{\mu}), \quad \forall v \in V_N, \quad (4.46)$$

and evaluate the scalar output $\tilde{s}_N(\boldsymbol{\mu}) \in \mathbb{R}$ as $\tilde{s}_N(\boldsymbol{\mu}) = \tilde{\ell}(\tilde{u}_N(\boldsymbol{\mu}); \boldsymbol{\mu})$. We recall also that we shall denote by $u_N(\boldsymbol{\mu})$ the RB approximation in the absence of EIM errors; the latter corresponds to an effectively exact affine expansion such that $\epsilon_{\text{EIM}}^{\mathcal{P}} = 0$ and thus

$\{\tilde{a}, \tilde{f}, \tilde{\ell}\} = \{a, f, \ell\}$. We denote the RB stability and continuity constants (hence over V_N) associated to $\tilde{a}$ by $\tilde{\alpha}_N(\boldsymbol{\mu})$, $\tilde{\gamma}_N(\boldsymbol{\mu})$, and $\tilde{\beta}_N(\boldsymbol{\mu})$ for any $\boldsymbol{\mu} \in \mathcal{P}$; we may then also define the corresponding worst case constants (over $\mathcal{P}$) as $\tilde{\alpha}_N$, $\tilde{\gamma}_N$, and $\tilde{\beta}_N$. We also introduce $\tilde{c}_N^{\mathrm{s}}(\boldsymbol{\mu}) \equiv \max(\tilde{\alpha}_N(\boldsymbol{\mu}), \tilde{\beta}_N(\boldsymbol{\mu}))$ and corresponding worst case (minimum over $\mathcal{P}$) constant $\tilde{c}_N^{\mathrm{s}}$. It is readily demonstrated that $\tilde{\alpha}_N \geq \tilde{\alpha}_h$, however in general $\tilde{\beta}_N \ngeq \tilde{\beta}_h$. In the noncoercive case we must therefore incorporate $\tilde{\beta}_N > 0$ as an additional hypothesis. Alternatively, we may consider a minimum-residual projection [25], which ensures a stable RB approximation, indeed $\tilde{\beta}_N \geq \tilde{\beta}_h$; however, Galerkin projection – the simplest and least expensive – is typically quite effective in practice.

We also introduce a basis for V_N, $\{\zeta_h^n\}_{n=1,\dots,N}, 1 \leq N \leq N_{\max}$. We shall choose, for purposes of stability, an orthonormal basis: $(\zeta_h^n, \zeta_h^m)_V = \delta_{mn}, 1 \leq m, n \leq N$, for δ_{mn} the Kronecker delta symbol. Given any function $v_N \in V_N$, we denote by $\mathbf{v}_N \in \mathbb{R}^N$ the corresponding vector of basis coefficients. The discrete RB equations then read

$$\tilde{\mathbb{A}}_N(\boldsymbol{\mu})\tilde{\boldsymbol{u}}_N(\boldsymbol{\mu}) = \tilde{\boldsymbol{f}}_N(\boldsymbol{\mu}) \tag{4.47}$$

and

$$\tilde{s}_N(\boldsymbol{\mu}) = \tilde{\boldsymbol{\ell}}_N^{\mathrm{T}}(\boldsymbol{\mu})\tilde{\boldsymbol{u}}_N(\boldsymbol{\mu}), \tag{4.48}$$

where $\tilde{\mathbb{A}}_N(\boldsymbol{\mu}) \in \mathbb{R}^{N \times N}, \tilde{\boldsymbol{f}}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$, and $\tilde{\boldsymbol{\ell}}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ are given by

$$(\tilde{\mathbb{A}}_N(\boldsymbol{\mu}))_{mn} = \tilde{a}(\zeta_h^n, \zeta_h^m; \boldsymbol{\mu}), \quad (\tilde{\boldsymbol{f}}_N(\boldsymbol{\mu}))_m = \tilde{f}(\zeta_h^m; \boldsymbol{\mu}), \quad (\tilde{\boldsymbol{\ell}}_N(\boldsymbol{\mu}))_m = \tilde{\ell}(\zeta_h^m; \boldsymbol{\mu}), \quad 1 \leq m, n \leq N. \tag{4.49}$$

We further note from (4.22) and (4.49) that

$$\tilde{\mathbb{A}}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu})\mathbb{A}_N^q, \quad \tilde{\boldsymbol{f}}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_f^q(\boldsymbol{\mu})\boldsymbol{f}_N^q, \quad \tilde{\boldsymbol{\ell}}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_\ell} \Theta_\ell^q(\boldsymbol{\mu})\boldsymbol{\ell}_N^q, \tag{4.50}$$

for parameter-independent $\mathbb{A}_N^q \in \mathbb{R}^{N \times N}, 1 \leq q \leq Q_a, \boldsymbol{f}_N^q \in \mathbb{R}^N, 1 \leq q \leq Q_f$, and $\boldsymbol{\ell}_N^q \in \mathbb{R}^N$, $1 \leq q \leq Q_\ell$; for example, $(\mathbb{A}_N^q)_{mn} = a^q(\zeta_h^m, \zeta_h^n), 1 \leq m, n \leq N, 1 \leq q \leq Q_a$.

In actual practice we may express our RB matrices and vectors in a nonintrusive fashion which invokes only standard operators and operations readily available in the FE context. To begin, we introduce, for $1 \leq N \leq N_{\max}$, the RB "basis matrix" $\mathbb{V}_N \in \mathbb{R}^{N_h \times N}, (\mathbb{V}_N)_{jn} = (\boldsymbol{\zeta}_h^n)_j, 1 \leq j \leq N_h, 1 \leq n \leq N$. It then follows from (4.49) that

$$\tilde{\mathbb{A}}_N(\boldsymbol{\mu}) = \mathbb{V}_N^{\mathrm{T}}\tilde{\mathbb{A}}_h(\boldsymbol{\mu})\mathbb{V}_N, \quad \tilde{\boldsymbol{f}}_N(\boldsymbol{\mu}) = \mathbb{V}_N^{\mathrm{T}}\tilde{\boldsymbol{f}}_h(\boldsymbol{\mu}), \quad \tilde{\boldsymbol{\ell}}_N(\boldsymbol{\mu}) = \mathbb{V}_N^{\mathrm{T}}\boldsymbol{\ell}_h(\boldsymbol{\mu}). \tag{4.51}$$

In the same fashion, from (4.50) (or directly (4.27)), we may write

$$\mathbb{A}_N^q = \mathbb{V}_N^{\mathrm{T}}\mathbb{A}_h^q\mathbb{V}_N, \ 1 \leq q \leq Q_a, \quad \boldsymbol{f}_N^q = \mathbb{V}_N^{\mathrm{T}}\boldsymbol{f}_h^q, \ 1 \leq q \leq Q_f, \quad \boldsymbol{\ell}_N^q = \mathbb{V}_N^{\mathrm{T}}\boldsymbol{\ell}_h^q, \ 1 \leq q \leq Q_\ell. \tag{4.52}$$

We recall that the FE matrices for our nodal basis are sparse, and hence the operation count to form (say) $\mathbb{A}_N^1$, for $\mathbb{A}_h^1$ already formed and represented in sparse format, is $\mathcal{O}(N^2 N_h)$ floating point operations (FLOPs). We discuss operation counts further in the context of the offline-online decomposition.

4.3.1.2 A priori error estimation

It is a simple application of Céa's lemma and Strang's first lemma to demonstrate [18] that, for any $\boldsymbol{\mu} \in \mathcal{P}$,

$$\|u_h(\boldsymbol{\mu}) - \tilde{u}_N(\boldsymbol{\mu})\|_V \leq \left(1 + \frac{\tilde{\gamma}_N}{\tilde{c}_N^s}\right) \underbrace{\inf_{w_N \in V_N} \|u_h(\boldsymbol{\mu}) - w_N\|_V}_{\text{best-approximation error}} + \frac{\epsilon_{\text{EIM}}^{\mathcal{P}}}{\tilde{c}_N^s}\left(1 + \frac{\|f(\cdot;\boldsymbol{\mu})\|_{V'}}{c_h^s}\right). \quad (4.53)$$

(We can then readily develop an associated error bound for $s_h - \tilde{s}_N$; we defer discussion of the latter to the a posteriori context.) Note in the error estimate (4.53) that $u_h(\boldsymbol{\mu})$ is the "true" FE approximation on which we build the RB approximation and with respect to which we estimate the RB accuracy, and $\tilde{u}_N(\boldsymbol{\mu})$ is the actual RB approximation including EIM approximation of the bilinear and linear forms; hence the bound (4.53) reflects both RB and EIM contributions to the error. In Section 4.4.1 we shall develop estimates for the decay of the best-approximation error with N relative to the corresponding Kolmogorov N-width associated with our parametric manifold.

Finally, we note that in the coercive case the RB discrete equations are provably well-conditioned under the assumption that $\epsilon_{\text{EIM}}^{\mathcal{P}} < \alpha$; the essential ingredient is orthonormalization of the RB basis functions with respect to the V inner product. In particular, it is readily demonstrated that, in the coercive case, the condition number of $\mathbb{A}_N(\mu)$ (measured in the usual ℓ_2-norm) is bounded by $(\gamma + \epsilon_{\text{EIM}}^{\mathcal{P}})/(\alpha - \epsilon_{\text{EIM}}^{\mathcal{P}})$ for all $\boldsymbol{\mu} \in \mathcal{P}$ and independent of N.

4.3.1.3 A posteriori error estimation

4.3.1.3.1 Dual norm of residual

A posteriori error estimators shall serve to control the error in the RB approximation: in the offline stage to (inexpensively) identify good spaces V_N; in the online stage to verify any particular query $\boldsymbol{\mu} \mapsto \tilde{u}_N(\boldsymbol{\mu}), \tilde{s}_N(\boldsymbol{\mu})$. In principle we would wish to control in both the offline stage and the online stage the total error $\|u(\boldsymbol{\mu}) - \tilde{u}_N(\boldsymbol{\mu})\|_V$, and in certain particular (but important) cases this is indeed possible [39]. More generally, we write $\|u(\boldsymbol{\mu}) - \tilde{u}_N(\boldsymbol{\mu})\|_V \leq \|u(\boldsymbol{\mu}) - u_h(\boldsymbol{\mu})\|_V + \|u_h(\boldsymbol{\mu}) - \tilde{u}_N(\boldsymbol{\mu})\|_V$: In the offline stage we control both $\|u - u_h(\boldsymbol{\mu})\|_V$ (as described in Section 4.2.1.2) and $\|u_h(\boldsymbol{\mu}) - \tilde{u}_N(\boldsymbol{\mu})\|_V$ over respective (finite-cardinality) train subsets of the parameter domain $\mathcal{P}$, Ξ_{FE} and Ξ_{EIM}, Ξ_{RB}; in the online stage, for *any* $\boldsymbol{\mu} \in \mathcal{P}$, we control – and in particular verify – only $\|u_h(\boldsymbol{\mu}) - \tilde{u}_N(\boldsymbol{\mu})\|_V$. We justify the online emphasis on only the FE-RB error: the operation count for the online stage shall not depend explicitly on N_h, and thus we may choose the FE approximation space somewhat conservatively; the latter would then accommodate the difference between the offline parameter train set Ξ_{FE} and the full (online) parameter domain $\mathcal{P}$. In the remainder of this section we consider only $\|u_h(\boldsymbol{\mu}) - \tilde{u}_N(\boldsymbol{\mu})\|_V$ (and, briefly, $|s_h(\boldsymbol{\mu}) - \tilde{s}_N(\boldsymbol{\mu})|$).

To begin, we consider the case in which $\epsilon_{\text{EIM}}^{\mathcal{P}} = 0$ and hence $\tilde{u}_N(\boldsymbol{\mu}) = u_N(\boldsymbol{\mu})$; for the purposes of this analysis, we explicitly remove the $\tilde{\cdot}$. We introduce the error $e(\boldsymbol{\mu}) \equiv u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})$ as well as the residual $\boldsymbol{\mu} \mapsto r_h(\cdot; \boldsymbol{\mu}) \in V'$ given by

$$r_h(v; \boldsymbol{\mu}) \equiv f(v; \boldsymbol{\mu}) - a(u_N(\boldsymbol{\mu}), v; \boldsymbol{\mu}), \quad \forall v \in V_h. \tag{4.54}$$

It is then standard to derive the error–residual relationship,

$$a(e(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = r_h(v; \boldsymbol{\mu}), \quad \forall v \in V_h. \tag{4.55}$$

For future reference we also introduce the Riesz representation of the residual,

$$R_h(\boldsymbol{\mu}) \equiv \mathcal{R}_h r_h(\cdot, \boldsymbol{\mu}), \tag{4.56}$$

for $\mathcal{R}_h$ given by (4.13); we recall from (4.14) that $\|r_h(\boldsymbol{\mu})\|_{V_h'} = \|R_h(\boldsymbol{\mu})\|_V$.

We now define our a posteriori error estimator, $\boldsymbol{\mu} \in \mathcal{P} \mapsto \Delta_N(\boldsymbol{\mu}) \in \mathbb{R}_{0+}$:

$$\Delta_N(\boldsymbol{\mu}) \equiv \frac{\|R_h(\boldsymbol{\mu})\|_V}{c_h^{\text{s,app}}(\boldsymbol{\mu})}, \tag{4.57}$$

where $c_h^{\text{s,app}}(\boldsymbol{\mu})$ is a (nonnegative) approximation to $c_h^{\text{s}}(\boldsymbol{\mu})$. It is then a simple matter to demonstrate [33] that

$$\frac{c_h^{\text{s}}(\boldsymbol{\mu})}{c_h^{\text{s,app}}(\boldsymbol{\mu})} \leq \frac{\Delta_N(\boldsymbol{\mu})}{\|e(\boldsymbol{\mu})\|_V} \leq \frac{\gamma_h(\boldsymbol{\mu})}{c_h^{\text{s,app}}(\boldsymbol{\mu})}. \tag{4.58}$$

We observe from the left inequality that, if $0 < c_h^{\text{s,app}}(\boldsymbol{\mu}) \leq c_h^{\text{s}}(\boldsymbol{\mu})$, then $\|e(\boldsymbol{\mu})\|_V \leq \Delta_N(\boldsymbol{\mu})$: Our error estimator is an error bound. We conclude from the right inequality that the error bound may overestimate the true error but by a factor which is bounded independent of N.

We note that we did not in fact use any special properties of $u_N(\boldsymbol{\mu})$ in our derivation of (4.58), and in particular we did not take advantage of the Galerkin projection. Hence, for the residual defined as (4.54) – with *unperturbed f* and *a* – our bound (4.58) in fact remains valid also for $\tilde{u}_N(\boldsymbol{\mu})$, and indeed for any function in V_h. However, we shall see that for ($\epsilon_{\text{EIM}}^{\mathcal{P}} \neq 0$ and thus certainly) *a* nonaffine we cannot compute $\|R_h(\boldsymbol{\mu})\|_V$ efficiently within the offline-online decomposition; more precisely, the operation count for evaluation of $\|R_h(\boldsymbol{\mu})\|_V$ directly as $(\mathbf{r}_h(\boldsymbol{\mu}) \mathbb{X}_h^{-1} \mathbf{r}_h(\boldsymbol{\mu}))^{1/2}$ will not be independent of N_h.[3] Furthermore, our "aggregate" error estimator (4.57) does not permit us to deduce, and hence control, the individual contributions of the RB approximation and EIM affine representation to the error $\|u_h(\boldsymbol{\mu}) - \tilde{u}_N(\boldsymbol{\mu})\|_V$. We next present an a

---

3 We note, however, that at least for problems in two space dimensions ($d = 2$), this direct evaluation though not ideal is nevertheless feasible, in particular since the parameter-independent sparse optimally ordered $\mathbb{X}_h$ can be Cholesky-factorized once.

posteriori error estimator for $\epsilon_{\text{EIM}}^{\mathcal{P}} \neq 0$ which addresses these two issues. We reinstate the $\tilde{\ }$ notation.

We first define $\tilde{e}(\boldsymbol{\mu}) \equiv u_h(\boldsymbol{\mu}) - \tilde{u}_N(\boldsymbol{\mu})$; we next introduce the perturbed residual $\boldsymbol{\mu} \mapsto \tilde{r}_h(\cdot;\boldsymbol{\mu}) \in V'$ given by

$$\tilde{r}_h(v;\boldsymbol{\mu}) \equiv \tilde{f}(v;\boldsymbol{\mu}) - \tilde{a}(\tilde{u}_N(\boldsymbol{\mu}),v;\boldsymbol{\mu}) \tag{4.59}$$

and the associated Riesz representation $\tilde{R}_h(\boldsymbol{\mu}) = \mathcal{R}_h \tilde{r}_h(\cdot,\boldsymbol{\mu})$. Our revised a posteriori error estimator is then given by

$$\tilde{\Delta}_N(\boldsymbol{\mu}) \equiv \frac{\|\tilde{R}_h(\boldsymbol{\mu})\|_V + \epsilon_{\text{EIM}}^{\mathcal{P}}(1 + \|\tilde{u}_N(\boldsymbol{\mu})\|_V)}{\tilde{c}_h^{\text{s,app}}(\boldsymbol{\mu}) - \epsilon_{\text{EIM}}^{\mathcal{P}}}, \tag{4.60}$$

where $\tilde{c}_h^{\text{s,app}}(\boldsymbol{\mu})$ is a (nonnegative) approximation to $\tilde{c}_h^{\text{s}}(\boldsymbol{\mu})$. It can be shown that, if $\epsilon_{\text{EIM}}^{\mathcal{P}} < \tilde{c}_h^{\text{s,app}}(\boldsymbol{\mu}) \leq \tilde{c}_h^{\text{s}}(\boldsymbol{\mu})$, then $\|\tilde{e}(\boldsymbol{\mu})\|_V \leq \tilde{\Delta}_N(\boldsymbol{\mu})$. We emphasize that the bound $\tilde{\Delta}_N(\boldsymbol{\mu})$ reflects – and thus can serve to efficiently control – both the RB and the EIM contributions to the error. Finally, we highlight the two constants which must be evaluated in (4.60): $\tilde{c}_h^{\text{s,app}}(\boldsymbol{\mu})$, to be discussed shortly, and $\epsilon_{\text{EIM}}^{\mathcal{P}}$, as introduced in (4.30). As regards the latter, we recall that we have direct control only over $\epsilon_{\text{EIM}}^{\Xi_{\text{EIM}}}$, and we must then assume that Ξ_{EIM} is sufficiently rich to represent $\mathcal{P}$.

We also develop a simple a posteriori error estimator for our output:

$$|s_h(\boldsymbol{\mu}) - \tilde{s}_N(\boldsymbol{\mu})| \leq (\|\tilde{\ell}(\cdot,\boldsymbol{\mu})\|_{V_h'} + \epsilon_{\text{EIM}}^{\mathcal{P}})\tilde{\Delta}_N(\boldsymbol{\mu}) + \epsilon_{\text{EIM}}^{\mathcal{P}}\|\tilde{u}_N(\boldsymbol{\mu})\|_V. \tag{4.61}$$

Note for $\epsilon_{\text{EIM}}^{\mathcal{P}} = 0$, symmetric coercive problems, and compliant outputs – $\ell(\cdot;\boldsymbol{\mu}) = f(\cdot;\boldsymbol{\mu})$ – the bound (4.61) is demonstrably pessimistic; the latter is remedied by adjoint techniques [31], [33], which also provide for better approximation of noncompliant outputs.

4.3.1.3.2 Approximate stability constant

We recall that we wish to apply the error bound in the online stage. We shall show in the next section that the dual norm of the residual, which appears in the numerators of (4.57) and (4.60), in fact admits a very efficient offline-online procedure. It remains to develop a formulation for $\tilde{c}_h^{\text{s,app}}(\boldsymbol{\mu})$ which also admits an efficient offline-online procedure and furthermore either rigorously, or plausibly, satisfies $0 < c_0 \leq \tilde{c}_h^{\text{s,app}}(\boldsymbol{\mu}) \leq \tilde{c}_h^{\text{s}}(\boldsymbol{\mu})$, $\forall \boldsymbol{\mu} \in \mathcal{P}$. There are a variety of approaches [33]. In some cases, we can explicitly deduce $\tilde{c}_h^{\text{s,app}}(\boldsymbol{\mu})$ in terms of the PDE coefficients: In Example 1.0, described by equations (4.9)–(4.11), for $c_{L^2} = 0$ in our inner product (4.1), we may choose $\tilde{c}_h^{\text{s,app}}(\boldsymbol{\mu}) = \min(1, \mu_1) \leq \alpha(\boldsymbol{\mu})$. However, for geometry variation, in particular in the vector case, inspection no longer suffices; and for the noncoercive case the situation is even more difficult. Although there are approaches which can treat the general case rigorously, such as the successive constraint method [22], these techniques are unfortunately quite complicated and often prohibitively expensive in the offline stage.

The simplest approach might be to take $\tilde{c}_h^{s,app}(\boldsymbol{\mu}) = $ (say) $\tilde{c}_N^s(\boldsymbol{\mu})/2$ [25]. If indeed $\tilde{c}_N^s(\boldsymbol{\mu})$ converges to $\tilde{c}_h^s(\boldsymbol{\mu})$ as N increases, then this simple recipe provides, at least asymptotically, a lower bound. However, the RB spaces V_N are not designed to well approximate the stability constant [39]. We thus present the obvious extension: a colateral RB approximation for the eigenproblem associated with the stability constant. Since even an order-unity error in the stability constant will yield a good error estimator and indeed a bound, even a modest RB approximation should perform very well. This approach also serves several secondary purposes relevant to this handbook: a brief summary of RB treatment of (albeit somewhat nonstandard) eigenproblems [24]; further reinforcement of RB concepts.

To begin, we recall the definition of the supremizer operators $\tilde{T}_h(\boldsymbol{\mu})$ (respectively, $\tilde{T}_N(\boldsymbol{\mu})$): For any $w \in V_h$, $\boldsymbol{\mu} \in \mathcal{P} \mapsto \tilde{T}_h(\boldsymbol{\mu}) \in \mathcal{L}(V_h, V_h)$ such that, for any $w \in V_h$, $(\tilde{T}_h(\boldsymbol{\mu})w, v)_V = \tilde{a}(w, v; \boldsymbol{\mu})$, $\forall v \in V_h$ (respectively, for any $w \in V_N$, $\boldsymbol{\mu} \in \mathcal{P} \mapsto \tilde{T}_N(\boldsymbol{\mu}) \in \mathcal{L}(V_N, V_N)$ such that, for any $w \in V_N$, $(\tilde{T}_N(\boldsymbol{\mu})w, v)_V = \tilde{a}(w, v; \boldsymbol{\mu})$, $\forall v \in V_N$). Here $\mathcal{L}(W, W)$ denotes the space of continuous mappings from W to W. We now introduce the following generalized symmetric eigenproblems: For the coercive case, find $\boldsymbol{\mu} \in \mathcal{P} \mapsto (\Psi_h(\boldsymbol{\mu}), \lambda_h(\boldsymbol{\mu})) \in V_h \times \mathbb{R}_+$ such that

$$\frac{1}{2}\big(\tilde{a}(\Psi_h(\boldsymbol{\mu}), v; \boldsymbol{\mu}) + \tilde{a}(v, \Psi_h(\boldsymbol{\mu}); \boldsymbol{\mu})\big) = \lambda_h(\boldsymbol{\mu})(\Psi_h(\boldsymbol{\mu}), v)_V, \quad \forall v \in V_h; \qquad (4.62)$$

and for the noncoercive case, find $\boldsymbol{\mu} \in \mathcal{P} \mapsto (\Phi_h(\boldsymbol{\mu}), \sigma_h^2(\boldsymbol{\mu})) \in V_h \times \mathbb{R}_+$ such that

$$(\tilde{T}_h(\boldsymbol{\mu})\Phi_h(\boldsymbol{\mu}), \tilde{T}_h(\boldsymbol{\mu})v)_V = \sigma_h^2(\boldsymbol{\mu})(\Phi_h(\boldsymbol{\mu}), v)_V, \quad \forall v \in V_h; \qquad (4.63)$$

we enumerate the modes in order of increasing magnitude of the eigenvalue. Note that the inner product constant c_{L^2} in (4.1) should be chosen large enough to ensure adequate separation of the lowest eigenvalues.[4]

It is readily demonstrated that

$$\tilde{\alpha}_h(\boldsymbol{\mu}) = (\lambda_h(\boldsymbol{\mu}))_1 = \frac{\tilde{a}((\Psi_h(\boldsymbol{\mu}))_1, (\Psi_h(\boldsymbol{\mu}))_1; \boldsymbol{\mu})}{((\Psi_h(\boldsymbol{\mu}))_1, (\Psi_h(\boldsymbol{\mu}))_1)_V} \qquad (4.64)$$

and

$$\tilde{\beta}_h(\boldsymbol{\mu}) = (\sigma_h(\boldsymbol{\mu}))_1 = \sqrt{\frac{(\tilde{T}_h(\boldsymbol{\mu})(\Phi_h(\boldsymbol{\mu}))_1, \tilde{T}_h(\boldsymbol{\mu})(\Phi_h(\boldsymbol{\mu}))_1)_V}{((\Phi_h(\boldsymbol{\mu}))_1, (\Phi_h(\boldsymbol{\mu}))_1)_V}}. \qquad (4.65)$$

We thus observe that a good approximation for the eigenfunction will yield a good approximation for the respective eigenvalue. We can now proceed to RB approximation.

---

[4] We suggest a value $c_{L^2} = \max_{\boldsymbol{\mu} \in \mathcal{P}} \max(\|\Upsilon_{00}(\cdot; \boldsymbol{\mu})\|_{L^\infty(\Omega)}, \max_{i \in \{1,2,3\}} \|\Upsilon_{ii}(\cdot; \boldsymbol{\mu})\|_{L^\infty(\Omega)}/l^2)$, where l is a characteristic minimum length scale associated with Ω.

In particular, we can envision that, just as u_h resides on a low-dimensional parametric manifold, so do $(\Psi_h)_1$ and $(\Phi_h)_1$. We may thus construct corresponding RB spaces and bases, respectively: For the coercive case, V_N^Ψ and $\mathbb{V}_N^\Psi$ for $1 \le N \le N_{\max}^\Psi$; for the inf-sup stable case, V_N^Φ and $\mathbb{V}_N^\Phi$ for $1 \le N \le N_{\max}^\Phi$. The manifolds may not be smooth for eigenproblems; however, we can accommodate mode crossing through proper choice of a sufficiently rich RB space and in particular through incorporation of the first few eigenfunctions. We note that our interest here is in the eigenvalue, not the eigenfunction, and in particular the latter serves only to develop a good approximation for the former.

The Galerkin weak statements of the RB approximations directly follow: For the coercive case, find $\boldsymbol{\mu} \in \mathcal{P} \mapsto (\Psi_N(\boldsymbol{\mu}), \lambda_N(\boldsymbol{\mu})) \in V_N^\Psi \times \mathbb{R}_+$ such that

$$\frac{1}{2}\big(\tilde{a}(\Psi_N(\boldsymbol{\mu}), v; \boldsymbol{\mu}) + \tilde{a}(v, \Psi_N(\boldsymbol{\mu}); \boldsymbol{\mu})\big) = \lambda_N(\boldsymbol{\mu})(\Psi_N(\boldsymbol{\mu}), v)_V, \quad \forall v \in V_N^\Psi, \tag{4.66}$$

and set $\tilde{c}_h^{\mathrm{s,app}}(\boldsymbol{\mu}) = (\lambda_N(\boldsymbol{\mu}))_1$; for the noncoercive case, we must consider $\boldsymbol{\mu} \in \mathcal{P} \mapsto (\Phi_N(\boldsymbol{\mu}), \sigma_N^2(\boldsymbol{\mu})) \in V_N^\Phi \times \mathbb{R}_+$ such that

$$(\tilde{T}_h(\boldsymbol{\mu})\Phi_N(\boldsymbol{\mu}), \tilde{T}_h(\boldsymbol{\mu})v)_V = \sigma_N^2(\boldsymbol{\mu})(\Phi_N(\boldsymbol{\mu}), v)_V, \quad \forall v \in V_N^\Phi, \tag{4.67}$$

and set $\tilde{c}_h^{\mathrm{s,app}}(\boldsymbol{\mu}) = (\sigma_N(\boldsymbol{\mu}))_1$. Note in (4.67) we retain $\tilde{T}_h(\boldsymbol{\mu})$, and do not substitute $\tilde{T}_N(\boldsymbol{\mu})$. In practice, we might even deflate $(\lambda_N(\boldsymbol{\mu}))_1$ and $(\sigma^N(\boldsymbol{\mu}))_1$ by some factor, say, $1/2$, to ensure that our estimates approach the true respective stability constants from below.

We also provide here the associated discrete equations: $(\boldsymbol{\Psi}_N(\boldsymbol{\mu}) \in \mathbb{R}^N, \lambda_N(\boldsymbol{\mu}) \in \mathbb{R}_+)$ satisfies

$$\underbrace{\Big(\mathbb{V}_N^{\Psi\,\mathrm{T}} \frac{1}{2}(\tilde{\mathbb{A}}_h(\boldsymbol{\mu}) + \tilde{\mathbb{A}}_h^{\mathrm{T}}(\boldsymbol{\mu})) \, \mathbb{V}_N^\Psi\Big)}_{\mathbb{E}_N^\Psi(\boldsymbol{\mu})} \boldsymbol{\Psi}_N(\boldsymbol{\mu}) = \lambda_N(\boldsymbol{\mu})(\mathbb{V}_N^{\Psi\,\mathrm{T}} \, \mathbb{X}_h \, \mathbb{V}_N^\Psi)\boldsymbol{\Psi}_N(\boldsymbol{\mu}); \tag{4.68}$$

similarly, $(\boldsymbol{\Phi}_N(\boldsymbol{\mu}) \in \mathbb{R}^N, \sigma_N^2(\boldsymbol{\mu}) \in \mathbb{R}_+)$ satisfies

$$\underbrace{\big(\mathbb{V}_N^{\Phi\,\mathrm{T}} \, (\tilde{\mathbb{A}}_h^{\mathrm{T}}(\boldsymbol{\mu})\mathbb{X}_h^{-1}\tilde{\mathbb{A}}_h(\boldsymbol{\mu})) \, \mathbb{V}_N^\Phi\big)}_{\mathbb{E}_N^\Phi(\boldsymbol{\mu})} \boldsymbol{\Phi}_N(\boldsymbol{\mu}) = \sigma_N^2(\boldsymbol{\mu})(\mathbb{V}_N^{\Phi\,\mathrm{T}} \, \mathbb{X}_h \, \mathbb{V}_N^\Phi)\boldsymbol{\Phi}_N(\boldsymbol{\mu}). \tag{4.69}$$

The RB matrices associated with these eigenproblems admit an affine decomposition: For (4.68), $\mathbb{E}_N^\Psi(\boldsymbol{\mu})$ can be expressed as

$$\mathbb{E}_N^\Psi(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu})\Big(\mathbb{V}_N^{\Psi\,\mathrm{T}} \frac{1}{2}(\mathbb{A}_h^q + \mathbb{A}_h^{q\,\mathrm{T}})\mathbb{V}_N^\Psi\Big), \tag{4.70}$$

which is the usual single-sum affine expansion; for (4.69), $\mathbb{E}_N^\Phi(\boldsymbol{\mu})$ can be expressed as

$$\mathbb{E}_N^\Phi(\boldsymbol{\mu}) = \sum_{q=1}^{Q_a}\sum_{q'=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu})\Theta_a^{q'}(\boldsymbol{\mu})(\mathbb{V}_N^{\Phi\,\mathrm{T}}(\mathbb{A}_h^{q\,\mathrm{T}}\mathbb{X}_h^{-1}\mathbb{A}_h^{q'})\mathbb{V}_N^\Phi), \tag{4.71}$$

which is now a double-sum affine expansion.

4.3.1.4 Offline-online computational procedures

We describe here the offline-online computational procedure. In the offline stage we prepare a parameter-independent data set. In the online (or deployed) stage we perform RB queries: $\boldsymbol{\mu} \mapsto (\tilde{u}_N(\boldsymbol{\mu}), \tilde{s}_N(\boldsymbol{\mu}))$. In general, the offline stage (respectively, a single online RB query) is very expensive (respectively, very inexpensive) relative to a single FE query $\boldsymbol{\mu} \mapsto \tilde{u}_h(\boldsymbol{\mu})$. As described in Section 4.1, the RB method, and in particular the offline expense, can be justified in the real-time context or the many-query context.

In Section 4.4.1 we shall describe the procedure by which we identify our RB spaces. The latter is performed as part of the offline procedure. In the current section we presume that the RB spaces are *given* in the form of the RB basis matrix $V_{N_{\max}}$. We consider here the offline-online procedure for subsequent (i) formation and solution of the RB discrete equations to obtain $\tilde{\boldsymbol{u}}_N(\boldsymbol{\mu})$ and $\tilde{s}_N(\boldsymbol{\mu})$, and (ii) evaluation of the dual norm of the residual, $\|\tilde{R}_h(\boldsymbol{\mu})\|_V$, as required by our a posteriori error indicator (4.60). More generally, the former is an example of a single-sum affine expansion, whereas the latter is an example of double-sum affine expansion. Other examples of single-sum and double-sum expansions include evaluation of $(\lambda_N(\boldsymbol{\mu}))_1$ and $(\sigma_N(\boldsymbol{\mu}))_1$, respectively, as required for the stability-constant approximation; the latter thus follow offline-online strategies very similar to those described below for the RB linear system.

4.3.1.4.1 RB linear system: formation and solution

In the offline stage, we form $\mathbb{A}_{N_{\max}}^q, 1 \le q \le Q_a$, $\boldsymbol{f}_{N_{\max}}^q, 1 \le q \le Q_f$, and $\boldsymbol{\ell}_{N_{\max}}^q, 1 \le q \le Q_\ell$ of (4.52); the operation count, taking into account FE sparsity, is $\mathcal{O}(Q_a N_{\max}^2 N_h) + \mathcal{O}(Q_f N_{\max} N_h) + \mathcal{O}(Q_\ell N_{\max} N_h)$. It is important to note that we form these matrices and vectors for the largest space, $V_{N_{\max}}$; as the RB spaces are hierarchical, we can then readily obtain the matrices (respectively, vectors) for any other RB space, V_N, by simply extracting the $N \times N$ first entries (respectively, N first entries). In the online stage, for any given N and $\boldsymbol{\mu} \in \mathcal{P}$, we first form $\tilde{\mathbb{A}}_N(\boldsymbol{\mu})$, $\tilde{\boldsymbol{f}}_N(\boldsymbol{\mu})$, and $\tilde{\boldsymbol{\ell}}_N(\boldsymbol{\mu})$ from (4.50) in $\mathcal{O}(Q_a N^2) + \mathcal{O}(Q_f N) + \mathcal{O}(Q_\ell N)$ FLOPs; we then solve (4.47) for $\boldsymbol{u}_N(\boldsymbol{\mu})$ in $\mathcal{O}(N^3)$ FLOPs – in general, $A_N(\boldsymbol{\mu})$ shall be a dense matrix; finally, we evaluate $s_N(\boldsymbol{\mu})$ from (4.48) in $\mathcal{O}(N)$ FLOPs. Note that if we wish to visualize the full field, then an additional $\mathcal{O}(NN_h)$ FLOPs are required to evaluate the FE basis coefficients of the RB approximation as $\mathbb{V}_N \boldsymbol{u}_N(\boldsymbol{\mu})$; the online operation count is independent of N_h except for this (elective) full field reconstruction. Note however that, if the visualization of each RB function is stored and prepared offline as a frame on a GPU, only the linear combination of these frames is required and we are back to an $\mathcal{O}(N)$ complexity.

4.3.1.4.2 Residual dual norm evaluation

To begin, we form the FE representation of $\tilde{r}_h(\cdot; \boldsymbol{\mu})$, $\tilde{r}_h(\varphi^j; \boldsymbol{\mu})$, $1 \le j \le N_h$:

$$\tilde{\mathbf{r}}_h(\boldsymbol{\mu}) = \tilde{\boldsymbol{f}}_h(\boldsymbol{\mu}) - \tilde{\mathbb{A}}_h(\boldsymbol{\mu}) \mathbb{V}_N \boldsymbol{u}_N(\boldsymbol{\mu}). \tag{4.72}$$

It is then readily demonstrated from $\tilde{R}_h(\boldsymbol{\mu}) = \mathcal{R}_h \tilde{r}_h(\cdot; \boldsymbol{\mu})$ and (4.14) that the dual norm of the residual, hence $\|\tilde{R}_h(\boldsymbol{\mu})\|_V$, is given by

$$\|\tilde{R}_h(\boldsymbol{\mu})\|_V = (\tilde{\boldsymbol{f}}_h(\boldsymbol{\mu}) - \tilde{\mathbb{A}}_h(\boldsymbol{\mu}) \mathbb{V}_N \boldsymbol{u}_N(\boldsymbol{\mu}))^{\mathrm{T}} \mathbb{X}_h^{-1} (\tilde{\boldsymbol{f}}_N(\boldsymbol{\mu}) - \tilde{\mathbb{A}}_h(\boldsymbol{\mu}) \mathbb{V}_N \boldsymbol{u}_N(\boldsymbol{\mu})). \tag{4.73}$$

We now introduce $\boldsymbol{\Theta}_N \in \mathbb{R}^{Q_f + Q_a N}$ as

$$\boldsymbol{\Theta}_N(\boldsymbol{\mu}) \equiv (\Theta_f^1(\boldsymbol{\mu}), \ldots, \Theta_f^{Q_f}(\boldsymbol{\mu}), \Theta_a^1(\boldsymbol{\mu})(\boldsymbol{u}_N(\boldsymbol{\mu}))_1, \ldots, \Theta_a^1(\boldsymbol{\mu})(\boldsymbol{u}_N(\boldsymbol{\mu}))_N,$$
$$\ldots, \Theta_a^{Q_a}(\boldsymbol{\mu})(\boldsymbol{u}_N(\boldsymbol{\mu}))_1, \ldots, \Theta_a^{Q_a}(\boldsymbol{\mu})(\boldsymbol{u}_N(\boldsymbol{\mu}))_N)^{\mathrm{T}}, \tag{4.74}$$

and also $\boldsymbol{L}_N \in \mathbb{R}^{N_h \times (Q_f + Q_a N)}$ as

$$\boldsymbol{L}_N \equiv (\underbrace{\boldsymbol{f}_h^1}_{N_h \times 1} \mid \ldots \mid \boldsymbol{f}_h^{Q_f} \mid \underbrace{-\tilde{\mathbb{A}}_h^1 \mathbb{V}_N}_{N_h \times N} \mid \ldots \mid -\tilde{\mathbb{A}}_h^{Q_a} \mathbb{V}_N); \tag{4.75}$$

note $\boldsymbol{L}_N$ is expressed in block column form with | as block delimiters. We now combine (4.73), the affine representations of the FE operators, (4.27), (4.74), and (4.75) to obtain

$$\|\tilde{R}_h(\boldsymbol{\mu})\|_V = (\boldsymbol{\Theta}_N^{\mathrm{T}}(\boldsymbol{\mu}) \underbrace{\boldsymbol{L}_N^{\mathrm{T}} \mathbb{X}_h^{-1} \boldsymbol{L}_N}_{W_N} \boldsymbol{\Theta}_N(\boldsymbol{\mu}))^{1/2}; \tag{4.76}$$

note that $W_N \in \mathbb{R}^{(Q_f + Q_a N) \times (Q_f + Q_a N)}$. We can now describe the offline-online decomposition.

In the offline stage we form $W_{N_{\max}}$. In the (say) direct-solution context, we would perform a sparse Cholesky of (optimally ordered) $\mathbb{X}_h$ once. We would then perform $Q_f + Q_a N_{\max}$ forward/back substitutions to find $\boldsymbol{L}'_{N_{\max}} \equiv \mathbb{X}_h^{-1} \boldsymbol{L}_{N_{\max}}$; we would then complete the formation of $W_{N_{\max}}$ by matrix multiplication, $\boldsymbol{L}_{N_{\max}}^{\mathrm{T}} \boldsymbol{L}'_{N_{\max}}$, at cost $\mathcal{O}((Q_f + Q_a N)^2 N_h)$ FLOPs. Note that $W_{N_{\max}}$ is *parameter-independent*. In the online stage, given $\boldsymbol{\mu} \in \mathcal{P}$ and our associated RB approximation $\boldsymbol{u}_N(\boldsymbol{\mu})$, we first extract submatrix W_N from $W_{N_{\max}}$ and evaluate $\boldsymbol{\Theta}_N(\boldsymbol{\mu})$. We then perform the sum (4.76), $(\boldsymbol{\Theta}_N^{\mathrm{T}}(\boldsymbol{\mu}) W_N \boldsymbol{\Theta}_N(\boldsymbol{\mu}))^{1/2}$. The operation count is $\mathcal{O}((Q_f + Q_a N)^2)$ FLOPs, which we note is independent of N_h.[5] We observe that

---

5 We can now readily define minimum-residual projection: Find $\bar{\boldsymbol{u}}_N^*(\boldsymbol{\mu})$ and hence $\boldsymbol{\Theta}^*(\boldsymbol{\mu})$ which minimizes $\|\tilde{R}_h(\boldsymbol{\mu})\|_V$. It follows from our offline-online discussion that minimum-residual projection will be more expensive than Galerkin projection but only as regards formation of the RB linear system and in particular for larger Q_f, Q_a.

the operation count scales quadratically with both Q_f, Q_a and also N, which emphasizes the important role of a posteriori error estimation to control both the RB but also the EIM costs.

Finally, we note one shortcoming of the offline-online approach. Let us denote machine precision by ϵ_{prec}. The construction (4.76) computes a small number, $\|\tilde{R}_h(\boldsymbol{\mu})\|_V$, as the square root of the cancellation of (many) large summands. To illustrate the difficulty, consider $\epsilon_{\text{prec}} = 1 \times 10^{-16}$, assume $\|\tilde{R}_h(\boldsymbol{\mu})\|_V = 1 \times 10^{-10}$, and furthermore say (for simplicity) that $\boldsymbol{\Theta}_N^{\mathsf{T}}(\boldsymbol{\mu}) W_N \boldsymbol{\Theta}_N(\boldsymbol{\mu})$ is the sum of just two terms, respectively $1 + \frac{1}{2} \times 10^{-20}$ and $-1 + \frac{1}{2} \times 10^{-20}$. Clearly upon truncation in finite precision we obtain for the sum (respectively, the square root of the sum) not 10^{-20} (respectively, 10^{-10}) but rather 0. This finite-precision effect can in principle compromise numerical convergence tests for sufficiently high accuracy, and remedies are proposed in the literature [11, 4]. However, for engineering calculations, the limitation is not significant: It will very rarely, if ever, be the case that the data for an engineering problem are known to sufficient digits to warrant a numerical error as small as 1×10^{-8}.

4.3.2 Extensions

4.3.2.1 Evolution problems: parabolic PDEs

4.3.2.1.1 Galerkin projection

As for elliptic PDEs, we are given a hierarchical set of RB spaces $\{V_N\}_{N=1,\dots,N_{\max}}$. We are further provided, for $1 \le N \le N_{\max}$, with an associated basis $\{\zeta_h^n\}_{n=1,\dots,N}$, and corresponding basis matrix $\mathbb{V}_N \in \mathbb{R}^{N_h \times N}$, $(\mathbb{V}_N)_{kn} = (\boldsymbol{\zeta}_h^n)_k$, $1 \le k \le N_h$, $1 \le n \le N$. The method by which we shall develop $V_{N_{\max}}$ for the parabolic case [20], related to but also different from the method with which we identify $V_{N_{\max}}$ for the elliptic case, shall be summarized in Section 4.4.3.1. We note here only that the construction shall ensure that u_0 (our initial condition) resides in V_N.

We can now state the Galerkin projection [19, 20]: Given $N \in \{1,\dots,N_{\max}\}$ and $\boldsymbol{\mu} \in \mathcal{P}$, we look for $\tilde{u}_{N,\Delta t}^j(\boldsymbol{\mu})(\approx u_h(t^j;\boldsymbol{\mu})) \in V_N$, $j = 1,\dots,J$, such that

$$\tilde{m}\left(\frac{\tilde{u}_{N,\Delta t}^j(\boldsymbol{\mu}) - \tilde{u}_{N,\Delta t}^{j-1}(\boldsymbol{\mu})}{\Delta t}, v;\boldsymbol{\mu}\right) + \tilde{a}(\tilde{u}_{N,\Delta t}^j(\boldsymbol{\mu}), v;\boldsymbol{\mu}) = \tau(t^j)\tilde{f}(v;\boldsymbol{\mu}), \quad \forall v \in V_N; \qquad (4.77)$$

we impose the initial condition $\tilde{u}_{N,\Delta t}^{j=0}(\boldsymbol{\mu}) = u_0 \ (\in V_N$, by construction). We note that in (4.77) there is no reduction in the temporal dimension: the RB projection (4.77) retains the "true" finite difference discretization of the FE projection (4.39); the RB acceleration is effected solely through the dimension reduction in the spatial dimension.

To develop the discrete equations we require the RB mass matrix, $\tilde{\mathbb{M}}_N(\boldsymbol{\mu}) \in \mathbb{R}^{N \times N}$,

$$\left(\tilde{\mathbb{M}}_N(\boldsymbol{\mu})\right)_{kn} = \tilde{m}(\zeta_h^n, \zeta_h^k;\boldsymbol{\mu}), \quad 1 \le k, n \le N, \qquad (4.78)$$

which we may form as

$$\tilde{\mathbb{M}}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_m} \Theta_m^q(\boldsymbol{\mu}) \mathbb{M}_N^q, \tag{4.79}$$

for $\mathbb{M}_N^q = m^q(\zeta_h^n, \zeta_h^k), 1 \le k, n \le N, 1 \le q \le Q_m$. We may directly formulate our RB mass matrix and constituents in terms of the corresponding FE quantities and our basis matrix:

$$\tilde{\mathbb{M}}_N(\boldsymbol{\mu}) = \mathbb{V}_N^T \tilde{\mathbb{M}}_h(\boldsymbol{\mu}) \mathbb{V}_N, \quad \mathbb{M}_N^q = \mathbb{V}_N^T \mathbb{M}_h^q \mathbb{V}_N, \quad 1 \le q \le Q_m. \tag{4.80}$$

We can then state the discrete equations to be solved at each time t^j:

$$\underbrace{\left(\tilde{\mathbb{A}}_N(\boldsymbol{\mu}) + \frac{1}{\Delta t} \tilde{\mathbb{M}}_N(\boldsymbol{\mu}) \right)}_{\tilde{\mathbb{H}}_N(\boldsymbol{\mu})} \tilde{\boldsymbol{u}}_{N,\Delta t}^j(\boldsymbol{\mu}) = \frac{1}{\Delta t} \tilde{\mathbb{M}}_N(\boldsymbol{\mu}) \boldsymbol{u}_{N,\Delta t}^{j-1}(\boldsymbol{\mu}) + \tau(t^j) \tilde{\boldsymbol{f}}_N(\boldsymbol{\mu}). \tag{4.81}$$

Note that, in general for RB methods, we prefer implicit temporal discretizations, since inversion of the small RB discrete operators is relatively inexpensive (and furthermore the RB mass matrix is not close to diagonal in particular given our V-orthonormalization of the basis) and implicit methods allow larger time steps.

4.3.2.1.2 Error estimation

We directly consider the more practically important case of a posteriori error estimation. We shall provide here an estimator for the case in which $\epsilon_{\mathrm{EIM}}^{\mathcal{P}} = 0$ and hence we shall suppress the $\tilde{}$ for the purposes of this analysis; extension to $\epsilon_{\mathrm{EIM}}^{\mathcal{P}} \ne 0$ is not difficult. We first introduce the error $e^j(\boldsymbol{\mu}) \equiv u_{h,\Delta t}^j - u_{N,\Delta t}^j(\boldsymbol{\mu}), 1 \le j \le J$. We next define, for $1 \le j \le J$, the residual $\boldsymbol{\mu} \mapsto r_{h,\Delta t}^j(\cdot; \boldsymbol{\mu}) \in V'$,

$$r_{h,\Delta t}^j(v; \boldsymbol{\mu}) \equiv \tau(t^j) \tilde{f}(v; \boldsymbol{\mu}) - m\left(\frac{u_{N,\Delta t}^j(\boldsymbol{\mu}) - u_{N,\Delta t}^{j-1}(\boldsymbol{\mu})}{\Delta t}, v; \boldsymbol{\mu} \right) - a(u_{N,\Delta t}^j(\boldsymbol{\mu}), v; \boldsymbol{\mu}), \quad \forall v \in V_h, \tag{4.82}$$

and associated Riesz representation, $R_{h,\Delta t}^j(\boldsymbol{\mu}) \equiv \mathcal{R}_h r_{h,\Delta t}^j(\cdot; \boldsymbol{\mu})$. We then define, for $1 \le j \le J$, our error estimator $\Delta_{N,\Delta t}^j(\boldsymbol{\mu})$:

$$\Delta_{N,\Delta t}^j(\boldsymbol{\mu}) \equiv \left(\frac{\Delta t}{c_h^{\mathrm{s,app}}(\boldsymbol{\mu})} \sum_{j'=1}^j \| R_{h,\Delta t}^{j'} \|_V^2 \right)^{1/2}, \tag{4.83}$$

where $c_h^{\mathrm{s,app}}(\boldsymbol{\mu})$ is an approximation to $\alpha_h(\boldsymbol{\mu})$, for example as developed in Section 4.3.1.3. It can then be shown [19] that, if $c_h^{\mathrm{s,app}}(\boldsymbol{\mu}) \le \alpha_h(\boldsymbol{\mu})$, then

$$\left(m(e^j(\boldsymbol{\mu}), e^j(\boldsymbol{\mu}); \boldsymbol{\mu}) + c_h^{\mathrm{s,app}}(\boldsymbol{\mu}) \sum_{j'=1}^j \| e^{j'}(\boldsymbol{\mu}) \|_V^2 \right)^{1/2} \le \Delta_{N,\Delta t}^j(\boldsymbol{\mu}), \quad j = 1, \ldots, J. \tag{4.84}$$

We note that, consistent with the continuous problem formulation, we control both the (discrete-time) $C^0((0,T]; L^2(\Omega))$ error and the $L^2((0,T]; V)$-error.

4.3.2.1.3 Offline-online computational procedures

As for the elliptic case, we presume here that the RB basis matrix, $V_{N_{\max}}$, is provided; we discuss the associated procedure, part of the offline stage, in Section 4.4.3.1.

We first consider the formation and solution of the RB linear system. In the offline stage, we form $\mathbb{A}^q_{N_{\max}}, 1 \le q \le Q_a$, $\mathbb{M}^q_{N_{\max}}, 1 \le q \le Q_m$, $\boldsymbol{f}^q_{N_{\max}}, 1 \le q \le Q_f$, and $\boldsymbol{\ell}^q_{N_{\max}}, 1 \le q \le Q_\ell$ of (4.52) and (4.80); the operation count, taking into account FE sparsity, is $\mathcal{O}((Q_a + Q_m)N^2_{\max}N_h) + \mathcal{O}(Q_f N_{\max} N_h) + \mathcal{O}(Q_\ell N_{\max} N_h)$. In the online stage, for any given N and $\boldsymbol{\mu} \in \mathcal{P}$, we first form $\tilde{\mathbb{A}}_N(\boldsymbol{\mu}), \tilde{\mathbb{M}}_N(\boldsymbol{\mu}), \tilde{\mathbb{H}}_N(\boldsymbol{\mu}), \tilde{\boldsymbol{f}}_N(\boldsymbol{\mu})$, and $\tilde{\boldsymbol{\ell}}_N(\boldsymbol{\mu})$ from (4.50) and (4.79) in $\mathcal{O}((Q_a + Q_m)N^2) + \mathcal{O}(Q_f N) + \mathcal{O}(Q_\ell N)$ FLOPs; we next perform the LU decomposition of $\tilde{\mathbb{H}}_N(\boldsymbol{\mu})$, once, at cost $\mathcal{O}(N^3)$; we then solve (4.81) as J forward/back substitutions at cost $\mathcal{O}(JN^2)$; finally, we evaluate $s_N(\boldsymbol{\mu})$ in $\mathcal{O}(N)$ FLOPs. Note that we take advantage here of the linear *time-invariant* nature of our operator: For J not too small, we expect the online cost of the J parabolic updates to scale as $\mathcal{O}(JN^2)$ FLOPs, hence roughly independent of Q. and only quadratically with N.

We next turn to the evaluation of the error bound, and in particular the contribution of the dual norm of the residual. We now introduce $\boldsymbol{\Theta}^j_{N,\Delta t} \in \mathbb{R}^{Q_f + Q_m N + Q_a N}, 1 \le j \le J$, as

$$\boldsymbol{\Theta}^j_{N,\Delta t}(\boldsymbol{\mu}) \equiv \Big(\Theta^1_f(\boldsymbol{\mu}), \dots, \Theta^{Q_f}_f(\boldsymbol{\mu}), $$

$$\Theta^1_m(\boldsymbol{\mu}) \frac{(\boldsymbol{u}^j_{N,\Delta t}(\boldsymbol{\mu}))_1 - (\boldsymbol{u}^{j-1}_{N,\Delta t}(\boldsymbol{\mu}))_1}{\Delta t}, \dots, \Theta^1_m(\boldsymbol{\mu}) \frac{(\boldsymbol{u}^j_{N,\Delta t}(\boldsymbol{\mu}))_N - (\boldsymbol{u}^{j-1}_{N,\Delta t}(\boldsymbol{\mu}))_N}{\Delta t}, \dots,$$

$$\Theta^{Q_m}_m(\boldsymbol{\mu}) \frac{(\boldsymbol{u}^j_{N,\Delta t}(\boldsymbol{\mu}))_1 - (\boldsymbol{u}^{j-1}_{N,\Delta t}(\boldsymbol{\mu}))_1}{\Delta t}, \dots, \Theta^{Q_m}_m(\boldsymbol{\mu}) \frac{(\boldsymbol{u}^j_{N,\Delta t}(\boldsymbol{\mu}))_N - (\boldsymbol{u}^{j-1}_{N,\Delta t}(\boldsymbol{\mu}))_N}{\Delta t},$$

$$\Theta^1_a(\boldsymbol{\mu})(\boldsymbol{u}^j_{N,\Delta t}(\boldsymbol{\mu}))_1, \dots, \Theta^1_a(\boldsymbol{\mu})(\boldsymbol{u}^j_{N,\Delta t}(\boldsymbol{\mu}))_N, \dots,$$

$$\Theta^{Q_a}_a(\boldsymbol{\mu})(\boldsymbol{u}^j_{N,\Delta t}(\boldsymbol{\mu}))_1, \dots, \Theta^{Q_a}_a(\boldsymbol{\mu})(\boldsymbol{u}^j_{N,\Delta t}(\boldsymbol{\mu}))_N \Big)^{\mathrm{T}}, \tag{4.85}$$

and also $\boldsymbol{L}_{N,\Delta t} \in \mathbb{R}^{N_h \times (Q_f + (Q_m + Q_a)N)}$ as

$$\boldsymbol{L}_{N,\Delta t} \equiv (\boldsymbol{f}^1_h | \dots | \boldsymbol{f}^{Q_f}_h | - \mathbb{M}^1_h \mathbb{V}_N | \dots | - \mathbb{M}^{Q_m}_h \mathbb{V}_N | - \tilde{\mathbb{A}}^1_h \mathbb{V}_N | \dots | - \tilde{\mathbb{A}}^{Q_a}_h \mathbb{V}_N); \tag{4.86}$$

note $\boldsymbol{L}_{N,\Delta t}$ is expressed in block column form with | as block delimiters. We now combine (4.73), the affine representations of the FE operators, (4.27), (4.85), and (4.86) to obtain

$$\|\tilde{R}^j_{h,\Delta t}(\boldsymbol{\mu})\|_V = \big((\boldsymbol{\Theta}^j_{N,\Delta t})^{\mathrm{T}}(\boldsymbol{\mu}) \underbrace{\boldsymbol{L}^{\mathrm{T}}_{N,\Delta t} \mathbb{X}^{-1}_h \boldsymbol{L}_{N,\Delta t}}_{W_{N,\Delta t}} \boldsymbol{\Theta}^j_{N,\Delta t}(\boldsymbol{\mu})\big)^{1/2}; \tag{4.87}$$

note that $W_{N,\Delta t} \in \mathbb{R}^{(Q_f + (Q_m + Q_a)N) \times (Q_f + (Q_m + Q_a)N)}$.

In the offline stage we form $W_{N_{\max},\Delta t}$. In the (say) direct-solution context, we would perform a sparse Cholesky of (optimally ordered) $\mathbb{X}_h$ once. We would then perform $Q_f + (Q_m + Q_a)N_{\max}$ forward/back substitutions to find $\boldsymbol{L}'_{N_{\max},\Delta t} \equiv \mathbb{X}_h^{-1} \boldsymbol{L}_{N_{\max},\Delta t}$; we would then complete the formation of $W_{N_{\max},\Delta t}$ by multiplication, $\boldsymbol{L}_{N_{\max},\Delta t}^{\mathrm{T}} \boldsymbol{L}'_{N_{\max},\Delta t}$, at cost $\mathcal{O}((Q_f + (Q_m + Q_a)N)^2 N_h)$ FLOPs. Note that $W_{N_{\max},\Delta t}$ is *parameter-independent*. In the online stage, given $\boldsymbol{\mu} \in \mathcal{P}$ and our associated RB approximation $\boldsymbol{u}^j_{N,\Delta t}(\boldsymbol{\mu})$, $1 \le j \le J$, we first extract submatrix $W_{N,\Delta t}$ from $W_{N_{\max},\Delta t}$ and evaluate $\boldsymbol{\Theta}^j_{N,\Delta t}(\boldsymbol{\mu})$, $1 \le j \le J$. We then perform the sum (4.87), $((\boldsymbol{\Theta}^j_{N,\Delta t}(\boldsymbol{\mu}))^{\mathrm{T}} W_{N,\Delta t} \boldsymbol{\Theta}^j_{N,\Delta t}(\boldsymbol{\mu}))^{1/2}$, $1 \le j \le J$; the operation count (for given j) is $\mathcal{O}((Q_f + (Q_m + Q_a)N)^2)$ FLOPs, which we note again is independent of N_h.

4.3.2.2 A nonlinear elliptic problem

4.3.2.2.1 Galerkin projection

As for linear elliptic problems, we are given a hierarchical set of RB spaces $\{V_N \subset V_h\}_{N=1,\dots,N_{\max}}$ and associated basis $\{\zeta_h^i\}_{i=1,\dots,N_{\max}}$. We may then define our RB-Galerkin approximation for (4.41): Given $N \in \{1, \dots, N_{\max}\}$ and $\boldsymbol{\mu} \in \mathcal{P}$, find $\tilde{u}_N(\boldsymbol{\mu}) \in V_N$ such that

$$\sum_{j=1}^{N^{\mathrm{quad},\Omega}} \rho_j^{\mathrm{quad},\Omega} \nabla \tilde{u}_N(x_j^{\mathrm{quad},\Omega}; \boldsymbol{\mu}) \cdot \nabla v(x_j^{\mathrm{quad},\Omega})$$

$$+ \sum_{m,m'=1}^{M} (B_M^{-1})_{mm'} \eta(\tilde{u}_N(x_{i_{m'}^*}^{\mathrm{quad},\Omega}, \boldsymbol{\mu})) \left[\sum_{j=1}^{N^{\mathrm{quad},\Omega}} (\boldsymbol{\xi}^m)_j \, v(x_j^{\mathrm{quad},\Omega}) \rho_j^{\mathrm{quad},\Omega} \right]$$

$$= \boldsymbol{\mu} \sum_{j=1}^{N^{\mathrm{quad},\Omega}} \rho_j^{\mathrm{quad},\Omega} v(x_j^{\mathrm{quad},\Omega}), \quad \forall v \in V_N, \tag{4.88}$$

where the EIM interpolation system is defined in Section 4.2.3.3 (recall (4.44)). We may further introduce the Newton update equation associated with (4.88): For given current iterate $\tilde{u}_N^k(\boldsymbol{\mu}) \in V_N$, find $\delta \tilde{u}_N(\boldsymbol{\mu}) \in V_N$ such that

$$\sum_{j=1}^{N^{\mathrm{quad},\Omega}} \rho_j^{\mathrm{quad},\Omega} \nabla \delta \tilde{u}_N(x_j^{\mathrm{quad},\Omega}; \boldsymbol{\mu}) \cdot \nabla v(x_j^{\mathrm{quad},\Omega})$$

$$+ \sum_{m,m'=1}^{M} (B_M^{-1})_{mm'} \dot{\eta}(\tilde{u}_N^k(x_{i_{m'}^*}^{\mathrm{quad},\Omega}, \boldsymbol{\mu})) \, \delta \tilde{u}_N^k(x_{i_{m'}^*}^{\mathrm{quad},\Omega}; \boldsymbol{\mu}) \left[\sum_{j=1}^{N^{\mathrm{quad},\Omega}} (\boldsymbol{\xi}^m)_j \, v(x_j^{\mathrm{quad},\Omega}) \rho_j^{\mathrm{quad},\Omega} \right]$$

$$= \tilde{r}_h^k(v; \boldsymbol{\mu}), \quad \forall v \in V_N. \tag{4.89}$$

Note $\dot{\eta}$ is the derivative of η (we reserve prime for dummy indices): For example, for $\eta(z) = z^3$, $\dot{\eta}(z) = 3z^2$, and for $\eta(z) = |z|z$, $\dot{\eta}(z) = 2|z|$; in general, $\dot{\eta} > 0$ from our

assumption of monotonic nondecreasing η. The right-hand side of (4.89), residual $\boldsymbol{\mu} \in \mathcal{P} \mapsto \tilde{r}_h^k(\cdot\,;\boldsymbol{\mu}) \in V_h'$, is defined as

$$
\begin{aligned}
\tilde{r}_h^k(v;\boldsymbol{\mu}) \equiv\ & \boldsymbol{\mu} \sum_{j=1}^{N^{\text{quad},\Omega}} \rho_j^{\text{quad},\Omega} v(x_j^{\text{quad},\Omega}) \\
& - \sum_{j=1}^{N^{\text{quad},\Omega}} \rho_j^{\text{quad},\Omega} \nabla \tilde{u}_N^k(x_j^{\text{quad},\Omega};\boldsymbol{\mu}) \cdot \nabla v(x_j^{\text{quad},\Omega}) \\
& - \sum_{m,m'=1}^{M} (B_M^{-1})_{mm'}\, \eta(\tilde{u}_N^k(x_{i^*_{m'}}^{\text{quad},\Omega},\boldsymbol{\mu})) \left[\sum_{j=1}^{N^{\text{quad},\Omega}} (\boldsymbol{\xi}^m)_j\, v(x_j^{\text{quad},\Omega})\rho_j^{\text{quad},\Omega} \right], \quad \forall v \in V_N.
\end{aligned}
$$

$$(4.90)$$

The left-hand side of (4.89) is the Gâteaux derivative of our RB nonlinear operator.

We now proceed to the discrete equations. We first introduce parameter-independent matrices and vectors $\mathbb{B}_N \in \mathbb{R}^{N \times N}$, $\mathbb{C}_N \in \mathbb{R}^{M \times N}$, and $\boldsymbol{f}_N^1 \in \mathbb{R}^N$ given by

$$
(\mathbb{B}_N)_{nn'} \equiv \sum_{j=1}^{N^{\text{quad},\Omega}} \rho_j^{\text{quad},\Omega}\, \nabla \zeta_h^{n'}(x_j^{\text{quad},\Omega}) \cdot \nabla \zeta_h^n(x_j^{\text{quad},\Omega}), \quad 1 \le n, n' \le N, \tag{4.91}
$$

$$
(\mathbb{C}_N)_{m'n} \equiv \sum_{m=1}^{M} (B_M^{-1})_{mm'} \sum_{j=1}^{N^{\text{quad},\Omega}} (\boldsymbol{\xi}^m)_j\, \zeta_h^n(x_j^{\text{quad},\Omega})\rho_j^{\text{quad},\Omega}, \quad 1 \le m' \le M, 1 \le n \le N, \tag{4.92}
$$

$$
(\boldsymbol{f}_N^1)_n \equiv \sum_{j=1}^{N^{\text{quad},\Omega}} \rho_j^{\text{quad},\Omega}\zeta_h^n(x_j^{\text{quad},\Omega}), \tag{4.93}
$$

respectively. Note that $\mathbb{B}_N$, $\mathbb{C}_N$, and $\boldsymbol{f}_N^1$ are all parameter-independent.

Our Newton update may then be expressed as

$$
\begin{aligned}
\sum_{n'=1}^{N} (\mathbb{J}_N^k(\boldsymbol{\mu}))_{nn'} (\delta \boldsymbol{u}_N^k(\boldsymbol{\mu}))_{n'} = \boldsymbol{\mu}(\boldsymbol{f}_N^1)_n &- \sum_{n'=1}^{N} (\mathbb{B}_N)_{nn'} (\boldsymbol{u}_N^k(\boldsymbol{\mu}))_{n'} \\
&- \sum_{m'=1}^{M} \eta_{Nm'}^k(\boldsymbol{\mu})\, (\mathbb{C}_N)_{m'n}, \quad 1 \le n \le N,
\end{aligned}
$$

$$(4.94)$$

where the Jacobian matrix $\mathbb{J}_N^k(\boldsymbol{\mu}) \in \mathbb{R}^{N \times N}$ is given by

$$
(\mathbb{J}_N^k(\boldsymbol{\mu}))_{nn'} \equiv (\mathbb{B}_N)_{nn'} + \sum_{m'=1}^{M} \dot{\eta}_{Nm'}^k(\boldsymbol{\mu})(\mathbb{C}_N)_{m'n}\, \zeta_h^{n'}(x_{i^*_{m'}}^{\text{quad},\Omega}), \quad 1 \le n, n' \le N. \tag{4.95}
$$

Here $\eta_{Nm'}^k$, $\dot{\eta}_{Nm'}^k$, $1 \le m' \le M$, are given by

$$
\eta_{Nm'}^k(\boldsymbol{\mu}) = \eta\left(\sum_{n=1}^{N} (\tilde{\boldsymbol{u}}_N^k(\boldsymbol{\mu}))_n\, \zeta_h^n(x_{i^*_{m'}}^{\text{quad},\Omega}) \right), \quad 1 \le m' \le M, \tag{4.96}
$$

$$\dot{\eta}^k_{N\,m'}(\boldsymbol{\mu}) = \dot{\eta}\left(\sum_{n=1}^{N} (\bar{\boldsymbol{u}}^k_N(\boldsymbol{\mu}))_n \zeta^n_h(x^{\mathrm{quad},\Omega}_{i^*_{m'}}) \right), \quad 1 \le m' \le M. \tag{4.97}$$

We note that for $\eta(z) = z^3$ it can be shown that $\mathbb{J}^k_N(\boldsymbol{\mu})$ of (4.95) is symmetric positive-definite for $\epsilon^{\mathrm{NL}}_{\mathrm{EIM}}$ sufficiently small.

4.3.2.2.2 Error estimation

We directly consider the more practically important case of a posteriori error estimation. We shall continue to assume (effectively) exact quadrature. By way of preliminaries, we define the Laplacian bilinear form $b : V \times V \to \mathbb{R}$,

$$b(w,v) = \int_{\Omega} \nabla w \cdot \nabla v, \quad \forall w, v \in V^2. \tag{4.98}$$

We then define (for consistency with earlier parts of the chapter)

$$\tilde{c}^s_h(\boldsymbol{\mu}) \equiv \inf_{v \in V_h} \frac{|b(v,v)|}{\|v\|_V}; \tag{4.99}$$

in the current context, $\tilde{c}^s_h(\boldsymbol{\mu})$ is a coercivity constant which is in fact parameter-independent.

We now define our error as $\tilde{e}^k(\boldsymbol{\mu}) \equiv u_h(\boldsymbol{\mu}) - \tilde{u}^k_N(\boldsymbol{\mu})$. We further introduce the Riesz representation of our residual, $\tilde{R}^k_h = \mathcal{R}_h \tilde{r}^k_h(\cdot; \boldsymbol{\mu})$ for $\tilde{r}^k_h(\cdot; \boldsymbol{\mu})$ defined in (4.90). We may then define our a posteriori error estimator,

$$\tilde{\Delta}^k_N(\boldsymbol{\mu}) = \frac{\|\tilde{R}^k_h(\boldsymbol{\mu})\|_V + c^{-1/2}_{L^2} |\Omega|^{1/2} \mathrm{tol}^{\mathrm{NL}}_{\mathrm{EIM}}}{\tilde{c}^{s,\mathrm{app}}_h(\boldsymbol{\mu})}. \tag{4.100}$$

Here c_{L^2} is the weight of the $L^2(\Omega)$-contribution to our norm $\|\cdot\|_V$, $|\Omega|$ is the measure of our domain in $\mathbb{R}^d$, and $\tilde{c}^{s,\mathrm{app}}_h(\boldsymbol{\mu})$ is an approximation to the coercivity constant $\tilde{c}^s_h(\boldsymbol{\mu})$, (4.99), as developed in Section 4.3.1.3. We can then show that, for any $\boldsymbol{\mu} \in \Xi^{\mathrm{NL}}_{\mathrm{EIM}}$, and $\tilde{c}^{s,\mathrm{app}}_h(\boldsymbol{\mu}) \le \tilde{c}^s_h(\boldsymbol{\mu})$,

$$\|\tilde{e}^k(\boldsymbol{\mu})\|_V \le \tilde{\Delta}^k_N(\boldsymbol{\mu}); \tag{4.101}$$

we can plausibly extend our result to any $\boldsymbol{\mu} \in \mathcal{P}$, perhaps with a safety factor, depending on the richness of $\Xi^{\mathrm{NL}}_{\mathrm{EIM}}$. The a posteriori error bound (4.100), (4.101), identifies three contributions to the error: the (incomplete) convergence of the Newton iteration, as reflected in superscript k; the RB approximation error, as reflected in the dual norm of the residual; and the perturbation of our problem to affine form, as reflected in the EIM tolerance parameter. We emphasize that, within this nonlinear iterative context, the error bound can serve not only to assess and control RB and EIM errors but also as an iterative termination criterion.

We briefly derive this result in particular to emphasize the simplification afforded by our monotonic nondecreasing nonlinearity and conversely the challenges associated with more difficult nonlinearities. We first note that (under our assumption of exact quadrature) our residual may be expressed as

$$\tilde{r}_h^k(v; \boldsymbol{\mu}) \equiv \boldsymbol{\mu} \int_\Omega v - b(\tilde{u}_N^k(\boldsymbol{\mu}), v) - \int_\Omega \mathcal{I}_M[\eta(\tilde{u}_N^k(\cdot; \boldsymbol{\mu}))]v, \quad \forall v \in V_h. \tag{4.102}$$

We also note that $u_h(\boldsymbol{\mu})$ satisfies

$$b(u_h(\boldsymbol{\mu}), v) + \int_\Omega \eta(u_h(\boldsymbol{\mu}))v = \boldsymbol{\mu} \int_\Omega v, \quad \forall v \in V_h. \tag{4.103}$$

It thus follows that

$$b(u_h(\boldsymbol{\mu}) - \tilde{u}_N^k(\boldsymbol{\mu}), v) + \int_\Omega [\eta(u_h(\boldsymbol{\mu})) - \eta(\tilde{u}_N^k(\boldsymbol{\mu}))]v$$
$$= \tilde{r}_h^k(v; \boldsymbol{\mu}) + \int_\Omega [\mathcal{I}_M[\eta(\tilde{u}_N^k(\cdot; \boldsymbol{\mu}))] - \eta(\tilde{u}_N^k(\boldsymbol{\mu}))]v, \quad \forall v \in V_h. \tag{4.104}$$

We now choose $v = u_h(\boldsymbol{\mu}) - \tilde{u}_N^k(\boldsymbol{\mu})$ and note from our assumption of monotonic nondecreasing η that

$$\int_\Omega [\eta(u_h(\boldsymbol{\mu})) - \eta(\tilde{u}_N^k(\boldsymbol{\mu}))][u_h(\boldsymbol{\mu}) - \tilde{u}_N^k(\boldsymbol{\mu})] > 0. \tag{4.105}$$

We then consider the interpolation error term and invoke the Cauchy–Schwarz inequality and our definition of the norm $\|\cdot\|_V$; the result directly follows. We emphasize that without the property (4.105) we would need to estimate the inf-sup constant associated with the Jacobian and then consider a contraction argument per the Brezzi–Rappaz–Raviart theory [37] – hence much more difficult to realize in practice if we wish to quantitatively evaluate the necessary constants.

4.3.2.2.3 Offline-online computational procedures

As for the linear case, we presume here that the RB space and basis are provided.

We first consider the formation and solution of the RB nonlinear system. In the offline stage, we perform the EIM for the nonlinear term and subsequently form $\mathbb{B}_{N_{\max}}$, $\mathbb{C}_{N_{\max}}$, and $\boldsymbol{f}_{N_{\max}}^1$. Then at each Newton iteration, we first form η_N^k and $\dot{\eta}_N^k$ at cost $\mathcal{O}(MN)$ FLOPs. We next form the Jacobian at cost $\mathcal{O}(MN^2)$ and the residual at cost $\mathcal{O}(MN)$. Finally, we invert the Jacobian at cost $O(N^3)$ FLOPs.

We next consider the a posteriori error estimator and in particular the dual norm of the residual. In fact, once we evaluate $(\eta_{N\,m}^k(\boldsymbol{\mu}))_{m=1,\dots,M}$, the residual of our nonlinear

elliptic problem is computationally analogous to the residual of a linear elliptic problem, with the $(\eta_{N\,m}^{k}(\boldsymbol{\mu}))_{m=1,\dots,M}$ playing a very similar role to $(\mathbf{u}_N(\boldsymbol{\mu}))_{n=1,\dots,N}$. The online operation count for the dual norm of the residual is then $\mathcal{O}((N+M)^2)$, hence quite inexpensive; of course, if the weak form included other parameters, hence $p > 1$, the operation count would increase commensurately.

We close this section by emphasizing the importance of hyperreduction. We first consider nonpolynomial nonlinearity, for example $\eta(z) = |z|z$: In the absence of hyperreduction, the online cost would depend on N_h, both for formation of the Jacobian and for evaluation of the dual norm of the residual. We next consider polynomial nonlinearities, hence $\eta(z) = z^s$ for s an odd integer (in order to honor our monotonic nondecreasing assumption): In the absence of hyperreduction, the online cost for formation of the residual is $\mathcal{O}(N^{s+1})$ and the online cost for evaluation of the dual norm of the residual is $\mathcal{O}(N^{2s})$, and thus potentially prohibitive even for $s = 3$. We illustrate the difficulty for a polynomial nonlinearity $\eta(z) = z^3$. In this case the nonlinear contribution to the residual is given by

$$\underbrace{\sum_{n=1}^{N}\sum_{n'=1}^{N}\sum_{n''=1}^{N}(\tilde{\boldsymbol{u}}_N(\boldsymbol{\mu}))_n(\tilde{\boldsymbol{u}}_N(\boldsymbol{\mu}))_{n'}(\tilde{\boldsymbol{u}}_N(\boldsymbol{\mu}))_{n''}\underbrace{\int_{\Omega}\zeta_h^n\zeta_h^{n'}\zeta_h^{n''}\zeta_h^{n'''}}_{\text{form offline}}}_{\text{evaluate online}}, \tag{4.106}$$

where $\zeta_h^{n'''}$ plays the role of test function.

4.4 Approximation spaces

4.4.1 Elliptic problems: weak greedy method

Perhaps the most simple – and the most popular – approach to the identification of reduced-order approximation spaces is the proper orthogonal decomposition (POD).[6] It consists in building a matrix with entries $(\tilde{u}_h(\boldsymbol{\mu}), \tilde{u}_h(\boldsymbol{\mu}'))_V$, $\boldsymbol{\mu}$, and $\boldsymbol{\mu}'$ belonging to some subset Ξ_{POD} of $\mathcal{P}$ with cardinal K_{POD} (large enough since Ξ_{POD} is supposed to scan well $\mathcal{P}$), and considering the eigenvectors associated to the largest values of its largest eigenvalues. The interest of this approach is two-fold:
– it is very simple to implement and does not rely on further mathematical ingredients as is required in other methods (such as error estimators for the weak greedy approach that will be explained next);

---

[6] The POD can also be interpreted in terms of singular value decomposition, principal component analysis, or even Karhunen–Loève transform.

- it provides some hope that an RB exists (when the eigenvalues' decay is fast enough), but also, on the contrary, a proof that the Kolmogorov N-width is (in some bad situations) large since $d_N(\mathcal{M}, V)$ is always larger than $\sum_{j>N} \sigma_j^2$.

However, in the parametric RB context, the POD has several drawbacks:
- formation of the covariance matrix requires many (K_{POD}) queries to the FE approximation, $\boldsymbol{\mu} \mapsto \tilde{u}_h(\boldsymbol{\mu})$; identification of the POD modes requires solution of a dense $K_{\mathrm{POD}} \times K_{\mathrm{POD}}$ eigenproblem;
- the norm in which the approximation is optimal is $L^2(\mathcal{P}; V)$, which is not ideal for pointwise (in parameter) queries, $\boldsymbol{\mu} \mapsto \tilde{u}_N(\boldsymbol{\mu})$.

Alternatives, widely used in the RB context, are the greedy and (because more amenable for implementation) the weak greedy approach: The optimality norm $L^\infty(\mathcal{P}; V)$ is more appropriate, consistent with the Kolmogorov N-width definition; only N_{max} FE queries are required to identify the RB spaces $\{V_N\}_{N=1,\dots,N_{\mathrm{max}}}$; the resulting approximation space nevertheless reflects information from $K_{\mathrm{trial}} \gg N_{\mathrm{max}}$ points on the parametric manifold. The POD is discussed in depth in several other chapters in this handbook; we focus here on the greedy-type methods.

We first present the algorithm: Let Ξ_{trial} be a given subset of $\mathcal{P}$ with cardinal K_{trial} (large enough since Ξ_{trial} is supposed to scan well $\mathcal{P}$).

Algorithm 4.2: Greedy method: We assume that the set $\{\tilde{u}_h(\boldsymbol{\mu})\}_{\boldsymbol{\mu} \in \Xi_{\mathrm{trial}}}$ is not embedded in a small finite-dimensional space.

Data: $\Xi_{\mathrm{trial}}, \tilde{u}_h : \mathcal{P} \to V_h, \| \cdot \|_{\mathrm{Greedy}}, \mathrm{tol}_{\mathrm{Greedy}}$

Result: $M, \{\boldsymbol{\mu}_m \in \Xi_{\mathrm{trial}}\}_{1 \le m \le M}, \{\boldsymbol{\zeta}^i \in V_h\}_{1 \le i \le M}, V_M \subset V, \dim V_M = M$

1 Set $M = 0$, $V_0 = \emptyset$, and err $= \infty$;

2 **while** err $> \mathrm{tol}_{\mathrm{Greedy}}$ **do**

3   Set $M \leftarrow M + 1$;

4   Find $\boldsymbol{\mu}_M = \arg \sup_{\boldsymbol{\mu} \in \Xi_{\mathrm{trial}}} \|\tilde{u}_h(\boldsymbol{\mu}) - \Pi_{M-1}\tilde{u}_h(\boldsymbol{\mu})\|_{\mathrm{Greedy}}$ (where Π_{M-1} denotes a projection approach – like an orthogonal projection in V or a Galerkin projection – onto V_{M-1} according to $\| \cdot \|_{\mathrm{Greedy}}$);

5   Define $\boldsymbol{\zeta}^M = (\tilde{u}_h(\boldsymbol{\mu}_M) - \Pi_{M-1}\tilde{u}_h(\boldsymbol{\mu}_M)) / \|(\tilde{u}_h(\boldsymbol{\mu}_M) - \Pi_{M-1}\tilde{u}_h(\boldsymbol{\mu}_M))\|_{\mathrm{Greedy}}$;

6   Update $V_M = \mathrm{Span}\{\boldsymbol{\zeta}^i, i = 1, \dots, M\}$;

7   Set err $= \|\tilde{u}_h(\boldsymbol{\mu}_M) - \Pi_{M-1}\tilde{u}_h(\boldsymbol{\mu}_M)\|_{\mathrm{Greedy}}$;

8 **end**

The pure greedy method is associated to the choice $\Xi_{\mathrm{trial}} = \mathcal{P}$ and $\| \cdot \|_{\mathrm{Greedy}} = \| \cdot \|_V$, the projection operator being the V-orthogonal projection. It is a theoretical algorithm as the computation of $\boldsymbol{\mu}^*$ on line 4 above is quite impossible to determine as it requires in particular the knowledge of every $\tilde{u}_h(\boldsymbol{\mu})$, for any $\boldsymbol{\mu} \in \Xi_{\mathrm{trial}}$. Alternatively a weak greedy approach can be implemented where the exact V-norm is replaced by a surrogate,

an example being given by the a posteriori error estimate for Galerkin-RB approxima-
tions, as the ones that have been presented in Section 3 (see [38] for the first actual
use of this weak greedy approach). From (4.58), we can thus interpret the weak ver-
sion with respect to the pure greedy method by replacing line 4 above by

$$\text{Select a } \boldsymbol{\mu}_m \text{ such that } \|\tilde{u}_h(\boldsymbol{\mu}_m) - \Pi_{M-1}\tilde{u}_h(\boldsymbol{\mu}_m)\|_V \geq \gamma \sup_{\boldsymbol{\mu}\in\mathcal{P}} \|\tilde{u}_h(\boldsymbol{\mu}) - \Pi_{M-1}\tilde{u}_h(\boldsymbol{\mu})\|_V \quad (4.107)$$

with some given γ, such that $0 < \gamma < 1$ (in this context, the denomination "weak
greedy" was first introduced in [8]). Another element that enters in the weak greedy
is the choice $\Xi_{\text{trial}} \subset_{\neq} \mathcal{P}$ that should be large enough and well chosen so that (4.107)
remains true.

4.4.2 Optimality of the weak greedy method

In this subsection we state recent results about the performance of the weak greedy al-
gorithm compared to the Kolmogorov N-width optimal choice given by $d_N(\mathcal{M}, V)$. First
of all, we define $\sigma_N(\mathcal{M}, V)$: a comparable quantity with respect to $d_N(\mathcal{M}, V)$ which is
associated to the series of spaces $\{V_n\}_{n\geq 0}$ defined in the greedy algorithm

$$\sigma_n(\mathcal{M}, V) = \text{dist}(\mathcal{M}, V_n) = \sup_{\boldsymbol{\mu}} \inf_{v_n \in V_n} \|\tilde{u}_h(\boldsymbol{\mu}) - v_n\|_V \quad (4.108)$$

that characterizes the approximation space resulting from the weak greedy algorithm.

Of course, if $(\sigma_n(\mathcal{M}, V))_{n\geq 0}$ decays at a rate comparable to $(d_n(\mathcal{M}, V))_{n\geq 0}$, this
means that the greedy selection provides essentially the best possible accuracy at-
tainable by n-dimensional subspaces. The first comparison between $(\sigma_n(\mathcal{M}, V))_{n\geq 0}$
and $(d_n(\mathcal{M}, V))_{n\geq 0}$ was given in [9]: $\sigma_n(\mathcal{M}, V) \leq Cn2^n d_n(\mathcal{M}, V)$ which, of course, is
only useful if $d_n(\mathcal{M}, V)$ decays to zero faster than $n^{-1}2^{-n}$. A more conservative esti-
mate results from the series of papers from [8] and [14]. We present here their proof in
the Hilbertian context.

For any choice Ξ_{trial} with cardinal $K_{\text{trial}} > 0$ large enough, let us consider the set
$X_{K_{\text{trial}}}$ of all vectors $\tilde{u}_h(\boldsymbol{\mu})$, $\boldsymbol{\mu} \in \Xi_{\text{trial}}$. It is included in $\mathcal{M}$ and thus we derive obviously

$$\forall m > 0, \quad d_m(X_{K_{\text{trial}}}, V) \leq d_m(\mathcal{M}, V). \quad (4.109)$$

This means that there exist m vectors $b_1, b_2, \ldots, b_m$, generating a finite-dimensional
space $\mathcal{H}_m$ such that

$$\max_{\boldsymbol{\mu}\in\Xi_{\text{trial}}} \min_{\mathbf{c}_m\in\mathcal{H}_m} \|\tilde{u}_h(\boldsymbol{\mu}) - \mathbf{c}_m\|_V \leq d_m(\mathcal{M}, V). \quad (4.110)$$

By taking the projection of these vectors $b_1, b_2, \ldots, b_m$ in $\text{Span}(X_{K_{\text{trial}}})$, we can determine
m orthonormal vectors in $\text{Span}(X_{K_{\text{trial}}})$, which we denote by $\tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_m$ that generate

a finite-dimensional space $\widehat{\mathcal{H}}_m \subset \mathrm{Span}(X_{K_{\mathrm{trial}}})$ of dimension m^7 such that

$$\max_{\boldsymbol{\mu} \in \Xi_{\mathrm{trial}}} \min_{\widetilde{\mathbf{c}_m} \in \widehat{\mathcal{H}}_m} \|\tilde{u}_h(\boldsymbol{\mu}) - \widetilde{\mathbf{c}_m}\|_V \le d_m(\mathcal{M}, V). \tag{4.111}$$

Let us now introduce the scalar products $a_{n,j} = (\tilde{u}_h(\boldsymbol{\mu}_n), \boldsymbol{\zeta}^j)_V$, $1 \le j, n \le K_{\mathrm{trial}}$ where $\boldsymbol{\zeta}^{\cdot}$ are defined on line 5 of Algorithm 4.2). We easily get $a_{n,j} = 0$ for $j > n$ and $\tilde{u}_h(\boldsymbol{\mu}_n) = \sum_{j=0}^{n} a_{n,j}\zeta_j$. The matrix A with entries $a_{n,j}$ is lower triangular and incorporates all the information about the weak greedy algorithm on $\mathcal{M}$. In addition, whatever the definition of the space V, this matrix leads us to an analysis in $\ell^2(\mathbb{N})$ (even $\ell^2(\{1, 2, \ldots, K_{\mathrm{trial}}\})$) more simple than the analysis in V. From the very definition of the greedy selection, we get the two following properties:

P$_1$: The diagonal elements of A satisfy $\gamma \sigma_n(\mathcal{M}, V) \le |a_{n,n}| \le \sigma_n(\mathcal{M}, V)$.

P$_2$: For every $m \ge n$, one has $\sum_{j=n}^{m} a_{m,j}^2 \le \sigma_n(\mathcal{M}, V)^2$.

For any K and m, $1 \le m < K$, and any $M \ge 0$, let us consider the $K \times K$ matrix $G = (g_{i,j})$ extracted from A by considering the rows and columns of A with indices from $\{M + 1, \ldots, M + K\}$. Each row $\mathbf{y}_i$ of G is the coordinate of the projection of $\tilde{u}_h(\boldsymbol{\mu}_{M+i})$ in the vector space spanned by $\zeta_{M+1}, \zeta_{M+2}, \ldots, \zeta_{M+K}$. Similarly as what we have done on the projection of $\tilde{u}_h(\boldsymbol{\mu}_{M+i})$ in the vector space spanned by $\zeta_{M+1}, \zeta_{M+2}, \ldots, \zeta_{M+K}$, we project each vector $\tilde{b}_1, \tilde{b}_2, \ldots, \tilde{b}_m$ in the vector space spanned by $\zeta_{M+1}, \zeta_{M+2}, \ldots, \zeta_{M+K}$ and further normalize them. The associated vectors are denoted as $\hat{b}_i$, $1 \le i \le m$ and span a finite-dimensional space $\widehat{\mathcal{H}}_m$ that satisfies, from (4.111),

$$\max_i \min_{\widehat{\mathbf{c}_m} \in \widehat{\mathcal{H}}_m} \|\mathbf{y}_i - \widehat{\mathbf{c}_m}\|_{\ell^2} \le d_m(\mathcal{M}, V). \tag{4.112}$$

Now we note that the ℓ^2-norm of each $\mathbf{y}_i$ is, from property **P$_2$** above,

$$\|\mathbf{y}_i\|_{\ell^2} \le \sigma_M(\mathcal{M}, V).$$

The projection of each of these vectors on the vectorial space spanned by $\hat{b}_i$, $1 \le i \le m$, belongs to a ball (in dimension m) of radius $\le \sigma_M(\mathcal{M}, V)$: $\mathcal{B}(0, \sigma_M(\mathcal{M}, V))$, due to the m-width property (4.112), each of the vectors $\mathbf{y}_i$ thus belong to a "cylinder" with basis $\mathcal{B}(0, \sigma_M(\mathcal{M}, V))$ and height $\le d_m(\mathcal{M}, V)$ (in dimension $K-m$). The volume of the convex set spanned by these vectors $\mathbf{y}_i$ (that is equal to the determinant of G) is thus upper bounded by $\mathcal{V}_m = \mathrm{Vol}(\mathcal{B}(0, \sigma_M(\mathcal{M}, V))) \times d_m(\mathcal{M}, V)^{K-m}$. Recalling that the matrix G is lower diagonal, we thus get from **P$_1$**

$$\det G = \Pi_{n=M+1}^{K+M}|a_{n,n}| \le \mathcal{V}_m \le C\sigma_M(\mathcal{M}, V))^m \times d_m(\mathcal{M}, V)^{K-m}, \tag{4.113}$$

---

7 Note that if the projection of $\mathcal{H}_m$ in $\mathrm{Span}(X_{K_{\mathrm{trial}}})$ is of dimension $< m$, we choose any vector $\tilde{b}$ in $\mathrm{Span}(X_{K_{\mathrm{trial}}})$ to complement the family $\{\tilde{b}_i\}_{1 \le i \le m}$.

where C is here a universal constant bounding the volume of the unit-ball, whatever the dimension (we can take for instance $C = 6$).

We thus get (a slightly improved version of) Theorem 3.2 in [14].

Theorem. *For the weak greedy algorithm with constant γ in a Hilbert space V and for any compact set $\mathcal{M}$, we have the following inequalities between $\sigma_n = \sigma_n(\mathcal{M}, V)$ and $d_n = d_n(\mathcal{M}, V)$ for any $M \geq 0, 1 \leq m < K$:*

$$\Pi_{i=1}^{K} \sigma_{M+i} \leq 6\gamma^{-K} \sigma_M^m d_m^{K-m}. \tag{4.114}$$

Then, as of consequence of various choices of the values M, K, and m, we get the following corollary.

Corollary. *For the weak greedy algorithm with constant γ in V, we have the following:*
– *For any $n \geq 1$, we have*

$$\sigma_n(\mathcal{M}, V) \leq 6^{\frac{1}{n}} \gamma^{-1} \min_{1 \leq m \leq n} d_m^{\frac{n-m}{n}} (\mathcal{M}, V).$$

In particular $\sigma_{2n}(\mathcal{M}, V) \leq \sqrt{2}\gamma^{-1} \sqrt{d_n(\mathcal{M}, V)}, n \geq 1$.
– *If $d_n(\mathcal{M}, V) \leq C_0 n^{-\alpha}$, then $\sigma_n(\mathcal{M}, V) \leq C_1 n^{-\alpha}$ with $C_1 = 2^{5\alpha+1}\gamma^{-2}C_0$.*
– *If $d_n(\mathcal{M}, V) \leq C_0 e^{-c_0 n^\alpha}$, then $\sigma_n(\mathcal{M}, V) \leq \sqrt{2C_0}\gamma^{-1}e^{-c_1 n^\alpha}$, where $c_1 = 2^{-1-2\alpha}c_0$.*

For the first item, we take $M = 0$, $K = n$, and any $1 \leq m < n$ in the previous theorem, and use the monotonicity of σ and the fact that $\sigma_0 \leq 1$. For the proof of the two other items we refer to [14].

As a final obvious remark we want to point out that $d_m(\mathcal{M}, V)$ depends on $\mathcal{M}$ and thus on $\mathcal{P}$. This suggests a localization greedy procedure so as to lower the effect of the large size of $\mathcal{P}$; we refer to [16, 2, 29] for various approaches to exploit this argument.

4.4.3 Extensions

4.4.3.1 Evolution problems: parabolic PDEs

The greedy method for elliptic problems has been extended to the parabolic case in the POD greedy method first proposed in [20]. On line 4 of Algorithm 4.2 we still identify $\boldsymbol{\mu}_m$ as the "worst-approximated" parameter value, but now the error bound is typically evaluated at the final time, T. The update on line 5 is then given as some number of POD modes associated with the projection error.

4.4.3.2 A nonlinear elliptic problem

For our particular nonlinear problem and associated error estimator the weak greedy Algorithm 4.2 requires little modification. More generally, apart from difficulties asso-

ciated with error estimation, the weak greedy does extend relatively easily to the non-linear case. However, as already noted, the EIM preparation of the nonlinear problem can be expensive, and for this reason more advanced techniques – which combine the EIM and RB greedy algorithms – are an active area of research (e. g., [13, 6]).

Bibliography

[1] B. O. Almroth, P. Stern, and F. A. Brogan, Automatic choice of global shape functions in structural analysis, *AIAA Journal*, **16** (5) (1978), 525–528.

[2] D. Amsallem, M. J. Zahr, and C. Farhat, Nonlinear model order reduction based on local reduced-order bases, *International Journal for Numerical Methods in Engineering*, **92** (10) (2012), 891–916.

[3] M. Bachmayr and A. Cohen, Kolmogorov widths and low-rank approximations of parametric elliptic PDEs, *Mathematics of Computation*, **86** (304) (2017), 701–724.

[4] O. Balabanov and A. Nouy, Randomized linear algebra for model reduction. Part I: Galerkin methods and error estimation. *Advances in Computational Mathematics*, **45** (2019), 2969–3019.

[5] M. Barrault, Y. Maday, N. Nguyen, and A. T. Patera, An empirical interpolation method: Application to efficient reduced-basis discretization of partial differential equations, *Comptes Rendus de L'Académie Des Sciences. Série 1, Mathématique*, **339** (2004), 667–672.

[6] A. Benaceur, V. Ehrlacher, A. Ern, and S. Meunier, A progressive reduced basis/empirical interpolation method for nonlinear parabolic problems, *SIAM Journal on Scientific Computing*, **40** (5) (2018), A2930–A2955.

[7] F. Bernard, A. Iollo, and S. Riffaud, Reduced-order model for the BGK equation based on POD and optimal transport, *Journal of Computational Physics*, **373** (2018), 545–570.

[8] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk, Convergence rates for greedy algorithms in reduced basis methods, *SIAM Journal on Mathematical Analysis*, **43** (3) (2011), 1457–1472.

[9] A. Buffa, Y. Maday, A. T. Patera, C. Prud'homme, and G. Turinici, *A priori* convergence of the greedy algorithm for the parametrized reduced basis method, *M2AN*, **46** (3) (2012), 595–603.

[10] N. Cagniart, Y. Maday, and B. Stamm, Model order reduction for problems with large convection effects, in *Contributions to Partial Differential Equations and Applications*, pp. 131–150, Springer, 2019.

[11] F. Casenave, A. Ern, and T. Lelièvre, Accurate and online-efficient evaluation of the *a posteriori* error bound in the reduced basis method, *M2AN*, **48** (1) (2014), 207–229.

[12] A. Cohen and R. DeVore, Kolmogorov widths under holomorphic mappings, *IMA Journal of Numerical Analysis*, **36** (1) (2016), 1–12.

[13] C. Daversin and C. Prud'homme, Simultaneous empirical interpolation and reduced basis method for non-linear problems, *Comptes Rendus de L'Académie Des Sciences. Série 1, Mathématique*, **353** (2015), 1105–1109.

[14] R. DeVore, G. Petrova, and P. Wojtaszczyk, Greedy algorithms for reduced bases in Banach spaces, *Constructive Approximation*, **37** (3) (2013), 455–466.

[15] M. Drohmann, B. Haasdonk, and M. Ohlberger, Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation, *SIAM Journal on Scientific Computing*, **34** (2) (2012), A937–A969.

[16] J. L. Eftang, A. T. Patera, and E. M. Rønquist, An "hp" certified reduced basis method for parametrized elliptic partial differential equations, *SIAM Journal on Scientific Computing*, **32** (6) (2010), 3170–3200.

[17] C. Farhat, T. Chapman, and P. Avery, Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models, *International Journal for Numerical Methods in Engineering*, **102** (5) (2015), 1077–1110.

[18] M. A. Grepl, Y. Maday, C. N. Nguyen, and A. T. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, *M2AN*, **41** (3) (2007), 575–605.

[19] M. A. Grepl and A. T. Patera, *A posteriori* error bounds for reduced-basis approximations of parametrized parabolic partial differential equations, *M2AN*, **39** (1) (2005), 157–181.

[20] B. Haasdonk and M. Ohlberger, Reduced basis method for finite volume approximation of parametrized linear evolution equations, *M2AN*, **42** (2) (2008), 277–302.

[21] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, Springer, 2016.

[22] 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 constants, *Comptes Rendus de L'Académie Des Sciences. Série 1, Mathématique*, **345** (2007), 473–478.

[23] G. G. Lorentz, Mv. Golitschek, and Y. Makovoz, *Constructive Approximation (Advanced Problems)*, Grundlehren der Mathematischen Wissenschaften, vol. 304, Springer-Verlag, 1966.

[24] L. Machiels, Y. Maday, I. B. Oliveira, A. T. Patera, and D. Rovas, Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems, *Comptes Rendus de L'Académie Des Sciences. Série 1, Mathématique*, **331** (2) (2000), 153–158.

[25] Y. Maday, A. T. Patera, and D. Rovas, A blackbox reduced-basis output bound method for noncoercive linear problems, in D. Cioranescu and J. L. Lions (eds.), *Nonlinear Partial Differential Equations and Their Applications, Collège de France Seminar*, vol. XIV, pp. 533–569, Elsevier, Amsterdam, 2002.

[26] Y. Maday, A. T. Patera, and G. Turinici, *A priori* convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations, *Journal of Scientific Computing*, **17** (1-4) (2002), 437–446.

[27] Y. Maday and B. Stamm, Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces, *SIAM Journal on Scientific Computing*, **35** (6) (2013), A2417–A2441.

[28] A. K. Noor and J. M. Peters, Reduced basis technique for nonlinear analysis of structures, *AIAA Journal*, **18** (4) (1980), 455–462.

[29] S. Pagani, A. Manzoni, and A. Quarteroni, Numerical approximation of parametrized problems in cardiac electrophysiology by a local reduced basis method. MATHICSE Technical Report Nr. 25.2017, 2017.

[30] A. Pinkus, *N-widths in Approximation Theory*, Springer Science and Business Media, 1985.

[31] 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, *Journal of Fluids Engineering*, **124** (1) (2002), 70–80.

[32] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations*, Springer, 2016.

[33] G. Rozza, D. B. P. Huynh, and A. T. Patera, Reduced basis approximation and *a posteriori* error estimation for affinely parametrized elliptic coercive partial differential equations, *Archives of Computational Methods in Engineering*, **15** (3) (2008), 229–275.

[34] G. Rozza and K. Veroy, On the stability of the reduced basis method for Stokes equations in parametrized domains, *Computer Methods in Applied Mechanics and Engineering*, **196** (7) (2007), 1244–1260.

[35] D. Ryckelynck, *A priori* hyperreduction method: an adaptive approach, *Journal of Computational Physics*, **202** (1) (2005), 346–366.

[36] K. Urban and A. T. Patera, An improved error bound for reduced basis approximation of linear parabolic problems, *Mathematics of Computation*, **83** (2014), 1599–1615.

[37] 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, *International Journal for Numerical Methods in Engineering*, **47** (2005), 773–788.

[38] 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, *AIAA Paper No. 2003-3847*, 2003.

[39] M. Yano, A reduced basis method for coercive equations with an exact solution certificate and spatio-parameter adaptivity: energy-norm and output error bounds, *SIAM Journal on Scientific Computing*, **40** (1) (2018), A388–A420.

[40] M. Yano and A. T. Patera, An LP empirical quadrature procedure for reduced basis treatment of parametrized nonlinear PDEs, *Computer Methods in Applied Mechanics and Engineering*, **344** (2019), 1104–1123.

Charbel Farhat, Sebastian Grimberg, Andrea Manzoni, and
Alfio Quarteroni

5 Computational bottlenecks for PROMs: precomputation and hyperreduction

Abstract: For many applications, the projected quantities arising from projection-based model order reduction (PMOR) must be repeatedly recomputed due to, for example, nonlinearities or parameter dependence. Such repetitive computations constitute a performance bottleneck. Specifically, they limit the ability of a projection-based reduced-order model (PROM) to deliver the coveted speedup factor. This chapter reviews several state-of-the-art approaches for mitigating this issue and organizes them into two categories. Methods in the first category divide the computation of projections, whenever possible, into two parts: one that is responsible for the aforementioned bottleneck but can be precomputed offline; and another part that must be repeatedly performed online but whose computational complexity scales only with the dimension of the PROM. Methods in the second category are known as hyperreduction methods: They achieve the desired computational efficiency by approximating either the quantity to be projected, or the result of the projection. The significance of hyperreduction for PMOR is highlighted using four numerical applications that are representative of academic and real-world problems.

Keywords: empirical interpolation, energy conserving sampling and weighting, hyperreduction, nonlinear model order reduction, reduced mesh

MSC 2010: 65M60, 65N12, 76M10

5.1 Introduction

There are two main purposes for performing model order reduction:
– Obtaining a low-dimensional model with *minimum storage requirements* in order to enable *online* computing in general and embedded computing in particular –

Charbel Farhat, Aeronautics and Astronautics, Mechanical Engineering, and Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA, USA
Sebastian Grimberg, Aeronautics and Astronautics, Stanford University, Stanford, CA, USA
Andrea Manzoni, MOX – Department of Mathematics, Politecnico di Milano, Milan, Italy
Alfio Quarteroni, MOX – Department of Mathematics, Politecnico di Milano & Institute of Mathematics, École Polytechnique Fédérale de Lausanne (Professor Emeritus), Milan, Italy

that is, computing inside a complete device that may include hardware and mechanical parts.

–   Obtaining a low-dimensional model with *minimum processing time requirements* in order to enable numerical predictions in *real-time*, or at least near real-time. For time-independent predictions, real-time is defined here as *the shortest possible wall-clock time*. For numerical predictions in the time interval $[t^0, T]$, where $t^0 \geq 0$ denotes an initial time and $T > t^0$ denotes a final time, real-time is defined here as *in $T - t^0$ units of time*: For example, simulating 1 second of flow around an aircraft in 1 second *wall-clock* time. Furthermore, this definition of real-time is expanded throughout the remainder of this chapter to include near real-time, in order to avoid repeating after each occurrence of the words "in real-time" the alternative words "or near real-time."

While both objectives stated above are typically desirable, the second objective implies in practice the first one. Unfortunately, the reciprocal is not true, as explained below.

For this purpose, the focus is set in this chapter on the parametric, time-dependent, semi-discrete, N_h-dimensional, full-order computational model (FOM)

$$\begin{cases} \mathbb{M}_{N_h}(\boldsymbol{\mu})\dot{\mathbf{u}}_{N_h}(t; \boldsymbol{\mu}) + \mathbf{f}_{N_h}(\mathbf{u}_{N_h}(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{g}_{N_h}(t; \boldsymbol{\mu}), \\ \mathbf{u}_{N_h}(0; \boldsymbol{\mu}) = \mathbf{u}_{N_h}^0(\boldsymbol{\mu}), \end{cases} \tag{5.1}$$

where:

– t denotes time, and the dot denotes a time derivative.
– $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^p$ denotes a parameter vector of dimension p, and $\mathcal{P}$ denotes the *bounded* parameter space of interest. It can be physical in the sense that it can pertain, for example, to initial conditions, boundary conditions, problem geometry, and/or material properties. It can also be nonphysical, for example, if $\boldsymbol{\mu}$ is a realization parameter or a hyperparameter vector associated with a stochastic process [36, 23].
– $\mathbb{M}_{N_h}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$ is the parametric, *mass* matrix of this parametric FOM. For simplicity, but without any loss of generality, $\mathbb{M}_{N_h}$ is assumed here to be a symmetric positive definite (SPD) matrix.
– $\mathbf{u}_{N_h}(t; \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is the parametric, time-dependent, semi-discrete solution vector modeling the state of the system represented by this FOM. It is also often referred to as the vector of degrees of freedom (DOFs).
– $\mathbf{u}_{N_h}^0(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is an initial condition for $\mathbf{u}_{N_h}(t; \boldsymbol{\mu})$.
– $\mathbf{f}_{N_h}(\mathbf{u}_{N_h}(t; \boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is a parametric, nonlinear function referred to here and throughout the remainder of this chapter as the *nonlinear, internal force* vector, and its Jacobian

$$\mathbb{K}_{N_h}(\mathbf{u}_{N_h}(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \frac{\partial \mathbf{f}_{N_h}}{\partial \mathbf{u}_{N_h}}(\mathbf{u}_{N_h}(t; \boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h} \tag{5.2}$$

represents the parametric, *tangent stiffness* of this FOM. In the linear case,

$$\mathbf{f}_{N_h}\big(\mathbf{u}_{N_h}(t; \boldsymbol{\mu}); \boldsymbol{\mu}\big) = \mathbb{K}_{N_h}(\boldsymbol{\mu})\mathbf{u}_N(t; \boldsymbol{\mu}), \tag{5.3}$$

where $\mathbb{K}_{N_h}(\boldsymbol{\mu})$ is a parametric but otherwise constant matrix representing simply the stiffness of the parametric FOM (5.1).

- $\mathbf{g}_{N_h}(t; \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ is a parametric, time-dependent vector referred to here and throughout the remainder of this chapter as the *external force* vector.

Let

$$\mathbf{u}_{N_h}(t; \boldsymbol{\mu}) \approx \mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}) \quad \forall t \in [t^0, T], \ \mathbb{V} \in \mathbb{R}^{N_h \times n}, \ \mathbf{u}_n \in \mathbb{R}^n, \ n \ll N_h. \tag{5.4}$$

In the above subspace approximation, $\mathbb{V}$ is referred to as the *right* reduced-order basis (ROB): It is typically orthonormalized with respect to some inner product for conditioning reasons. The vector $\mathbf{u}_n$ is referred to as the vector of *reduced* or *generalized coordinates*.

Let $\mathbb{W} \in \mathbb{R}^{N_h \times n}$ denote the *left* ROB associated with $\mathbb{V}$ and orthogonal to it – that is,

$$\mathbb{W}^T \mathbb{V} = \mathbb{I}_n, \tag{5.5}$$

where the superscript T designates the transpose of the quantity to which it is applied, and $\mathbb{I}_{N_h} \in \mathbb{R}^{N_h \times N_h}$ denotes the identity matrix of size N_h.

The projection-based, Petrov–Galerkin $(\mathbb{W}, \mathbb{V})$, reduced-order model (PROM) associated with the FOM (5.1) and the subspace approximation (5.4) can be written as

$$\begin{cases} \mathbb{M}_n(\boldsymbol{\mu})\dot{\mathbf{u}}_n(t; \boldsymbol{\mu}) + \mathbf{f}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{g}_n(t; \boldsymbol{\mu}), \\ \mathbf{u}_n(0; \boldsymbol{\mu}) = \mathbf{u}_n^0(\boldsymbol{\mu}), \end{cases} \tag{5.6}$$

where:

- The reduced matrix

$$\mathbb{M}_n(\boldsymbol{\mu}) = \mathbb{W}^T \mathbb{M}_{N_h}(\boldsymbol{\mu})\mathbb{V} \in \mathbb{R}^{n \times n} \tag{5.7}$$

is the parametric, mass matrix of the above parametric PROM of dimension $n \ll N_h$; therefore, it is the *reduced* mass matrix.

- The reduced vector

$$\mathbf{f}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbb{W}^T \mathbf{f}_{N_h}\big(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}\big) \in \mathbb{R}^n \tag{5.8}$$

is the parametric, *reduced, nonlinear, internal force* vector. From the definition of the tangent stiffness given in (5.2) and that of the subspace approximation (5.4),

it follows that the reduced Jacobian matrix $\mathbb{K}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \frac{\partial \mathbf{f}_n}{\partial \mathbf{u}_n}(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ is related to the full-order Jacobian matrix $\mathbb{K}_{N_h}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ by

$$\mathbb{K}_n = \frac{\partial \mathbf{f}_n}{\partial \mathbf{u}_n} = \mathbb{W}^T \frac{\partial \mathbf{f}_{N_h}}{\partial \mathbf{u}_n} = \mathbb{W}^T \frac{\partial \mathbf{f}_{N_h}}{\partial \mathbf{u}_{N_h}} \frac{\partial \mathbf{u}_{N_h}}{\partial \mathbf{u}_n} = \mathbb{W}^T \mathbb{K}_{N_h} \mathbb{V} \in \mathbb{R}^{n \times n}. \tag{5.9}$$

Hence, the reduced Jacobian matrix $\mathbb{K}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ represents the parametric, *reduced tangent stiffness* of the parametric PROM (5.6). In the linear case where (5.3) holds, $\mathbb{K}_n(\boldsymbol{\mu})$ is parametric but otherwise constant – because in this case $\mathbb{K}_{N_h}$ in (5.9) is parametric but otherwise constant – and represents simply the parametric stiffness of this PROM.

– The reduced vector

$$\mathbf{g}_n(t; \boldsymbol{\mu}) = \mathbb{W}^T \mathbf{g}_{N_h}(t; \boldsymbol{\mu}) \in \mathbb{R}^n \tag{5.10}$$

is the parametric, time-dependent, *reduced, external force* vector.

– $\mathbf{u}_n^0(\boldsymbol{\mu}) = \mathbb{W}^T \mathbf{u}_{N_h}^0(\boldsymbol{\mu}) \in \mathbb{R}^n$ is the initial condition for $\mathbf{u}_n(t; \boldsymbol{\mu})$ associated with the initial condition for $\mathbf{u}_{N_h}(t; \boldsymbol{\mu})$: Its expression results from the subspace approximation (5.4) and the orthogonality condition (5.5).

The counterpart of the PROM (5.6) based on a Galerkin projection is obtained by setting $\mathbb{W} = \mathbb{V}$.

5.1.1 Computational bottlenecks

First, consider specifically the linear instance of the FOM (5.1) – that is, the case where $\mathbf{f}_{N_h}(\mathbf{u}_{N_h}(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbb{K}_{N_h}(\boldsymbol{\mu})\mathbf{u}_{N_h}(t; \boldsymbol{\mu})$ – which can be written as

$$\begin{cases} \mathbb{M}_{N_h}(\boldsymbol{\mu})\dot{\mathbf{u}}_{N_h}(t; \boldsymbol{\mu}) + \mathbb{K}_{N_h}(\boldsymbol{\mu})\mathbf{u}_{N_h}(t; \boldsymbol{\mu}) = \mathbf{g}_{N_h}(t; \boldsymbol{\mu}), \\ \mathbf{u}_{N_h}(0; \boldsymbol{\mu}) = \mathbf{u}_{N_h}^0(\boldsymbol{\mu}). \end{cases} \tag{5.11}$$

From (5.3) and (5.7)–(5.10), it follows that the PROM (5.6) can be written in this case as

$$\begin{cases} \underbrace{(\mathbb{W}^T \mathbb{M}_{N_h}(\boldsymbol{\mu})\mathbb{V})}_{\mathbb{M}_n(\boldsymbol{\mu})} \dot{\mathbf{u}}_n(t; \boldsymbol{\mu}) + \underbrace{(\mathbb{W}^T \mathbb{K}_{N_h}(\boldsymbol{\mu})\mathbb{V})}_{\mathbb{K}_n(\boldsymbol{\mu})} \mathbf{u}_n(t; \boldsymbol{\mu}) = \underbrace{\mathbb{W}^T \mathbf{g}_{N_h}(t; \boldsymbol{\mu})}_{\mathbf{g}_n(t; \boldsymbol{\mu})}, \\ \mathbf{u}_n(0; \boldsymbol{\mu}) = \mathbb{W}^T \mathbf{u}_{N_h}^0(\boldsymbol{\mu}). \end{cases} \tag{5.12}$$

The real-time performance of the processing of the above parametric, linear PROM hinges on the real-time evaluation of the reduced matrices $\mathbb{M}_n(\boldsymbol{\mu})$ and $\mathbb{K}_n(\boldsymbol{\mu})$, and reduced vectors $\mathbf{g}_n(t; \boldsymbol{\mu})$ and $\mathbf{u}_n(0; \boldsymbol{\mu})$. However, the computational complexity of the evaluation of each of these reduced matrices and reduced vectors, which must be repeated every time $\boldsymbol{\mu}$ is varied, scales as $\mathcal{O}(N_h^2 n)$ and $\mathcal{O}(N_h n)$, respectively. In other

words, this computational complexity scales with integer powers of the size N_h of the FOM. For large values of N_h, this prohibits the evaluation in real-time of the reduced quantities, even when the dimension n of the linear PROM (5.12) is very small.

Nonlinearity aggravates the computational complexity issue raised above for parametric, linear PROMs in the following sense. Even in the *nonparametric* case, achieving a low dimension $n \ll N_h$ for a nonlinear PROM such as (5.6), while meeting accuracy requirements, does not suffice to enable real-time numerical predictions. To highlight this aggravation, which of course persists in the case of the parametric, nonlinear PROM (5.6), consider next the nonparametric instance of the nonlinear FOM (5.1). Because in this case $\mathbb{M}$ is independent of $\boldsymbol{\mu}$, it is more convenient to construct the left ROB $\mathbb{W}$ such that it satisfies

$$\mathbb{W}^T \mathbb{M}_{N_h} \mathbb{V} = \mathbb{I}_n \tag{5.13}$$

instead of the orthogonality condition stated in (5.5). Indeed, from (5.7), (5.13), and (5.8), it follows that the expression of the nonlinear PROM (5.6) simplifies in this case to

$$\begin{cases} \mathbb{I}_n \dot{\mathbf{u}}_n(t) + \underbrace{\mathbb{W}^T \mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t))}_{\mathbf{f}_n(\mathbf{u}_n(t))} = \underbrace{\mathbb{W}^T \mathbf{g}_{N_h}(t)}_{\mathbf{g}_n(t)}, \\ \mathbf{u}_n(0) = \mathbb{W}^T \mathbf{u}_{N_h}^0. \end{cases} \tag{5.14}$$

There are two main approaches for processing the above nonparametric, nonlinear PROM – namely, the explicit approach and the implicit approach:

- Processing the PROM (5.14) using an explicit time integration scheme requires the evaluation at each time step of the *reduced, nonlinear, force balance* vector $\mathbb{W}^T(\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t)) - \mathbf{g}_{N_h}(t))$. Again, the computational complexity of this evaluation, which must be performed at least once at each time step of the time integration process, scales as $\mathcal{O}(N_h n)$. Therefore, it scales with the size of the FOM. Even when the dimension n is very small, this prohibits the processing of the PROM (5.14) in real-time (for example, see [21] for wall-clock timings associated with realistic engineering computations).
- On the other hand, processing the PROM (5.14) by an implicit time integration scheme gives rise at each time step to a system of nonlinear algebraic equations. Solving this system of equations by the Newton method or a variant requires reconstructing at each Newton iteration, within each time step, the reduced tangent stiffness matrix $\mathbb{K}_n(\mathbf{u}_n(t)) = \mathbb{W}^T \mathbb{K}_{N_h}(\mathbb{V}\mathbf{u}_n(t))\mathbb{V}$ (see (5.9)) in addition to the reduced, nonlinear, internal force vector. The computational complexity of each reconstruction of $\mathbb{K}_n(\mathbf{u}_n(t))$ scales as $\mathcal{O}(N_h^2 n)$. Again, this prohibits in general the processing of the nonparametric, nonlinear PROM (5.14) in real-time.

The computational issue highlighted above also hinders the real-time performance of time-independent PROMs in almost the same way. Indeed, for steady-state (or static)

problems, the parametric, linear PROM (5.12) simplifies to

$$\underbrace{\left(\mathbb{W}^T \mathbb{K}_{N_h}(\boldsymbol{\mu})\mathbb{V}\right)}_{\mathbb{K}_n(\boldsymbol{\mu})} \mathbf{u}_n(\boldsymbol{\mu}) = \underbrace{\mathbb{W}^T \mathbf{g}_{N_h}(\boldsymbol{\mu})}_{\mathbf{g}_n(\boldsymbol{\mu})} \tag{5.15}$$

and the nonlinear PROM (5.6) simplifies to

$$\underbrace{\mathbb{W}^T \mathbf{f}_{N_h}\left(\mathbb{V}\mathbf{u}_n(\boldsymbol{\mu}); \boldsymbol{\mu}\right)}_{\mathbf{f}_n(\mathbf{u}_n(\boldsymbol{\mu}); \boldsymbol{\mu})} = \underbrace{\mathbb{W}^T \mathbf{g}_{N_h}(\boldsymbol{\mu})}_{\mathbf{g}_n(\boldsymbol{\mu})}. \tag{5.16}$$

In the case of the parametric, linear, static PROM (5.15), the solution for each different parameter vector $\boldsymbol{\mu}^\star$ sampled in $\mathcal{P}$ of the corresponding reduced problem (5.15), in order to determine $\mathbf{u}_n(\boldsymbol{\mu}^\star)$, gives rise to the same computational bottleneck associated with rebuilding the reduced matrix $\mathbb{W}^T \mathbb{K}_{N_h}(\boldsymbol{\mu}^\star)\mathbb{V}$ in $\mathcal{O}(N_h{}^2 n)$ operations. In the case of the counterpart nonlinear PROM (5.16), the same computational bottleneck arises at each iteration of Newton's method – or a variant of this method – applied to the solution of the underlying nonlinear problem, during the update of the tangent stiffness matrix $\mathbb{K}_n(\mathbf{u}_n)$, whether the nonlinear PROM (5.16) is parametric or not.

5.1.2 Solution approaches

Two major approaches have been developed so far for addressing the computational bottlenecks associated with the repeated reconstructions of a PROM highlighted above. Both of them share the offline-online paradigm that underlies most if not all PROM computations. Both approaches can also be described, broadly speaking, as divide-and-conquer approaches.

The first approach divides the computation of the reduced quantities, whenever possible, into two parts: one part that is responsible for the aforementioned computational bottlenecks and can be precomputed offline; and another part whose computational complexity scales with integer powers of the small size n of the PROM and therefore can be processed online and in real-time. This approach is both *feasible* and *exact* for at least two different classes of problems: the class of parametric, linear FOMs (5.11) admitting an *efficient* parameter-affine representation; and the class of nonlinear FOMs with a low-order polynomial dependence of the internal force vector on the solution (see (5.22)) and a time-independent external force vector.

By contrast, the second major approach for achieving the same objective can be characterized as an *inexact* approach, where an additional layer of approximations is introduced into the PROM construction process in order to enable the evaluation of all reduced quantities in real-time. This approach includes the computational paradigm known today as *hyperreduction*.

5.1.3 Chapter organization

Throughout this chapter, the parametric FOM of interest is assumed to be reducible. Therefore, the objective is set to squeezing the most performance out of the corresponding PROM. To this end, this chapter focuses on presenting the state of the art of both major approaches outlined above for mitigating if not eliminating the computational bottlenecks highlighted in Section 5.1.1. It is organized as follows.

Section 5.2 reviews the notions of global and local ROBs, and their associated global and local PROMs. Section 5.3 presents the first approach for addressing the computational bottlenecks associated with the repeated reconstructions of a PROM highlighted above. Section 5.4 reviews the second approach, and more specifically, two methodologies that belong to it. The first one targets parametric, linear PROMs. It is based on the concept of a database of pointwise linear PROMs, and that of interpolation on matrix manifolds. The second methodology is known as hyperreduction. It is equally applicable to parametric and nonparametric, linear and nonlinear PROMs. Section 5.5 illustrates two of the most successful hyperreduction methods with both academic and real-world, parametric and nonparametric, linear and nonlinear applications. Section 5.6 summarizes and concludes this chapter.

5.2 Global and pointwise ROBs for parametric PROMs

Because of the sheer number of computations (or simulations) they imply, parametric numerical predictions are a major source of motivations for projection-based model order reduction (PMOR). However, they simultaneously constitute a source of complications for the construction of a PROM that is robust with respect to variations in the parameter vector $\boldsymbol{\mu}$. A popular approach for addressing this issue is to construct a *single* right ROB $\mathbb{V}$ for which the approximation (5.4) is accurate in the entire parameter space of interest $\mathcal{P}$ – that is,

$$\mathbf{u}_{N_h}(t; \boldsymbol{\mu}) \approx \mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}) \;\; \forall t \in [t^0, T], \;\; \underline{\forall \boldsymbol{\mu} \in \mathcal{P}}; \quad \mathbb{V} \in \mathbb{R}^{N_h \times n}, \, \mathbf{u}_n \in \mathbb{R}^n, \, n \ll N_h. \quad (5.17)$$

In this case, the right ROB $\mathbb{V}$ is typically called a *global* right ROB. It is constructed by: sampling many points in $\mathcal{P}$ using any sampling procedure, but preferably a greedy procedure; generating one or multiple solution snapshots at each of these points; and compressing them if needed or desired to construct $\mathbb{V}$. Note that for a Petrov–Galerkin-PROM, the construction of a global right ROB $\mathbb{V}$ calls for the additional construction of a corresponding global left ROB $\mathbb{W}$.

While it is particularly effective in the nonlinear setting, the global ROB approach is equally valid in the linear one. In this sense, it is a comprehensive approach. However, if $\mathcal{P}$ is high-dimensional and/or the solution of the parametric FOM is highly

sensitive with respect to variations in $\boldsymbol{\mu}$ – in which case the solution manifold of the governing FOM equations has a large Kolmogorov n-width – the global ROB approach may be inefficient or simply unfeasible. Indeed, the dimension n of $\mathbb{V}$ (and, if applicable, $\mathbb{W}$) required so that the subspace approximation (5.17) delivers the desired accuracy may be too large in this case to enable a PROM to achieve its "compactness" (minimum storage requirement) and performance objectives. In this case, an alternative approach for addressing the parameter dependence of a PROM based on the concept of *pointwise* ROBs may be considered. In this other approach, a set of parameter vectors $\boldsymbol{\mu}_i$ is properly sampled in the parameter space $\mathcal{P}$ – for example, using also a greedy procedure – and a pointwise right ROB $\mathbb{V}_i = \mathbb{V}(\boldsymbol{\mu}_i)$ (and if applicable, a left ROB $\mathbb{W}_i = \mathbb{W}(\boldsymbol{\mu}_i)$) is constructed at each sampled parameter point $\boldsymbol{\mu}_i \in \mathcal{P}$. Section 5.4.1 presents an efficient computational strategy for exploiting this concept of pointwise ROBs in the linear setting.

Because the concept of a global ROB is feasible in both linear and nonlinear settings, but that of pointwise ROBs – which so far has been supported by interpolation at queried but unsampled parameter points – has been exploited to date only in the linear setting, the following assumption is made throughout the remainder of this chapter: Whenever the context is set to that of a parametric PROM, a right ROB $\mathbb{V}$ (and if applicable, the corresponding left ROB $\mathbb{W}$) is assumed to be a global ROB, unless otherwise specified (as in Section 5.4.1).

5.3 Exact precomputation-based methodologies

The solution approach for eliminating the computational bottlenecks associated with the repeated reconstructions of a PROM characterized in Section 5.1.2 as an exact approach is overviewed here in two different representative contexts: that defined by a special class of parametric, linear FOMs; and that defined by a special class of parametric or nonparametric but nonlinear FOMs. In both contexts, this approach consists in reorganizing the exact computation of the reduced matrices, tangent matrices, and vectors, as applicable, in two parts: a first part that is responsible for the aforementioned bottlenecks and can be precomputed offline; and a second part whose computational complexity is independent of N_h, and whose real-time evaluation is feasible.

5.3.1 Linear FOMs and efficient parameter-affine representation

Consider again the *linear* instance (5.11) of the parametric FOM (5.1). Note that whatever is the dependence of $\mathbb{M}_{N_h}$, $\mathbb{K}_{N_h}$, and $\mathbf{g}_{N_h}$ on the parameter vector $\boldsymbol{\mu} \in \mathcal{P}$, these two matrices and vector can always be expressed as follows:

$$\mathbb{M}_{N_h}(\boldsymbol{\mu}) = \mathbb{M}_{0_{N_h}} + \sum_{i=1}^{i_{\mathrm{M}}} m_i(\boldsymbol{\mu})\mathbb{M}_{i_{N_h}},$$

$$\mathbb{K}_{N_h}(\boldsymbol{\mu}) = \mathbb{K}_{0_{N_h}} + \sum_{i=1}^{i_{\mathrm{K}}} k_i(\boldsymbol{\mu})\mathbb{K}_{i_{N_h}}, \tag{5.18}$$

$$\mathbf{g}_{N_h}(t;\boldsymbol{\mu}) = \mathbf{g}_{0_{N_h}}(t) + \sum_{i=1}^{i_{\mathbf{g}}} g_i(\boldsymbol{\mu})\mathbf{g}_{i_{N_h}}(t),$$

where $\mathbb{M}_{i_{N_h}} \in \mathbb{R}^{N_h \times N_h}$, $i = 0, \ldots, i_{\mathrm{M}}$; $\mathbb{K}_{i_{N_h}} \in \mathbb{R}^{N_h \times N_h}$, $i = 0, \ldots, i_{\mathrm{K}}$; $\mathbf{g}_{i_{N_h}} \in \mathbb{R}^{N_h}$, $i = 0, \ldots, i_{\mathbf{g}}$; each of the scalar functions $m_i(\boldsymbol{\mu})$, $k_i(\boldsymbol{\mu})$, and $g_i(\boldsymbol{\mu})$ describes the dependence of $\mathbb{M}_{i_{N_h}}$, $\mathbb{K}_{i_{N_h}}$, and $\mathbf{g}_{i_{N_h}}$ on $\boldsymbol{\mu}$, respectively; and $i_{\mathrm{M}} \leq N_h{}^2$, $i_{\mathrm{K}} \leq N_h{}^2$, and $i_{\mathbf{g}} \leq N_h$.

In the context of the two-sided Petrov–Galerkin projection of the parametric, linear FOM (5.11), the parameter-affine representation (5.18) of the otherwise *arbitrary* dependence of this linear FOM on $\boldsymbol{\mu}$ is parameter-preserving, in the sense that it eases the reduction of the FOM (5.11) to a PROM that has the following similar form:

$$
\begin{cases}
\mathbb{M}_n(\boldsymbol{\mu})\dot{\mathbf{u}}_n(t;\boldsymbol{\mu}) + \mathbb{K}_n(\boldsymbol{\mu})\mathbf{u}_n(t;\boldsymbol{\mu}) = \mathbf{g}_n(t;\boldsymbol{\mu}), \\
\qquad\qquad\qquad \mathbf{u}_n(0;\boldsymbol{\mu}) = \mathbf{u}_n^0(\boldsymbol{\mu}), \\
\text{where} \\
\mathbb{M}_n(\boldsymbol{\mu}) = \underbrace{\mathbb{W}^T \mathbb{M}_{0_{N_h}} \mathbb{V}}_{\substack{\text{precomputable} \\ \in \mathbb{R}^{n \times n}}} + \sum_{i=1}^{i_{\mathrm{M}}} m_i(\boldsymbol{\mu}) \underbrace{\mathbb{W}^T \mathbb{M}_{i_{N_h}} \mathbb{V}}_{\substack{\text{precomputable} \\ \in \mathbb{R}^{n \times n}}}, \\
\mathbb{K}_n(\boldsymbol{\mu}) = \underbrace{\mathbb{W}^T \mathbb{K}_{0_{N_h}} \mathbb{V}}_{\substack{\text{precomputable} \\ \in \mathbb{R}^{n \times n}}} + \sum_{i=1}^{i_{\mathrm{K}}} k_i(\boldsymbol{\mu}) \underbrace{\mathbb{W}^T \mathbb{K}_{i_N} \mathbb{V}}_{\substack{\text{precomputable} \\ \in \mathbb{R}^{n \times n}}}, \\
\mathbf{g}_n(t;\boldsymbol{\mu}) = \underbrace{\mathbb{W}^T \mathbf{g}_{0_{N_h}}(t)}_{\substack{\text{precomputable} \\ \in \mathbb{R}^n}} + \sum_{i=1}^{i_{\mathbf{g}}} g_i(\boldsymbol{\mu}) \underbrace{\mathbb{W}^T \mathbf{g}_{i_{N_h}}(t)}_{\substack{\text{precomputable} \\ \in \mathbb{R}^n}}, \\
\mathbf{u}_n^0(\boldsymbol{\mu}) = \mathbb{W}^T \mathbf{u}_{N_h}^0(\boldsymbol{\mu}).
\end{cases}
\tag{5.19}
$$

From (5.19), it follows that all building blocks of the reduced quantities $\mathbb{M}_n$, $\mathbb{K}_n$, and $\mathbf{g}_n(t)$ – that is, $\{\mathbb{W}^T\mathbb{M}_i\mathbb{V}\}_{i=0}^{i_{\mathrm{M}}}$, $\{\mathbb{W}^T\mathbb{K}_i\mathbb{V}\}_{i=0}^{i_{\mathrm{K}}}$, and $\{\mathbb{W}^T\mathbf{g}_i(t)\}_{i=0}^{i_{\mathbf{g}}}$ – can be precomputed offline; for any queried parameter point $\boldsymbol{\mu}^\star \in \mathcal{P}$, the reduced matrices $\mathbb{M}_n(\boldsymbol{\mu}^\star)$ and $\mathbb{K}_n(\boldsymbol{\mu}^\star)$ can be computed in $\mathcal{O}(i_{\mathrm{M}}n^2)$ and $\mathcal{O}(i_{\mathrm{K}}n^2)$ operations, respectively; and for any queried parameter point $\boldsymbol{\mu}^\star \in \mathcal{P}$, the reduced vector $\mathbf{g}_n(t;\boldsymbol{\mu}^\star)$ can be computed in $\mathcal{O}(i_{\mathbf{g}}n)$ operations.

Hence, the reduced quantities $\mathbb{M}_n(\boldsymbol{\mu}^\star)$, $\mathbb{K}_n(\boldsymbol{\mu}^\star)$, and $\mathbf{g}_n(t;\boldsymbol{\mu}^\star)$ – and therefore the parametric, linear PROM $(\mathbb{M}_n(\boldsymbol{\mu}^\star), \mathbb{K}_n(\boldsymbol{\mu}^\star), \mathbf{g}_n(t;\boldsymbol{\mu}^\star))$ – can be computed in real-time only if

$$i_{\mathrm{M}} \ll N_h^2, \quad i_{\mathrm{K}} \ll N_h^2, \quad \text{and} \quad i_{\mathbf{g}} \ll N_h, \tag{5.20}$$

in which case the linear FOM (5.11) is said to admit an *efficient* parameter-affine representation. In other words, the representation (5.18) and the conditions (5.20) define the class of parametric, linear FOMs whose associated Petrov–Galerkin $(\mathbb{W}, \mathbb{U})$ PROMs can be processed online and in real-time using the exact method for treating the parameter dependency highlighted in (5.19).

Remark 5.1. Given a linear problem with a complex parameter dependency, the conditions (5.20) may or may not hold, depending on the level of fidelity of the modeling of this dependency. For example, consider the case of the finite element (FE) modeling of a geometrically complex and highly heterogeneous mechanical structure. If the geometric and material properties of such a structure are homogenized, or at least piecewise homogenized, the conditions (5.20) will hold. On the other hand, if these properties are represented in the computational model with the highest possible level of fidelity and such a representation leads to an FE model where each element has different values of the geometric and/or material properties, conditions (5.20) will not hold. In this case, the modeling of the parameter dependency could be simplified to the extent where the additional modeling errors induced by such a simplification are of the same or a lower order than any other modeling error – and particularly, the PMOR error – and the conditions (5.20) will be satisfied, in order to enable real-time processing.

5.3.2 Nonlinear FOMs with polynomial dependence on the generalized coordinates

Consider next the parametric, *nonlinear* FOM (5.1) and its associated PROM (5.6). As discussed in Section 5.1.1, the real-time processing of this parametric, nonlinear PROM faces two computational bottlenecks: the reevaluation at each queried point $\boldsymbol{\mu}^\star \in \mathcal{P}$ of the reduced matrix $\mathbb{M}_n(\boldsymbol{\mu}^\star) = \mathbb{W}^T \mathbb{M}_{N_h}(\boldsymbol{\mu}^\star)\mathbb{V}$, which requires $\mathcal{O}(N_h{}^2 n)$ operations; and the evaluation at each explicit time step of the reduced, nonlinear, internal force vector (5.8), which necessitates $\mathcal{O}(N_h n)$ operations, or at each Newton iteration of each implicit time step of the associated Jacobian (5.9), which requires $\mathcal{O}(N_h{}^2 n)$ operations. The first bottleneck was addressed in Section 5.3.1, and an alternative methodology for evaluating in real-time the parametric, reduced mass matrix $\mathbb{M}_n(\boldsymbol{\mu}^\star)$ is presented in Section 5.4.1. This alternative technique is particularly effective when the conditions (5.20) are not satisfied.

Hence, attention is focused here on eliminating the second computational bottleneck recalled above. This bottleneck is more significant than the first one because it arises even in the context of nonparametric problems, as long as they are nonlinear.

To this end, let

$$
\begin{aligned}
\mathbf{b}_n\big(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}\big) &= \mathbf{f}_n\big(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}\big) - \mathbf{g}_n(t;\boldsymbol{\mu}) \\
&= \underbrace{\mathbb{W}^T \mathbf{f}_{N_h}\big(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}\big)}_{\mathbf{b}_{1_n}(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})} - \underbrace{\mathbb{W}^T \mathbf{g}_{N_h}(t;\boldsymbol{\mu})}_{\mathbf{b}_{2_n}(t;\boldsymbol{\mu})}
\end{aligned}
\tag{5.21}
$$

denote the reduced, nonlinear, time-dependent, force balance vector. The computation of this nonlinear PROM quantity can be organized around the computation of its two terms: that associated with the reduced, nonlinear, internal force vector $\mathbf{b}_{1_n}(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$; and that associated with the reduced, external force vector $\mathbf{b}_{2_n}(t; \boldsymbol{\mu})$. The potential for the evaluation online and in real-time of each of these two components of the reduced, nonlinear, force balance vector, and in the case of $\mathbf{b}_{1_n}(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$, that of its Jacobian with respect to $\mathbf{u}_n(t; \boldsymbol{\mu})$, is discussed below.

5.3.2.1 Real-time computation of the reduced nonlinear internal force vector and its Jacobian

For many, but not all, partial differential equation (PDE)-based applications, the non-linearity in the FOM (5.1) is *polynomial* in $\mathbf{u}_{N_h}$ of degree $d \geq 2$. In this case, each i-th entry of the nonlinear, internal force vector can be written, for example, as follows:

$$
\begin{aligned}
\left[\mathbf{f}_{N_h}\big(\mathbf{u}_{N_h}(t; \boldsymbol{\mu}); \boldsymbol{\mu}\big)\right]_i &= \mathbb{F}_{i_1}(\boldsymbol{\mu})\mathbf{u}_{N_h}(t; \boldsymbol{\mu}) \\
&+ \sum_{k=1}^{\lceil(d-1)/2\rceil} \big(\mathbf{u}_{N_h}^T(t; \boldsymbol{\mu})\,\mathbb{G}_{i_{2k}}(\boldsymbol{\mu})\mathbf{u}_{N_h}(t; \boldsymbol{\mu})\big)^k \\
&+ \sum_{k=1}^{\lceil(d-1)/2\rceil} \big(\mathbf{u}_{N_h}^T(t; \boldsymbol{\mu})\,\mathbb{G}_{i_{2k+1}}(\boldsymbol{\mu})\mathbf{u}_{N_h}(t; \boldsymbol{\mu})\big)^k \mathbb{F}_{i_{2k+1}}(\boldsymbol{\mu})\mathbf{u}_{N_h}(t; \boldsymbol{\mu}),
\end{aligned}
\tag{5.22}
$$

where $i = 1, \ldots, N_h$; $[\heartsuit]_i$ designates here and throughout the remainder of this chapter the entry in row i of the vector $\heartsuit$; $\mathbb{F}_{i_{2k+1}}(\boldsymbol{\mu}) \in \mathbb{R}^{1 \times N_h}$, $k = 0, \ldots, \lceil(d-1)/2\rceil$; $\mathbb{G}_{i_{2k}}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$; and $\mathbb{G}_{i_{2k+1}}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h \times N_h}$, $k = 1, \ldots, \lceil(d-1)/2\rceil$.

From the subspace approximation (5.17) and from (5.22), it follows that

$$
\begin{aligned}
\left[\mathbf{f}_{N_h}\big(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}\big)\right]_i &= \mathbb{F}_{i_1}(\boldsymbol{\mu})\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}) \\
&+ \sum_{k=1}^{\lceil(d-1)/2\rceil} \big(\mathbf{u}_n^T(t; \boldsymbol{\mu})\mathbb{V}^T\mathbb{G}_{i_{2k}}(\boldsymbol{\mu})\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu})\big)^k \\
&+ \sum_{k=1}^{\lceil(d-1)/2\rceil} \big(\mathbf{u}_n^T(t; \boldsymbol{\mu})\mathbb{V}^T\mathbb{G}_{i_{2k+1}}(\boldsymbol{\mu})\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu})\big)^k \mathbb{F}_{i_{2k+1}}(\boldsymbol{\mu})\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu})
\end{aligned}
\tag{5.23}
$$

is a polynomial function in $\mathbf{u}_n$ of the same degree d. Next, from (5.21) and (5.23), it follows that

$$
\left[\mathbf{b}_{1_n}\big(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}\big)\right]_i = \left[\mathbb{W}^T\mathbf{f}_{N_h}\big(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}\big)\right]_i = \sum_{j=1}^{N_h}[\mathbb{W}]_{ji}\mathbb{F}_{j_1}(\boldsymbol{\mu})\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu})
$$

$$
+ \sum_{j=1}^{N_h}[\mathbb{W}]_{ji} \sum_{k=1}^{\lceil(d-1)/2\rceil} \big(\mathbf{u}_n^T(t; \boldsymbol{\mu})\mathbb{V}^T\mathbb{G}_{j_{2k}}(\boldsymbol{\mu})\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu})\big)^k
$$

$$+ \sum_{j=1}^{N_h} [\mathbb{W}]_{ji} \sum_{k=1}^{\lceil (d-1)/2 \rceil} \big(\mathbf{u}_n^T(t; \boldsymbol{\mu}) \mathbb{V}^T \mathbb{G}_{j_{2k+1}}(\boldsymbol{\mu}) \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}) \big)^k \mathbb{F}_{j_{2k+1}}(\boldsymbol{\mu}) \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}),$$

where $[\diamond]_{ij}$ designates here and throughout the remainder of this chapter the entry in row i and column j of the matrix $\diamond$. The above expression for the i-th entry of the reduced, nonlinear, internal force vector can be rewritten as

$$\big[\mathbf{b}_{1_n}(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})\big]_i = \big[\mathbb{W}^T \mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})\big]_i = \sum_{j=1}^{N_h} [\mathbb{W}]_{ji} \underbrace{\mathbb{F}_{j_1}(\boldsymbol{\mu}) \mathbb{V}}_{\substack{\text{precomputable} \\ \in \mathbb{R}^{1 \times n}}} \mathbf{u}_n(t; \boldsymbol{\mu})$$

$$+ \sum_{j=1}^{N_h} \sum_{k=1}^{\lceil (d-1)/2 \rceil} [\mathbb{W}]_{ji} \bigg(\mathbf{u}_n^T(t; \boldsymbol{\mu}) \underbrace{\mathbb{V}^T \mathbb{G}_{j_{2k}}(\boldsymbol{\mu}) \mathbb{V}}_{\substack{\text{precomputable} \\ \in \mathbb{R}^{n \times n}}} \mathbf{u}_n(t; \boldsymbol{\mu}) \bigg)^k \tag{5.24}$$

$$+ \sum_{j=1}^{N_h} \sum_{k=1}^{\lceil (d-1)/2 \rceil} \bigg(\mathbf{u}_n^T(t; \boldsymbol{\mu}) \underbrace{\mathbb{V}^T \mathbb{G}_{j_{2k+1}}(\boldsymbol{\mu}) \mathbb{V}}_{\substack{\text{precomputable} \\ \in \mathbb{R}^{n \times n}}} \mathbf{u}_n(t; \boldsymbol{\mu}) \bigg)^k \underbrace{[\mathbb{W}]_{ji} \mathbb{F}_{j_{2k+1}}(\boldsymbol{\mu}) \mathbb{V}}_{\substack{\text{precomputable} \\ \in \mathbb{R}^{1 \times n}}} \mathbf{u}_n(t; \boldsymbol{\mu}).$$

This shows that each quantity arising in the evaluation of $\mathbf{b}_{1_n}(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ whose computational complexity scales with an integer power of the large dimension N_h of the FOM can be precomputed offline for each queried parameter point, and contributes to the nonlinear PROM (5.6) a term that can be processed online, in a number of operations that scales with an integer power of its small dimension $n \ll N_h$.

Expression (5.24) also shows that for a low-order polynomial nonlinearity in the internal force vector – say, a quadratic nonlinearity ($d = 2$) – and a queried parameter point $\boldsymbol{\mu}^\star \in \mathcal{P}$, $\mathbf{b}_{1_n}(\mathbf{u}_n(t; \boldsymbol{\mu}^\star); \boldsymbol{\mu}^\star)$ can be computed online and in real-time. However, for polynomial nonlinearities of higher degrees ($d > 2$), the offline cost associated with precomputing the various projections identified in (5.24) becomes prohibitively expensive and the real-time evaluation of $\mathbf{b}_{1_n}(\mathbf{u}_n(t; \boldsymbol{\mu}^\star); \boldsymbol{\mu}^\star)$ becomes less likely.

Note that if the reduced, nonlinear, internal force vector $\mathbf{b}_{1_n}(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ is a polynomial function of $\mathbf{u}_n$ of degree d, the Jacobian of $\mathbf{b}_{1_n}$ with respect to $\mathbf{u}_n$ – and therefore the Jacobian of the reduced, nonlinear, force balance vector $\mathbf{b}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ (5.21) with respect to $\mathbf{u}_n$ – is a polynomial function of $\mathbf{u}_n$ of degree $d-1$. For this reason, all conclusions stated above regarding the real-time evaluation of $\mathbf{b}_{1_n}(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ equally apply to the real-time evaluation of the Jacobian of $\mathbf{b}_{1_n}(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ with respect to $\mathbf{u}_n$ – and therefore the Jacobian of the reduced, nonlinear, force balance vector $\mathbf{b}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ (5.21) with respect to $\mathbf{u}_n$.

The practical significance of the offline-online methodology highlighted in (5.24) for computing in real-time the reduced, nonlinear, internal force vector of a nonlinear PROM is underscored by its availability in many software products, including Vega, the physics-based computer graphics library for the simulation of the dynamics of three-dimensional deformable objects [8].

Remark 5.2. In general, the reduced, nonlinear, internal force vector exhibits a low-order polynomial dependence on the vector of generalized coordinates $\mathbf{u}_n$ if the nonlinear, internal force vector exhibits a low-order polynomial dependence on the vector of DOFs $\mathbf{u}_{N_h}$. In practice, whether the latter holds true or not may depend on the choice of DOFs. For example in computational fluid dynamics, the discretization of the primitive form of the Navier–Stokes equations using primitive variables leads to a quadratic dependence of the nonlinear, internal force vector on these variables. On the other hand, the discretization of the conservative form of these equations using conservative variables cannot lead to such a dependence.

Now, it is noteworthy to mention that if the external force vector is fixed in time, $\mathbf{b}_{2_n} = \mathbf{b}_{2_n}(\boldsymbol{\mu})$ can be precomputed offline (see (5.21)). In this case, the low-order polynomial dependence on the solution vector examplified in (5.22) – with $d = 2$ – defines the class of nonlinear FOMs for which, given a queried parameter point $\boldsymbol{\mu}^\star \in \mathcal{P}$, the computational bottlenecks associated with the evaluation of the reduced, nonlinear, force balance vector (5.21) and/or its Jacobian can be eliminated using the exact, precomputation-based methodology described in this section.

5.3.2.2 Real-time computation of the reduced time-dependent external force vector

If the external force vector is time-dependent, $\mathbf{b}_{2_n} = \mathbf{b}_{2_n}(t; \boldsymbol{\mu})$, and the real-time computation of this reduced vector requires special attention.

To this effect, it is noted that in many, but not all, PDE-based applications, the time-dependent, external force vector can be divided into spatial and temporal components as follows:

$$\mathbf{g}_{N_h}(t; \boldsymbol{\mu}) = \mathbb{B}(\boldsymbol{\mu})\,\mathbf{a}_m(t; \boldsymbol{\mu}), \quad \text{where } \mathbb{B} \in \mathbb{R}^{N_h \times m},\ \mathbf{a}_m \in \mathbb{R}^m,\ m \le N_h, \tag{5.25}$$

$\mathbb{B}$ is a Boolean matrix that specifies the nonzero entries of $\mathbf{g}_{N_h}(t; \boldsymbol{\mu})$, and $\mathbf{a}_m(t; \boldsymbol{\mu})$ is an amplitude vector that stores their time histories. In this case, the reduced, time-dependent, external force vector can be computed as follows:

$$\mathbf{b}_{2_n}(t; \boldsymbol{\mu}) = \underbrace{\mathbb{W}^T \mathbb{B}(\boldsymbol{\mu})}_{\substack{\text{precomputable} \\ \in \mathbb{R}^{n \times m}}}\ \mathbf{a}_m(t; \boldsymbol{\mu}).$$

Hence, if

$$\mathbf{g}_{N_h}(t; \boldsymbol{\mu}) = \mathbb{B}(\boldsymbol{\mu})\,\mathbf{a}_m(t; \boldsymbol{\mu}), \quad \mathbb{B} \in \mathbb{R}^{N_h \times m},\ \mathbf{a}_m \in \mathbb{R}^m,\ \text{and } \underline{m \ll N_h}, \tag{5.26}$$

where $m \ll N_h$ can be interpreted as m is of the order of n, $\mathbf{b}_{2_n}(t; \boldsymbol{\mu})$ can be efficiently computed online and in real-time by precomputing offline its time-independent part $\mathbb{W}^T \mathbb{B}(\boldsymbol{\mu})$.

5.3.2.3 Real-time computation of the reduced nonlinear force balance vector and its Jacobian

Consider again the low-order polynomial dependence on the solution vector exemplified in (5.22) – with $d = 2$ – and the spatial-temporal multiplicative decomposition conditions (5.25) and (5.26). Collectively, these conditions define the class of nonlinear FOMs for which, given a queried parameter point $\boldsymbol{\mu}^\star \in \mathcal{P}$, the computational bottlenecks associated with the evaluation of the reduced, nonlinear, force balance vector (5.21) and/or its Jacobian can be eliminated using the exact, precomputation-based methodologies described in Sections 5.3.2.1 and 5.3.2.2.

5.3.2.4 Real-time treatment of a parametric dependence in the reduced nonlinear force balance vector and its Jacobian

There are two main cases to anticipate here:
- Case 1: The interest is in a suite of nonlinear simulations, where the parameter point $\boldsymbol{\mu} \in \mathcal{P}$ is fixed within a simulation but varies across the simulations.
- Case 2: The interest is in a single nonlinear simulation, where either or both of the following issues must be dealt with:
    - Case 2a: $\boldsymbol{\mu}$ is configuration-dependent and therefore may vary when the solution vector $\mathbf{u}_{N_h}(t; \boldsymbol{\mu})$ – and hence, the vector of generalized coordinates $\mathbf{u}_n(t; \boldsymbol{\mu})$, the nonlinear, internal force vector $\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$, and the reduced, nonlinear, force balance vector $\mathbf{b}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ (5.21) – vary within the simulation. This is the case when, for example, the FOM (5.1) is associated with a structural dynamics or solid mechanics problem, $\boldsymbol{\mu}$ contains one or several shape parameters, and the deformations are sufficiently large to change the shape of the computational domain at each time step.
    - Case 2b: $\boldsymbol{\mu}$ is varied in time – for example, in the external force vector $\mathbf{g}_{N_h}(t; \boldsymbol{\mu})$ and therefore in the reduced, external force vector $\mathbf{g}_n(t; \boldsymbol{\mu})$ and the reduced, nonlinear, force balance vector $\mathbf{b}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ (5.21). This is the case when, for example, the FOM (5.1) is associated again with a structural dynamics or solid mechanics problem, μ contains one or several shape parameters, the deformations are sufficiently large to change the shape of the computational domain at each time step, and $\mathbf{g}_{N_h}(t; \boldsymbol{\mu})$ is a pressure-induced, time-dependent, external force vector.

In Case 1, the exact, precomputation-based methodologies can be applied as described in Sections 5.3.2.1 and 5.3.2.2. In Case 2a, the parameter dependency in the reduced matrices $[\mathbb{W}]_{ji}\mathbb{F}_{j_1}(\boldsymbol{\mu})\mathbb{V}$, $[\mathbb{W}]_{ji}\mathbb{F}_{j_{2k+1}}(\boldsymbol{\mu})\mathbb{V}$, $\mathbb{V}^T\mathbb{G}_{j_{2k}}(\boldsymbol{\mu})\mathbb{V}$, and $\mathbb{V}^T\mathbb{G}_{j_{2k+1}}(\boldsymbol{\mu})\mathbb{V}$ (see (5.24)) must be efficiently treated: This can be done using the method of interpolation on matrix manifolds described in Section 5.4.1. Similarly in Case 2b, the parameter

dependency in the reduced, external force vector can be efficiently treated using the interpolation method described in Section 5.4.1.

5.4 Approximate reconstruction methodologies

Within the inexact approach developed so far for eliminating the computational bottlenecks associated with the repeated constructions of the reduced quantities defining a PROM, two methodologies stand out from the rest. The first one specifically targets parametric, linear PROMs. The second methodology is equally applicable to parametric and nonparametric, linear and nonlinear PROMs.

5.4.1 Database of linear PROMs and interpolation on matrix manifolds

For a sufficiently large and/or high-dimensional bounded parameter space $\mathcal{P}$, and/or in the presence of large sensitivities of the parametric FOM with respect to $\boldsymbol{\mu}$, the dimension n of the global right ROB $\mathbb{V}$ for which the subspace approximation (5.17) is sufficiently accurate may be too large to enable a PMOR to be "compact" and/or perform in real-time. Furthermore, if for a parametric, linear FOM the conditions (5.20) are not satisfied, the reconstruction in real-time of the PROM (5.19) at a queried but unsampled point $\boldsymbol{\mu}^\star \in \mathcal{P}$ may not be feasible. For these reasons, an alternative methodology has also been developed in the literature for eliminating the computational bottlenecks associated with the efficient processing of parametric, linear PROMs.

Specifically, an alternative methodology was proposed in [1] for efficiently treating the dependence of a linear PROM on a parameter vector. It consists in: generating offline a database of pointwise, linear PROMs; equipping it with a family of algorithms for interpolation on matrix manifolds; and applying these algorithms online to build in real-time a PROM at a queried but unsampled point $\boldsymbol{\mu}$ of the parameter space $\mathcal{P}$. Subsequently, this alternative methodology was fully developed in [3] and demonstrated for realistic fluid, structure, and fluid–structure interaction problems. Recently, it was refined in [5]. In the context of the parametric, linear PROM (5.12) – which can be defined by the triplet $(\mathbb{M}_n(\boldsymbol{\mu}), \mathbb{K}_n(\boldsymbol{\mu}), \mathbf{g}_n(t; \boldsymbol{\mu}))$ – this methodology for treating the parameter dependency operates as follows:

1. **Offline** (divide)
   – Apply an appropriate greedy procedure to sample $n_{\boldsymbol{\mu}}$ points $\boldsymbol{\mu}_i$ in the parameter space $\mathcal{P}$.
   – At each sampled point $\boldsymbol{\mu}_i \in \mathcal{P}$, $i = 1, \ldots, n_{\boldsymbol{\mu}}$, apply any preferred technique such as, for example, the proper orthogonal decomposition (POD) method of snapshots [35] in the case of a linear Galerkin-PROM, or the balanced POD

method in the case of a linear Petrov–Galerkin PROM [40], to construct

$$
\begin{aligned}
\mathbb{M}_n^i = \mathbb{M}_n(\boldsymbol{\mu}_i) &= \mathbb{W}_i^T \mathbb{M}_{N_h}^i \mathbb{V}_i = \mathbb{W}^T(\boldsymbol{\mu}_i)\mathbb{M}_{N_h}(\boldsymbol{\mu}_i)\mathbb{V}(\boldsymbol{\mu}_i), \\
\mathbb{K}_n^i = \mathbb{K}_n(\boldsymbol{\mu}_i) &= \mathbb{W}_i^T \mathbb{K}_{N_h}^i \mathbb{V}_i = \mathbb{W}^T(\boldsymbol{\mu}_i)\mathbb{K}_{N_h}(\boldsymbol{\mu}_i)\mathbb{V}(\boldsymbol{\mu}_i), \\
\text{and} & \\
\mathbf{g}_n^i(t) = \mathbf{g}_n(t; \boldsymbol{\mu}_i) &= \mathbb{W}_i^T \mathbf{g}_{N_h}^i(t) = \mathbb{W}^T(\boldsymbol{\mu}_i)\mathbf{g}_{N_h}(t; \boldsymbol{\mu}_i).
\end{aligned}
\tag{5.27}
$$

Here, the notation $\mathbb{W}_i = \mathbb{W}(\boldsymbol{\mu}_i)$ ($\mathbb{V}_i = \mathbb{V}(\boldsymbol{\mu}_i)$) specifies that the left (right) ROB is constructed at the point $\boldsymbol{\mu}_i \in \mathcal{P}$ and that in this sense, it is a pointwise ROB.

–   Store the set of precomputed reduced matrices $\{\mathbb{M}_n^i\}_{i=1}^{n_\mu}$ in a database $\mathcal{D}_{\mathbb{M}}$ of pointwise, linear $\mathbb{M}_n$-PROMs, the set of precomputed reduced matrices $\{\mathbb{K}_n^i\}_{i=1}^{n_\mu}$ in a counterpart database $\mathcal{D}_{\mathbb{K}}$, and the set of precomputed reduced vectors $\{\mathbf{g}_n^i(t)\}_{i=1}^{n_\mu}$ in a counterpart database $\mathcal{D}_{\mathbf{g}}$.

2. **Online** (conquer)
    –   *For each queried but unsampled point $\boldsymbol{\mu}^\star \in \mathcal{P}$, construct $\mathbb{M}_n^\star = \mathbb{M}_n(\boldsymbol{\mu}^\star)$, $\mathbb{K}_n^\star = \mathbb{K}_n(\boldsymbol{\mu}^\star)$, and $\mathbf{g}_n^\star(t) = \mathbf{g}_n(t; \boldsymbol{\mu}^\star)$ as follows:*
    –   Identify a priori three matrix manifolds $\mathcal{M}_{\mathbb{M}}$, $\mathcal{M}_{\mathbb{K}}$, and $\mathcal{M}_{\mathbf{g}}$ on which $\mathbb{M}_n$, $\mathbb{K}_n$, and $\mathbf{g}_n(t)$ lie, respectively, by identifying the most important algebraic property characterizing each of these matrices as explained below (note that by default, any $p \times q$ real-valued matrix belongs to the manifold $\mathbb{R}^{p \times q}$).
    –   Interpolate in real-time the precomputed reduced matrices $\mathbb{M}_n^i$ (5.27) on $\mathcal{M}_{\mathbb{M}}$, the precomputed reduced matrices $\mathbb{K}_n^i$ (5.27) on $\mathcal{M}_{\mathbb{K}}$, and the precomputed reduced vectors $\mathbf{g}_n^i(t)$ (5.27) on $\mathcal{M}_{\mathbf{g}}$.

The first version of the methodology described above for treating efficiently the parameter dependence of a linear PROM was developed in the context of linear/linearized Galerkin-PROMs [2]. It focused on: precomputing a set of pointwise right ROBs $\{\mathbb{V}_i\}_{i=1}^{n_\mu}$ *of the same dimension $N_h \times n$*; storing these pointwise ROBs in a database $\mathcal{D}_{\mathbb{V}}$; and for each queried but unsampled point $\boldsymbol{\mu}^\star \in \mathcal{P}$, interpolating the precomputed set of pointwise right ROBs to construct $\mathbb{V}_{\boldsymbol{\mu}^\star} = \mathbb{V}(\boldsymbol{\mu}^\star)$. The variant methodology outlined above and first proposed in [1, 3] interpolates directly the pointwise PROMs $\{(\mathbb{M}_n^i, \mathbb{K}_n^i, \mathbf{g}_n^i(t))\}_{i=1}^{n_\mu}$ (5.27) – instead of their underlying ROBs $\{\mathbb{V}_i, \mathbb{W}_i\}_{i=1}^{n_\mu}$ – in order to construct the pointwise PROM $(\mathbb{M}_n^\star, \mathbb{K}_n^\star, \mathbf{g}_n^\star(t))$ at $\boldsymbol{\mu}^\star \in \mathcal{P}$. Hence, this variant offers the following added benefits:

1. It replaces the interpolation requirement that all precomputed pointwise ROBs $\mathbb{V}_i$ have the same large and small dimensions N_h and n, respectively, by the relatively lesser requirement that all PROMs $\{(\mathbb{M}_n^i, \mathbb{K}_n^i, \mathbf{g}_n^i(t))\}_{i=1}^{n_\mu}$ have the same dimension n. Indeed, while the former requirement restricts the construction of the pointwise FOMs $\{(\mathbb{M}_{N_h}^i, \mathbb{K}_{N_h}^i, \mathbf{g}_{N_h}^i(t))\}_{i=1}^{n_\mu}$ to the same mesh or to topologically identical meshes in order to guarantee the same large dimension N_h for each FOM $(\boldsymbol{\mu}_i)$, the latter requirement allows these FOMs to have different dimensions $(N_h)_i$. Therefore, it frees their construction on different meshes with different sizes [5].

2. For each queried but unsampled point $\boldsymbol{\mu}^{\star} \in \mathcal{P}$, it eliminates the need to form explicitly the matrix-matrix-matrix products $\mathbb{W}_{\star}^{T} \mathbb{M}_{N_h}^{\star} \mathbb{V}_{\star}$ and $\mathbb{W}_{\star}^{T} \mathbb{K}_{N_h}^{\star} \mathbb{V}_{\star}$ after the right ROB $\mathbb{V}_{\star}$ has been computed in real-time by interpolation of the set of point-wise, right ROBs $\{\mathbb{V}_i\}_{i=1}^{n_\mu}$. This is a critical advantage over the first version [2] based on the interpolation of ROBs, as the computational complexity of each of the aforementioned triple matrix product grows as $\mathcal{O}(N_h^2 n)$.

Each of the reduced matrices $\mathbb{M}_n$ and $\mathbb{K}_n$ and the reduced vector $\mathbf{g}_n(t)$ can be characterized by identifying a priori the most relevant manifold it belongs to. For example, if $\mathbb{M}_{N_h}$ is SPD and a Galerkin projection is chosen ($\mathbb{W} = \mathbb{V}$), $\mathbb{M}_n$ belongs to three different manifolds: the manifold of $n{\times}n$ real matrices, $\mathbb{R}^{n\times n}$; the manifold of invertible matrices of size n, GL (n); and the manifold of SPD matrices of size n, SPD (n). In this case, SPD (n) is the most relevant manifold as any matrix that lies on it is real-valued, of size n, and nonsingular. On the other hand, $\mathbf{g}_n(t)$ belongs to the manifold of $n{\times}1$ matrices. The main objective of interpolation on a matrix manifold is to preserve such a characterization during the interpolation process, so that at the queried but unsampled point $\boldsymbol{\mu}^{\star}$, the interpolated PROM ($\mathbb{M}_n^{\star}, \mathbb{K}_n^{\star}, \mathbf{g}_n^{\star}(t)$) inherits this characterization. Otherwise, the standard (coefficient-by-coefficient) interpolation method does not guarantee this objective, except for the manifold $\mathbb{R}^{p\times q}$ – that is, for the case of $\mathbf{g}_n(t; \boldsymbol{\mu})$ ($p = n$, $q = 1$), where the characterization $\mathbf{g}_n(t; \boldsymbol{\mu}) \in \mathbb{R}^n$ is not particularly specific.

Let $\{\mathbb{A}_n^i\}_{i=1}^{n_\mu}$ denote the set of matrices to be interpolated on a matrix manifold $\mathcal{M}$ on which they lie. For example:

1. $\mathbb{A}_n^i = \mathbb{M}_n^i \Rightarrow \mathcal{M} =$ SPD (n), GL (n), or simply $\mathbb{R}^{n\times n}$.
2. $\mathbb{A}_n^i = \mathbb{K}_n^i \Rightarrow \mathcal{M} =$ SPD (n), GL (n), or simply $\mathbb{R}^{n\times n}$.
3. $\mathbb{A}_n^i = \mathbf{g}_n^i(t) \Rightarrow \mathcal{M} = \mathbb{R}^{n\times 1}$.

Among these matrices, choose $\mathbb{A}_n^j$ as a reference point on $\mathcal{M}$. Let G_n denote an element of the tangent space $\mathcal{T}_{\mathbb{A}_n^j} \mathcal{M}$ to $\mathcal{M}$ at $\mathbb{A}_n^j$, and let $\mathbb{B}_n \in \mathcal{M}$ denote a point on $\mathcal{M}$ in a neighborhood of $\mathbb{A}_n^j$. Whichever matrix manifold $\mathcal{M}$ is chosen to support the interpolation to be performed and whichever reduced matrix $\mathbb{A}_n^j$ is chosen as a reference point on this manifold, the set of matrices $\{\mathbb{A}_n^i\}_{i=1}^{n_\mu}$ can be interpolated on $\mathcal{M}$ using the method described in Algorithm 5.1 and graphically depicted in Figure 5.1.

Essentially, Algorithm 5.1 starts by "moving" the set of matrices $\mathbb{A}_n^i$, $i = 1, \ldots, n_\mu$ but $i \neq j$, to the tangent space to the matrix manifold $\mathcal{M}$ at the chosen reference point $\mathbb{A}_n^j$, $\mathcal{T}_{\mathbb{A}_n^j} \mathcal{M}$, using the *logarithmic mapping* $\text{Log}_{\mathbb{A}_n^j}(\mathbb{A}_n^i)$. This leads to the set of matrices $G_n^i = G_n(\boldsymbol{\mu}_i)$, $i = 1, \ldots, n_\mu$. Since $\mathcal{T}_{\mathbb{A}_n^j} \mathcal{M}$ is a linear vector space, Algorithm 5.1 applies standard interpolation in this space to obtain the interpolated matrix $G_n^{\star} = G_n(\boldsymbol{\mu}^{\star})$ at the queried but unsampled point $\boldsymbol{\mu}^{\star} \in \mathcal{P}$. Next, it moves back $G_n^{\star}$ to $\mathcal{M}$ using the *exponential mapping* $\text{Exp}_{\mathbb{A}_n^j}(G_n^{\star})$ to deliver the desired matrix $\mathbb{A}_n^{\star} = \mathbb{A}_n(\boldsymbol{\mu}^{\star})$.

Table 5.1 gives the expressions of the logarithmic and exponential mappings for each of the example matrix manifolds mentioned above. As shown in [3], these

---

Algorithm 5.1: Interpolation of $\{\mathbb{A}_n^i\}_{i=1}^{n_\mu}$ on a matrix manifold $\mathcal{M}$.

Input: n_μ matrices $\mathbb{A}_n^1, \ldots, \mathbb{A}_n^{n_\mu}$ lying on $\mathcal{M}$
Output: Interpolated matrix $\mathbb{A}_n^\star = \mathbb{A}_n(\boldsymbol{\mu}^\star)$ at a queried but unsampled point $\boldsymbol{\mu}^\star \in \mathcal{P}$

1: Choose $j \in 1, \ldots, n_\mu$ {the interpolation process occurs in the linear vector space $\mathcal{T}_{\mathbb{A}_n^j} \mathcal{M}$}
2: **for** $i = 1, \ldots, n_\mu$ **do**
3:     Compute $G_n^i = \mathrm{Log}_{\mathbb{A}_n^j}(\mathbb{A}_n^i)$
4: **end for**
5: Interpolate independently each entry of the matrices G_n^i, $i = 1, \ldots, n_\mu$ in order to obtain $G_n^\star = G_n(\boldsymbol{\mu}^\star)$
6: Compute $\mathbb{A}_n^\star = \mathrm{Exp}_{\mathbb{A}_n^j}(G_n^\star)$
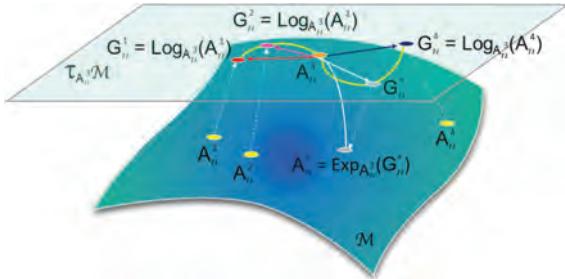
Figure 5.1: Interpolation of a set of matrices $\{\mathbb{A}_n^i\}_{i=1}^{n_\mu}$ on a matrix manifold $\mathcal{M}$.

Table 5.1: Logarithm and exponential mappings for some matrix manifolds $\mathcal{M}$.

$\mathcal{M}$	$\mathbb{R}^{M \times N}$	GL (n)	SPD (n)
$\mathrm{Log}_{\mathbb{A}_n}(\mathbb{B}_n)$	$\mathbb{B}_n - \mathbb{A}_n$	$\log(\mathbb{B}_n \mathbb{A}_n^{-1})$	$\log(\mathbb{A}_n^{-1/2} \mathbb{B}_n \mathbb{A}_n^{-1/2})$
$\mathrm{Exp}_{\mathbb{A}_n}(G)$	$\mathbb{A}_n + G$	$\exp(G)\mathbb{A}_n$	$\mathbb{A}_n^{1/2} \exp(G)\mathbb{A}_n^{1/2}$

mappings and the entire Algorithm 5.1 can be implemented in real-time. For high-dimensional parameter spaces $\mathcal{P}$, standard interpolation in $\mathcal{T}_{\mathbb{A}_n^j} \mathcal{M}$ is perhaps most conveniently performed using radial basis functions [34].

Remark 5.3. Here, two comments are noteworthy. First, the systems of generalized coordinates underlying the linear PROMs stored in a database must be coherent, in order to enable the coherent interpolation of these PROMs. This requirement can be enforced offline, using the simple postprocessing algorithm described in [3]. Second, Algorithm 5.1 is not guaranteed in principle to preserve the numerical stability of the

interpolated PROMs. While this has never been observed to be an issue, the real-time stabilization algorithm described in [4] can be applied to the interpolated PROM to guarantee its numerical stability.

5.4.2 Hyperreduction

Finally, consider the most general case where any of the following conditions holds:
- The parametric, nonlinear FOM (5.1) is not characterized by a low-order polynomial nonlinearity in the internal force vector $\mathbf{f}_{N_h}(\mathbf{u}_{N_h}(t; \boldsymbol{\mu}); \boldsymbol{\mu})$.
- The parameter-affine representation (5.18) of the linear instance (5.11) of this high-dimensional computational model does not satisfy the conditions (5.20).
- The parameter vector $\boldsymbol{\mu}$ characterizing the nonlinear FOM (5.1) or its linear instance (5.11) may vary within a single simulation for any of the reasons explained in Section 5.3.2.4, or any similar or related reason.

In this case, the exact, precomputation-based methodologies described in Sections 5.3.1 and 5.3.2 cannot be applied to eliminate the computational bottlenecks associated with the repeated reconstructions of the reduced matrices or tangent matrices, and/or repeated evaluations of the reduced vectors that may arise during the processing of parametric, linear PROMs, and parametric or nonparametric, nonlinear PROMs. Furthermore, there does not seem to be a clear path that efficiently extends the inexact approach based on a database of pointwise, linear PROMs described in Section 5.4.1 to arbitrarily nonlinear problems. For all these reasons, another family of inexact methods has been developed for addressing in the most general case outlined above the aforementioned computational bottlenecks. Here, these methods are collectively referred to as *hyperreduction* methods – or the hyperreduction paradigm – even though their common underlying idea had emerged well before the word "hyperreduction" was coined in [33].

Hyperreduction methods mitigate or eliminate the computational bottlenecks associated with the repeated reconstructions of projection-based reduced-order operators by approximating these operators using a computational complexity that is independent of the dimension N_h of the high-dimensional FOM. Hence, they trade some of the accuracy achieved by a PROM for speed. They can be classified in two categories: the *approximate-then-project methods*, which appeared first in the literature; and the *project-then-approximate* methods, which were developed more recently.

As suggested by their label, approximate-then-project hyperreduction methods approximate first an operator of interest and then project the performed approximation on the left ROB $\mathbb{W}$ (Petrov–Galerkin-PROM) or $\mathbb{V}$ (Galerkin-PROM). Their underlying common idea for avoiding a computational complexity that scales with the high dimension N_h of the problem can be traced back to the gappy POD method [20]. This idea, which was originally developed for image reconstruction, can be described as

follows. First, the operator to be reduced – for example, in this case the nonlinear, internal force vector $\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ – is approximated using a small number $m \ll N_h$ of empirically derived basis functions, where m is not necessarily related to the dimension n of the PROM to be hyperreduced. This can be written as $\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) \approx \hat{\mathbf{f}}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \mathbb{U}\mathbf{f}_m(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$, where $\mathbb{U} \in \mathbb{R}^{N_h \times m}$ stores in its columns the empirically derived basis functions and $\mathbf{f}_m(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) \in \mathbb{R}^m$ denotes the vector of reduced or generalized coordinates of this approximation. This vector is computed such as to minimize the error of this approximation at a small number of computed rows of $\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$ – that is, if $\mathcal{I}$ represents the set of computed rows $|\mathcal{I}| \ll N_h$, $\mathbf{f}_m^{\text{opt}}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \arg\min \|\mathbf{f}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) - \mathbb{U}^{\mathcal{I}}\mathbf{f}_m(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})\|_2 = (\mathbb{U}^{\mathcal{I}})^{\dagger}\mathbf{f}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$, where $\spadesuit^{\mathcal{I}}$ designates the restriction of the vector or matrix $\spadesuit$ to its rows specified by the elements of $\mathcal{I}$; $\mathbf{f}_{N_h}^{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$; $\mathbb{U}^{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}| \times m}$; the superscript † designates the Moore–Penrose pseudo-inverse; and therefore $(\mathbb{U}^{\mathcal{I}})^{\dagger} \in \mathbb{R}^{m \times |\mathcal{I}|}$. Note that if the number of empirical basis functions m is equal to $|\mathcal{I}|$, the optimal approximation of $\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$ becomes $\hat{\mathbf{f}}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \mathbb{U}\mathbf{f}_m^{\text{opt}}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \mathbb{U}(\mathbb{U}^{\mathcal{I}})^{-1}\mathbf{f}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$.

Then, assuming the general case of a Petrov–Galerkin projection, the hyperreduced, nonlinear, internal force vector is computed as $\widetilde{\mathbf{f}}_n(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \mathbb{W}^T\hat{\mathbf{f}}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \mathbb{W}^T\mathbb{U}(\mathbb{U}^{\mathcal{I}})^{\dagger}\mathbf{f}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$, where the tilde symbol applied to an algebraic quantity designates here and throughout the remainder of this chapter the hyperreduction of this quantity. The hyperreduced vector $\widetilde{\mathbf{f}}_n(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$ is reconstructed each time t and/or μ is varied by: precomputing offline the matrix-matrix product $\mathbb{Q} = \mathbb{W}^T\mathbb{U}(\mathbb{U}^{\mathcal{I}})^{\dagger} \in \mathbb{R}^{n \times |\mathcal{I}|}$; and reconstructing online the approximation $\widetilde{\mathbf{f}}_n(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \mathbb{Q}\mathbf{f}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$ in $\mathcal{O}(n|\mathcal{I}|)$ operations, where $n \ll N_h$ and $|\mathcal{I}| \ll N_h$.

The empirical interpolation method (EIM) introduced in [9, 26] – that is, almost a decade after the gappy POD method – and a variant of its discrete version known as the discrete EIM (DEIM) [15], as well as many other hyperreduction methods, including the missing point estimation approach [7], follow the same general idea, albeit for PDE-based applications. For this reason, the EIM is, unlike gappy POD, grounded in the continuum level. For elliptic problems, it enjoys some level of theoretical support. On the other hand, DEIM is arguably the most popular approximate-then-project hyperreduction method to date, due perhaps to its black-box algebraic formulation. A version of this method tailored to FE approximations was recently described in [38] under the name "unassembled discrete empirical interpolation method" (UDEIM). Another noteworthy approximate-then-project hyperreduction method is the Gauss–Newton with approximated tensors (GNAT) method introduced in [12]. GNAT is also related to the gappy POD method. However, unlike EIM and DEIM, GNAT was conceived from the beginning for the Petrov–Galerkin rather than the Galerkin framework of model reduction. As such, it has a few distinctive features. More importantly, GNAT has been

successfully applied to the real-time solution of large-scale, three-dimensional, turbulent flow problems of industrial relevance [13].

On the other hand, project-then-approximate hyperreduction methods approximate directly the projection on the left ROB $\mathbb{W}$ or $\mathbb{V}$ of an operator such as $\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$. They avoid a computational complexity that scales with N_h by avoiding the construction in this case of $\mathbf{f}_{N_h}$, and the matrix-vector product $\mathbb{W}^T\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^n$ or $\mathbb{V}^T\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^n$. Among this family of hyperreduction methods, the two most notable ones are the cubature-based approximation method developed in [6] for computer graphics applications, and the energy-conserving sampling and weighting (ECSW) method developed in [22], characterized in [24] for second-order dynamical systems, and parallelized in [14]. Essentially, such methods proceed in two steps. First, they sample offline a given high-dimensional mesh to construct a reduced mesh, and attribute to each sampled element a weighting coefficient that they determine by solving a suitable optimization problem. Then, they approximate online each projection-based reduced-order operator to be reconstructed using a quadrature rule determined by the reduced mesh and its associated weights. For second-order dynamical systems such as those arising in wave propagation, solid mechanics, and structural dynamics applications, ECSW is to date the only known project-then-approximate hyperreduction method with a provable structure-preserving property. For such problems, it specifically preserves the Lagrangian structure associated with Hamilton's principle. As such, ECSW can preserve the numerical stability properties of a discrete second-order dynamical system to which it is applied to, unlike the EIM, the DEIM, and GNAT.

Hence, because they represent the state of the art of hyperreduction and have demonstrated success at enabling, for practical applications, the wall-clock time reduction factors expected from PMOR, the remainder of this chapter focuses on the description and discussion of: the EIM, the DEIM, and GNAT within the category of approximate-then-project hyperreduction methods; and ECSW within the category of project-then-approximate counterpart methods.

5.4.2.1 Approximate-then-project hyperreduction methods

First, the hyperreduction of $\boldsymbol{\mu}$-dependent vector quantities such as the external force vector $\mathbf{g}_{N_h}(t; \boldsymbol{\mu})$ is considered, followed next by the hyperreduction of nonlinear, solution- and $\boldsymbol{\mu}$-dependent vector quantities such as the internal force vector $\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$. In each case, the computation of the approximate function $\hat{\mathbf{g}}_{N_h}(t; \boldsymbol{\mu})$ (and $\hat{\mathbf{f}}_{N_h}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$) is discussed, and then the efficient computation of the final result $\widetilde{\mathbf{g}}_n(t; \boldsymbol{\mu})$ (and $\widetilde{\mathbf{f}}_n(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$) is described.

5.4.2.1.1 Empirical interpolation method for μ-dependent functions

The EIM was introduced in [9, 26] for the approximation at the continuum level of a family of parameter-dependent functions $\mathcal{G} = \{g(\cdot; \mu), \mu \in \mathcal{P}\} \subset C^0(\overline{\Omega})$, where in three dimensions, $\Omega \subset \mathbb{R}^3$. Here and throughout this section, the time dependence of a function $g \in \mathcal{G}$ is accounted for by enlarging the dimension of $\mathcal{P}$ by 1 and treating time as a component of the parameter vector μ. The method is built on basis functions obtained by sampling g at a suitably selected set of points in $\mathcal{P}$, rather than on predefined basis functions as in polynomial interpolation. Its purpose is to find approximations to elements of $\mathcal{G}$ through an operator $\mathcal{I}_m^{\mathbf{x}}$ that interpolates the function $g(\cdot; \mu)$ at some carefully selected points in Ω. Given a set of m basis functions $\{\rho_1, \ldots, \rho_m\}$ that are linear combinations of m particular snapshots $g(\cdot; \mu_{\text{EIM}}^1), \ldots, g(\cdot; \mu_{\text{EIM}}^m)$, and a set of m interpolation points $T_m = \{\mathbf{t}^1, \ldots, \mathbf{t}^m\} \subset \overline{\Omega}$ to be properly selected – also referred to as *magic points* – the interpolant $\mathcal{I}_m^{\mathbf{x}} g(\cdot; \mu)$ of $g(\cdot; \mu)$, for $\mu \in \mathcal{P}$, admits a separable expansion and therefore can be written as follows:

$$\mathcal{I}_m^{\mathbf{x}} g(\mathbf{x}; \mu) = \sum_{j=1}^{m} \gamma_j(\mu) \rho_j(\mathbf{x}), \quad \mathbf{x} \in \Omega. \tag{5.28}$$

The μ-dependent coefficients $\gamma_j(\mu)$ are obtained by fulfilling the interpolation constraints

$$\mathcal{I}_m^{\mathbf{x}} g(\mathbf{t}^i; \mu) = g(\mathbf{t}^i; \mu), \quad i = 1, \ldots, m, \tag{5.29}$$

which yields the linear system of equations

$$\sum_{j=1}^{m} \gamma_j(\mu) \rho_j(\mathbf{t}^i) = g(\mathbf{t}^i; \mu), \quad i = 1, \ldots, m. \tag{5.30}$$

The matrix form of this system reads

$$\mathbb{B}_m \boldsymbol{\gamma}(\mu) = \mathbf{g}_m(\mu), \quad \forall \mu \in \mathcal{P},$$

where $[\mathbb{B}_m]_{ij} = \rho_j(\mathbf{t}^i)$, $[\boldsymbol{\gamma}(\mu)]_j = \gamma_j(\mu)$, and $[\mathbf{g}_m(\mu)]_i = g(\mathbf{t}^i; \mu)$, for $i, j = 1, \ldots, m$.

The construction of the basis functions $\rho_i(\mathbf{x})$, $i = 1, \ldots, m$, yielding the approximation space $X_m = \text{span}\{\rho_1, \ldots, \rho_m\}$, is governed by a greedy procedure [29] that additionally yields the interpolation points $T_m = \{\mathbf{t}^1, \ldots, \mathbf{t}^m\}$ and a sample set of parameters $S_m = \{\mu_{\text{EIM}}^1, \ldots, \mu_{\text{EIM}}^m\}$, all of which are needed for generating the basis functions. The greedy procedure is a two-step algorithm that can be described as follows:

– *Initialization step*. The first sample point is chosen as

$$\mu_{\text{EIM}}^1 = \arg \max_{\mu \in \mathcal{P}} \|g(\cdot; \mu)\|_{L^\infty(\Omega)}$$

and therefore $S_1 = \{\mu_{\text{EIM}}^1\}$. The first generating function is defined as

$$\xi_1(\mathbf{x}) = g(\mathbf{x}; \mu_{\text{EIM}}^1).$$

The first interpolation point is selected as

$$\mathbf{t}^1 = \arg\max_{\mathbf{x} \in \overline{\Omega}} |\xi_1(\mathbf{x})|$$

and therefore $T_1 = \{\mathbf{t}^1\}$. Hence, the first basis function is constructed as

$$\rho_1(\mathbf{x}) = \xi_1(\mathbf{x})/\xi_1(\mathbf{t}^1)$$

and therefore $X_1 = \text{span}\{\rho_1\}$. Finally, the initial interpolation matrix is defined as

$$[\mathbb{B}_m]_{11} = \rho_1(\mathbf{t}^1) = 1.$$

At this stage, the available information allows the definition of the interpolant as the only function that is collinear with ρ_1 and coincides with g at $\mathbf{t}^1$ – that is, $\mathcal{I}_1^{\mathbf{x}} g(\mathbf{x}; \boldsymbol{\mu}) = g(\mathbf{t}^1; \boldsymbol{\mu})\rho_1(\mathbf{x})$.

– *Recursive step*. At each j-th step, $j = 1, \ldots, m-1$, given the nested set of interpolation points $T_j = \{\mathbf{t}^1, \ldots, \mathbf{t}^j\}$ and the set of basis functions $\{\rho_1, \ldots, \rho_j\}$, the $(j+1)$-th generating function is selected as the snapshot that is worst approximated by the current interpolant – that is, the snapshot that maximizes the difference between g and $\mathcal{I}_j^{\mathbf{x}} g$. This can be written as

$$\boldsymbol{\mu}_{\text{EIM}}^{j+1} = \arg\max_{\boldsymbol{\mu} \in \mathcal{P}} \big\| g(\cdot; \boldsymbol{\mu}) - \mathcal{I}_j^{\mathbf{x}} g(\cdot; \boldsymbol{\mu}) \big\|_{L^\infty(\Omega)}, \tag{5.31}$$

which yields the generating function

$$\xi_{j+1}(\mathbf{x}) = g(\mathbf{x}; \boldsymbol{\mu}_{\text{EIM}}^{j+1})$$

and the set $S_{j+1} = S_j \cup \{\boldsymbol{\mu}_{\text{EIM}}^{j+1}\}$. The $(j+1)$-th interpolation point is selected by first solving the linear system

$$\sum_{l=1}^{j} \rho_l(\mathbf{t}^i)\gamma_l = \xi_{j+1}(\mathbf{t}^i), \quad i = 1, \ldots, j,$$

in order to characterize the interpolant $\mathcal{I}_j^{\mathbf{x}} \xi_{j+1}$ (5.28), then evaluating the residual

$$r_{j+1}(\mathbf{x}) = \xi_{j+1}(\mathbf{x}) - \mathcal{I}_j^{\mathbf{x}} \xi_{j+1}(\mathbf{x}), \tag{5.32}$$

and finally choosing the point in Ω where ξ_{j+1} is worst approximated – that is,

$$\mathbf{t}^{j+1} = \arg\max_{\mathbf{x} \in \overline{\Omega}} |r_{j+1}(\mathbf{x})|. \tag{5.33}$$

This point selection is followed by the update $T_{j+1} = T_j \cup \{\mathbf{t}^{j+1}\}$. Then, the new basis function is constructed as

$$\rho_{j+1}(\mathbf{x}) = \frac{\xi_{j+1}(\mathbf{x}) - \mathcal{I}_j^{\mathbf{x}} \xi_{j+1}(\mathbf{x})}{\xi_{j+1}(\mathbf{t}^{j+1}) - \mathcal{I}_j^{\mathbf{x}} \xi_{j+1}(\mathbf{t}^{j+1})} = \frac{r_{j+1}(\mathbf{x})}{r_{j+1}(\mathbf{t}^{j+1})}$$

and therefore $X_{j+1} = \text{span}\{\rho_i\}$, $i = 1, \ldots, j+1$.

The entire procedure described above is repeated until a given tolerance $\varepsilon_{\mathrm{EIM}}$ is reached for the maximum norm of the residual (5.32), or until a given maximum number $m_{\max}$ of terms is calculated.

EIM yields a sequence of hierarchical spaces $X_1 \subset X_2 \subset \cdots \subset X_m$ and a set $\{\rho_1, \ldots, \rho_m\}$ of linearly independent basis functions. Moreover, the m interpolation points where the approximation is required to match the function being interpolated are iteratively determined in an adaptive fashion – that is, without having to recompute all previously selected points.

An a priori error estimation, bounded in terms of the best approximation error, holds for EIM. Precisely, it is first noted that for any given m

$$\mathcal{I}_m^{\mathbf{x}} g(\mathbf{x}; \boldsymbol{\mu}) = \sum_{i=1}^{m} g(\mathbf{t}^i; \boldsymbol{\mu}) l_i^m(\mathbf{x}), \quad \text{where } l_i^m(\mathbf{x}) = \sum_{j=1}^{m} \rho_j(\mathbf{x}) (\mathbb{B}_m^{-1})_{ji},$$

and by definition, $l_i^m(\mathbf{x}^j) = \delta_{ij}$, $i, j = 1, \ldots, m$. Then for any $g \in \mathcal{G}$, the interpolation error satisfies

$$\left\| g(\cdot; \boldsymbol{\mu}) - \mathcal{I}_m^{\mathbf{x}} g(\cdot; \boldsymbol{\mu}) \right\|_{L^\infty(\Omega)} \leq (1 + \Lambda_m) \inf_{g_m \in X_m} \left\| g(\cdot; \boldsymbol{\mu}) - g_m \right\|_{L^\infty(\Omega)},$$

where

$$\Lambda_m = \sup_{\mathbf{x} \in \Omega} \sum_{i=1}^{m} |l_i^m(\mathbf{x})|$$

is the Lebesgue constant (see, e. g., [29]) and $\{l_i^m \in X_m\}$ denotes a set of characteristic Lagrangian functions ($l_i^m(\mathbf{x}^j) = \delta_{ij}$, $i, j = 1, \ldots, m$). Here Λ_m depends on X_m and on the magic points T_m, but is $\boldsymbol{\mu}$-independent. A pessimistic upper bound for the Lebesgue constant is $\Lambda_m \leq 2^m - 1$. A posteriori estimates for the interpolation error can be found in [9, 19], and a link between the convergence rate of the EIM approximation and the Kolmogorov n-width of the manifold $\mathcal{G}$ is discussed in [29].

In practice, finding the supremum in (5.31) and (5.33) is not computationally feasible unless approximations of both Ω and $\mathcal{P}$ are considered. Two noteworthy approximations are:

- A fine sample $\Xi_{\mathrm{train}}^{\mathrm{EIM}} \subset \mathcal{P}$ of cardinality $|\Xi_{\mathrm{train}}^{\mathrm{EIM}}| = n_{\mathrm{train}}$ to train EIM.
- A discrete approximation $\Omega_h = \{\mathbf{x}^k\}_{k=1}^{n_k}$ of Ω of dimension n_k. For example in an FE context, the points $\mathbf{x}^k$ can be the vertices of the computational mesh, or the quadrature points of its elements. This approximation leads to N_h DOFs, where $N_h = N_h(n_k)$ is the dimension of the FOM (5.1) and typically $N_h \geq n_k$.

In this setting, a computable, algebraic version of EIM can be provided after the following quantities are introduced:

- A vector representation $\mathbf{g}_{N_h}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ of $g(\cdot; \boldsymbol{\mu})$, with entries defined by

$$[\mathbf{g}_{N_h}(\boldsymbol{\mu})]_i = g(\mathbf{x}^{\mathrm{map}(i)}; \boldsymbol{\mu}), \quad i = 1, \ldots, N_h,$$

  where map is the function that maps each DOF of the FOM (5.1) to the vertex of Ω_h to which it is attached. This representation is obtained by evaluating the function g in Ω_h, for any $\boldsymbol{\mu} \in \mathcal{P}$.
- The matrix $\mathbb{U} \in \mathbb{R}^{N_h \times m}$ defined as

$$\mathbb{U} = [\, \boldsymbol{\rho}_1 \,|\, \ldots \,|\, \boldsymbol{\rho}_m \,] \tag{5.34}$$

  whose columns encode the discrete representation of the basis functions $\{\rho_1, \ldots, \rho_m\}$, i. e., $[\mathbb{U}]_{ij} = \rho_j(\mathbf{x}^{\mathrm{map}(i)})$.
- A set of m interpolation indices $\mathcal{I} = \{i_1, \ldots, i_m\}$ associated with $n_m \le m$ interpolation points $\{\mathbf{t}^1, \ldots, \mathbf{t}^{n_m}\}$ such that $\{\mathbf{t}^1, \ldots, \mathbf{t}^{n_m}\} = \{\mathbf{x}^{j_1}, \ldots, \mathbf{x}^{j_{n_m}}\}$, where for $k = 1, \ldots, n_m$, j_k is related to an interpolation index $i_l \in \mathcal{I}$ via $j_k = \mathrm{map}(i_l)$.

From (5.28), (5.30), and (5.34), it follows that the discrete representation of the interpolation operator $\mathcal{I}_m^{\mathbf{x}}$, denoted by $\hat{\mathbf{g}}_{N_h}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$, is given by

$$\hat{\mathbf{g}}_{N_h}(\boldsymbol{\mu}) = \mathbb{U}\mathbf{g}_m(\boldsymbol{\mu}), \tag{5.35}$$

where $\mathbf{g}_m(\boldsymbol{\mu}) \in \mathbb{R}^m$ is the solution of the linear system

$$[\hat{\mathbf{g}}_{N_h}(\boldsymbol{\mu})]_{i_l} = \sum_{j=1}^{m} [\mathbf{g}_m(\boldsymbol{\mu})]_j [\boldsymbol{\rho}_j]_{i_l} = [\mathbf{g}_{N_h}(\boldsymbol{\mu})]_{i_l}, \quad l = 1, \ldots, m. \tag{5.36}$$

Denoting by $\mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu}) \in \mathbb{R}^m$ the vector whose components are $[\mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu})]_l = [\mathbf{g}_{N_h}(\boldsymbol{\mu})]_{i_l}$ for $l = 1, \ldots, m$ and noting that the $m \times m$ matrix $\mathbb{B}_m$ is easily formed by restricting the $N_h \times m$ matrix $\mathbb{U}$ to the rows indexed in $\mathcal{I}$, i. e., $\mathbb{B}_m = \mathbb{U}^{\mathcal{I}}$, (5.29) can be written in compact form as

$$\mathbb{U}^{\mathcal{I}}\mathbf{g}_m(\boldsymbol{\mu}) = \mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu}). \tag{5.37}$$

From (5.35) and (5.37), it follows that

$$\hat{\mathbf{g}}_{N_h}(\boldsymbol{\mu}) = \mathbb{U}(\mathbb{U}^{\mathcal{I}})^{-1}\mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{P}, \tag{5.38}$$

which completes the description of the algebraic version of the EIM. A corresponding algorithm is given in Algorithm 5.2. Note that the solution of the dense linear system (5.37) requires $\mathcal{O}(m^2)$ operations, thanks to the lower triangular structure of $\mathbb{U}^{\mathcal{I}}$. Note also that at each iteration, Algorithm 5.2 involves the evaluation of $\mathbf{g}_{N_h}(\boldsymbol{\mu})$ for $\boldsymbol{\mu} \in \Xi_{\mathrm{train}}^{\mathrm{EIM}}$. Should this operation be expensive, one may form and store once for all the (possibly dense) matrix

$$\mathbb{S} = [\, \mathbf{g}_{N_h}(\boldsymbol{\mu}^1) \,|\, \ldots \,|\, \mathbf{g}_{N_h}(\boldsymbol{\mu}^{n_{\mathrm{train}}}) \,] \in \mathbb{R}^{N_h \times n_{\mathrm{train}}}$$

Algorithm 5.2: EIM (computable version): offline and online phases.

Offline phase

Input: $\Xi_{\text{train}}^{\text{EIM}}$, Ω_h, $m_{\max}$, ε_{EIM}

Output: $\mathbb{U}$, $\mathcal{I}$

1: $\boldsymbol{\mu}^1 = \arg\max\limits_{\boldsymbol{\mu}\in\Xi_{\text{train}}^{\text{EIM}}} \|\mathbf{g}_{N_h}(\boldsymbol{\mu})\|_\infty$

2: $\mathbf{r}_1 = \mathbf{g}_{N_h}(\boldsymbol{\mu}^1)$

3: $i_1 = \arg\max\limits_{i=1,\dots,N_h} |[\mathbf{g}_{N_h}(\boldsymbol{\mu}^1)]_i|$

4: $\boldsymbol{\rho}_1 = \mathbf{g}_{N_h}(\boldsymbol{\mu}^1)/[\mathbf{g}_{N_h}(\boldsymbol{\mu}^1)]_{i_1}$

5: $\mathbb{U} \leftarrow [\boldsymbol{\rho}_1]$, $\mathcal{I} \leftarrow \{i_1\}$, $T \leftarrow \{\mathbf{x}^{j_1}\}$

6: **for** $k = 2$ **to** $m_{\max}$ **do**

7: $\quad\boldsymbol{\mu}^k = \arg\max\limits_{\boldsymbol{\mu}\in\Xi_{\text{train}}^{\text{EIM}}} \|\mathbf{g}_{N_h}(\boldsymbol{\mu}) - \mathbb{U}(\mathbb{U}^{\mathcal{I}})^{-1}\mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu})\|_\infty$

8: $\quad$**if** $\|\mathbf{g}_{N_h}(\boldsymbol{\mu}^k) - \mathbb{U}(\mathbb{U}^{\mathcal{I}})^{-1}\mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu}^k)\|_\infty < \varepsilon_{\text{EIM}}$ **then**

9: $\quad\quad$**break**

10: $\quad$**end if**

11: $\quad\mathbf{r}_k = \mathbf{g}_{N_h}(\boldsymbol{\mu}^k) - \mathbb{U}(\mathbb{U}^{\mathcal{I}})^{-1}\mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu}^k)$

12: $\quad i_k = \arg\max\limits_{i=1,\dots,N_h} |[\mathbf{r}_k]_i|$

13: $\quad\boldsymbol{\rho}_k = \mathbf{r}_k/[\mathbf{r}_k]_{i_k}$

14: $\quad\mathbb{U} \leftarrow [\mathbb{U}\ \boldsymbol{\rho}_k]$, $\mathcal{I} \leftarrow \mathcal{I} \cup \{i_k\}$, $T \leftarrow T \cup \{\mathbf{x}^{j_k}\}$

15: **end for**

Online phase

Input: $\boldsymbol{\mu}^\star$, T_{n_m}, $\mathbb{U}^{\mathcal{I}}$

Output: $\mathbf{g}_m(\boldsymbol{\mu}^\star)$

1: Form $\mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu}^\star)$ by evaluating $g(\cdot;\boldsymbol{\mu}^\star)$ at the interpolation points $T_{n_m} = \{\mathbf{t}^1,\dots,\mathbf{t}^{n_m}\}$

2: Solve $\mathbb{U}^{\mathcal{I}}\mathbf{g}_m(\boldsymbol{\mu}^\star) = \mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu}^\star)$

---

before entering the while loop. However, already for moderately large values of N_h and n_{train}, storing the matrix $\mathbb{S}$ can be quite challenging.

After the approximation (5.38) has been computed, the final computation of the hyperreduced quantity $\widetilde{\mathbf{g}}_n(\boldsymbol{\mu}) = \mathbb{W}^T\hat{\mathbf{g}}_{N_h}(\boldsymbol{\mu})$ (or $\widetilde{\mathbf{g}}_n(\boldsymbol{\mu}) = \mathbb{V}^T\hat{\mathbf{g}}_{N_h}(\boldsymbol{\mu})$ in the case of a Galerkin projection) is performed as follows. In the offline phase, the matrix $\mathbb{Q} = \mathbb{W}^T\mathbb{U}(\mathbb{U}^{\mathcal{I}})^{-1} \in \mathbb{R}^{n\times m}$ is precomputed. Then, for each queried parameter point $\boldsymbol{\mu}^\star$ encountered in the online phase, $\widetilde{\mathbf{g}}_n(\boldsymbol{\mu}^\star) = \mathbb{Q}\mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu}^\star)$ is computed using a matrix-vector product whose computational complexity is only $\mathcal{O}(nm)$.

Remark 5.4. Note that if, for whatever reason, a set of linearly independent basis functions $\{\rho_1,\dots,\rho_m\}$ were given, then the procedure of finding the interpolation points (or magic points) is well-defined. This remark underlies the idea exploited by DEIM, as al-

ready highlighted in [29] in a general setting. While the computable, algebraic version of EIM given in Algorithm 5.2 was never published before – and certainly not before the introduction of DEIM in [15] – it can be reasonably assumed that this algebraic version or some variant of it (for example, see [10]) was used to compute the numerical results reported in [26] and in subsequent papers [11, 25].

5.4.2.1.2 Discrete empirical interpolation method for μ-dependent functions

Introduced in [15], the DEIM can be described as a variant of Algorithm 5.2 in which the construction of the basis $\mathbb{U}$ is not necessarily embedded in the greedy procedure used for guiding the remainder of the computation of the approximation $\hat{\mathbf{g}}_{N_h}(\boldsymbol{\mu})$. Specifically, DEIM approximates a function $\mathbf{g}_{N_h}(\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ similarly to EIM, by projection onto a low-dimensional subspace spanned by a basis $\mathbb{U}$. This can be written as

$$\mathbf{g}_{N_h}(\boldsymbol{\mu}) \approx \hat{\mathbf{g}}_{N_h}(\boldsymbol{\mu}) = \mathbb{U}\mathbf{g}_m(\boldsymbol{\mu}),$$

where $\mathbb{U} = [\boldsymbol{\rho}_1 \,|\, \ldots \,|\, \boldsymbol{\rho}_m] \in \mathbb{R}^{N_h \times m}$, $\mathbf{g}_m(\boldsymbol{\mu}) \in \mathbb{R}^m$ is the corresponding vector of generalized coordinates, and $m \ll N_h$. DEIM also selects the interpolation points iteratively, using the same greedy procedure as EIM. However, DEIM was introduced in [15] by proposing to construct $\mathbb{U}$ through the application of POD to a set of computed snapshots

$$\mathbb{S} = [\,\mathbf{g}_{N_h}(\boldsymbol{\mu}_{\text{DEIM}}^1) \,|\, \ldots \,|\, \mathbf{g}_{N_h}(\boldsymbol{\mu}_{\text{DEIM}}^{n_s})\,], \quad n_s \geq m.$$

Hence, the key descriptors of this method can be summarized as follows:

1. Construction of a set of snapshots obtained by sampling $\mathbf{g}_{N_h}(\boldsymbol{\mu})$ at values $\boldsymbol{\mu}_{\text{DEIM}}^i$, $i = 1, \ldots, n_s$, and application of POD to extract the basis

$$\mathbb{U} = [\boldsymbol{\rho}_1 \,|\, \ldots \,|\, \boldsymbol{\rho}_m] = \text{POD}([\,\mathbf{g}_{N_h}(\boldsymbol{\mu}_{\text{DEIM}}^1) \,|\, \ldots \,|\, \mathbf{g}_{N_h}(\boldsymbol{\mu}_{\text{DEIM}}^{n_s})\,], \varepsilon_{\text{POD}}), \tag{5.39}$$

   where ε_{POD} is the usual prescribed tolerance for basis truncation.

2. Iterative selection of the set of m indices $\mathcal{I} \subset \{1, \ldots, N_h\}$, where $|\mathcal{I}|$ is therefore equal to the dimension m of the basis $\mathbb{U}$, using a greedy procedure and this basis. This operation, which minimizes at each step the interpolation error measured in the maximum norm over the set of aforementioned snapshots, is the same as the selection of the magic points in EIM.

3. Computation of the vector of generalized coordinates $\mathbf{g}_m(\boldsymbol{\mu})$, given a new $\boldsymbol{\mu}$. This computation is performed by imposing interpolation constraints at the m points of $\mathcal{I}$ corresponding to the selected indices of $\mathbf{g}_{N_h}(\boldsymbol{\mu})$. It requires the solution of the following linear system:

$$\mathbb{U}^{\mathcal{I}}\mathbf{g}_m(\boldsymbol{\mu}) = \mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu}), \tag{5.40}$$

   where, as previously, $\mathbb{U}^{\mathcal{I}} \in \mathbb{R}^{m \times m}$ and $\mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu}) \in \mathbb{R}^m$ are the matrix and vector formed by the rows of $\mathbb{U}$ and $\mathbf{g}_{N_h}(\boldsymbol{\mu})$ indexed in $\mathcal{I}$, respectively. As a result,

$$\hat{\mathbf{g}}_{N_h}(\boldsymbol{\mu}) = \mathbb{U}(\mathbb{U}^{\mathcal{I}})^{-1}\mathbf{g}_{N_h}^{\mathcal{I}}. \tag{5.41}$$

Algorithm 5.3: DEIM: offline and online phases.

Offline phase

Input: $\mathbb{S}, \varepsilon_{\text{POD}}$

Output: $\mathbb{U}, \mathcal{I}$

1: $[\boldsymbol{\rho}_1 \mid \ldots \mid \boldsymbol{\rho}_m] = \text{POD}(\mathbb{S}, \varepsilon_{\text{POD}})$

2: $i_1 = \arg\max\limits_{i=1,\ldots,N_h} |[\boldsymbol{\rho}_1]_i|$

3: $\mathbb{U} \leftarrow [\boldsymbol{\rho}_1], \mathcal{I} \leftarrow \{i_1\}$

4: **for** $k = 2$ **to** m **do**

5:     $\mathbf{r}_k = \boldsymbol{\rho}_k - \mathbb{U}(\mathbb{U}^{\mathcal{I}})^{-1}\boldsymbol{\rho}_k^{\mathcal{I}}$

6:     $i_k = \arg\max\limits_{i=1,\ldots,N_h} |[\mathbf{r}_k]_i|$

7:     $\mathbb{U} \leftarrow [\mathbb{U} \ \boldsymbol{\rho}_k], \mathcal{I} \leftarrow \mathcal{I} \cup \{i_k\}$

8: **end for**

Online phase

Input: $\boldsymbol{\mu}^\star, \mathcal{I}, \mathbb{U}^{\mathcal{I}}$

Output: $\mathbf{g}_m(\boldsymbol{\mu}^\star)$

1: Form $\mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu}^\star)$ by evaluating $\mathbf{g}_{N_h}(\boldsymbol{\mu}^\star)$ at the interpolation indices $\mathcal{I}$

2: Solve $\mathbb{U}^{\mathcal{I}}\mathbf{g}_m(\boldsymbol{\mu}^\star) = \mathbf{g}_{N_h}^{\mathcal{I}}(\boldsymbol{\mu}^\star)$

The construction of the basis $\mathbb{U}$ and the selection of the set of indices $\mathcal{I}$ are summarized in Algorithm 5.3. The well-posedness of the general, dense, linear system (5.40) is amply discussed in [15] and related papers. The computational complexity of its solution is $O(m^3)$.

The approximation error $\mathbf{g}_{N_h}(\boldsymbol{\mu}) - \hat{\mathbf{g}}_{N_h}(\boldsymbol{\mu})$ can be bounded as

$$\|\mathbf{g}_{N_h}(\boldsymbol{\mu}) - \hat{\mathbf{g}}_{N_h}(\boldsymbol{\mu})\|_2 \leq \|(\mathbb{U}^{\mathcal{I}})^{-1}\|_2 \|(\mathbb{I} - \mathbb{U}\mathbb{U}^T)\mathbf{g}_{N_h}(\boldsymbol{\mu})\|_2, \tag{5.42}$$

where

$$\|(\mathbb{I} - \mathbb{U}\mathbb{U}^T)\mathbf{g}_{N_h}(\boldsymbol{\mu})\|_2 \approx \sigma_{m+1} \tag{5.43}$$

and σ_{m+1} is the first discarded singular value of the matrix $\mathbb{S}$ when applying the POD procedure to construct $\mathbb{U} \in \mathbb{R}^{N_h \times m}$. The result (5.43) holds for any $\boldsymbol{\mu} \in \mathcal{P}$, provided that a suitable sampling of the parameter space has been carried out to build the snapshot matrix $\mathbb{S}$. In that case, the predictive projection error (5.43) is comparable to the training projection error σ_{m+1}. The estimate (5.42) is built on the information related to the first discarded term: It can be seen as a heuristic measure of the DEIM error. Further results about a posteriori error estimation for POD-DEIM reduced nonlinear dynamical systems can be found in [41].

After the approximation (5.41) has been computed, the final computation of the hyperreduced quantity $\bar{\mathbf{g}}_n(\boldsymbol{\mu}) = \mathbb{W}^T\hat{\mathbf{g}}_{N_h}(\boldsymbol{\mu})$ (or $\bar{\mathbf{g}}_n(\boldsymbol{\mu}) = \mathbb{V}^T\hat{\mathbf{g}}_{N_h}(\boldsymbol{\mu})$ in the case of a

Galerkin projection) is performed as previously described for the case of the EIM method, in $\mathcal{O}(nm)$ computations only.

5.4.2.1.3 EIM and DEIM for solution- and $\boldsymbol{\mu}$-dependent functions

The computable, algebraic version of either the EIM or the DEIM can be equally used to speed up the evaluation of the parametric, reduced, nonlinear, internal force vector $\mathbf{f}_n(\mathbf{u}_n(t;\boldsymbol{\mu}))$ defined in (5.8). Because of space limitations, however, only the case of DEIM and the context of the Galerkin projection are considered here. As far as hyperreduction is concerned, similar conclusions can be drawn for the EIM and the more general context of the Petrov–Galerkin projection.

The problem is to find m $(t,\boldsymbol{\mu})$-independent basis functions $\boldsymbol{\rho}_j \in \mathbb{R}^{N_h}$, $j = 1,\ldots,m$, and a $(t,\boldsymbol{\mu})$-dependent vector $\mathbf{f}_m(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) \in \mathbb{R}^m$, such that the nonlinear, internal force vector can be approximated as

$$\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) \approx \hat{\mathbf{f}}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \mathbb{U}_f \mathbf{f}_m(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}),$$

where $\mathbb{U}_f = [\boldsymbol{\rho}_1 \mid \ldots \mid \boldsymbol{\rho}_m] \in \mathbb{R}^{N_h \times m}$. This problem can be solved as follows.

In an offline stage, DEIM is applied to a set of snapshots

$$\mathbf{S}_f = \{\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t^k;\boldsymbol{\mu}^s);\boldsymbol{\mu}^s),\ s = 1,\ldots,n_s,\ k = 0,\ldots,N_t - 1\} \tag{5.44}$$

in order to build the basis $\mathbb{U}_f$ and a set of m interpolation indices $\mathcal{I}_f = \{i_l\}_{l=1}^m$, where $|\mathcal{I}_f| = m$ is the number of empirical basis functions. For any new $\boldsymbol{\mu} \in \mathcal{P}$, $\mathbf{f}_m(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) \in \mathbb{R}^m$ is then obtained during an online stage by solving, similarly to (5.36), the following linear system of equations:

$$\left[\hat{\mathbf{f}}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})\right]_{i_l} = \sum_{j=1}^m \left[\mathbf{f}_m(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})\right]_j [\boldsymbol{\rho}_j]_{i_l} = \left[\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})\right]_{i_l}$$

for $l = 1,\ldots,m$ – that is,

$$\hat{\mathbf{f}}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \mathbb{U}_f (\mathbb{U}_f^{\mathcal{I}})^{-1} \mathbf{f}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) \quad \forall \boldsymbol{\mu} \in \mathcal{P}.$$

As before, $\mathbb{U}_f^{\mathcal{I}}$ and $\mathbf{f}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$ denote here the matrix and vector formed by the $\mathcal{I}$ rows of $\mathbb{U}_f$ and $\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$, respectively. Then, the parametric, reduced, nonlinear, internal force vector $\mathbf{f}_n(\mathbf{u}_n(t;\boldsymbol{\mu})) = \mathbb{V}^T \mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$ can be approximated as follows:

$$\begin{aligned}
\mathbf{f}_n(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) \approx \bar{\mathbf{f}}_n(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) &= \mathbb{V}^T \hat{\mathbf{f}}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) \\
&= \underbrace{\mathbb{V}^T \mathbb{U}_f (\mathbb{U}_f^{\mathcal{I}})^{-1}}_{n \times m} \underbrace{\mathbf{f}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})}_{m \times 1}.
\end{aligned}$$

Note that $\mathbf{f}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$ can be efficiently evaluated by employing on the reduced mesh associated with the selected interpolation indices the same assembly routine used in the context of the full-order problem: This is further described in Section 5.4.2.1.5.

A key fact is that the snapshots of the internal force vector highlighted in (5.44) depend on the PROM-based approximation of the state variable. This implies that in principle, a nonhyperreduced PROM must be first built to provide the snapshots $\mathbb{V}\mathbf{u}_n(t^k;\boldsymbol{\mu}^s)$, $s = 1,\ldots,n_s$, $k = 0,\ldots,N_t - 1$, which is computationally inefficient. Depending on the application however, this issue can be avoided and the following snapshots of the internal force vector can be precomputed instead:

$$\bar{\mathbf{S}}_f = \{\mathbf{f}_{N_h}(\mathbf{u}_{N_h}(t^k;\boldsymbol{\mu}^s);\boldsymbol{\mu}^s), \ s = 1,\ldots,n_s, \ k = 0,\ldots,N_t - 1\}. \tag{5.45}$$

Such snapshots can be evaluated while processing the FOM and computing solution snapshots of the state variable in order to construct the PROM. If there exists a $K > 0$ independent of N_h such that

$$\left\|\mathbf{f}_{N_h}(\mathbf{u}_{N_h}(t;\boldsymbol{\mu});\boldsymbol{\mu}) - \mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})\right\| \le K\left\|\mathbf{u}_{N_h}(t;\boldsymbol{\mu}) - \mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu})\right\|$$

for every t and $\boldsymbol{\mu}$, then the error due to using the snapshots (5.45) instead of their counterparts (5.44) for performing the hyperreduction of the internal force vector can be kept under control.

The DEIM can also be used to compute an approximation of the reduced Jacobian matrix $\mathbb{K}_n(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$ defined in (5.9). Indeed,

$$\begin{aligned}
\mathbb{K}_n(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) &\approx \widetilde{\mathbb{K}}_n(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) \\
&= \frac{\partial \widetilde{\mathbf{f}}_n}{\partial \mathbf{u}_n}(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) \\
&= \mathbb{V}^T \widehat{\mathbb{K}}_{N_h} \mathbb{V} \\
&= \underbrace{\mathbb{V}^T \mathbb{U}_f (\mathbb{U}_f^{\mathcal{I}})^{-1}}_{n \times m} \underbrace{\mathbb{K}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})\mathbb{V}^{\mathcal{I}}}_{m \times n},
\end{aligned}$$

where the $\boldsymbol{\mu}$-independent quantity $\mathbb{V}^T \mathbb{U}_f (\mathbb{U}_f^{\mathcal{I}})^{-1}$ can be precomputed, and $\mathbb{K}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})\mathbb{V}^{\mathcal{I}} \in \mathbb{R}^{m \times n}$ can be assembled online.

5.4.2.1.4 Gauss–Newton with approximated tensors method

Whereas the EIM and DEIM were developed in the context of the Galerkin-PMOR, GNAT was introduced in [12] in the Petrov–Galerkin context. Like the DEIM, it was conceived to operate at the fully discrete level. However unlike the DEIM and the computable version of the EIM, it was designed to operate on discrete PROMs of the form

$$\mathbb{W}^T \mathbf{r}_{N_h}(\mathbb{V}\mathbf{u}_n(t^{k+1};\boldsymbol{\mu});\boldsymbol{\mu}) = 0, \tag{5.46}$$

where t^{k+1} denotes the time instance at the $(k+1)$-th computational time step;

$$\mathbf{r}_{N_h}(\mathbb{V}\mathbf{u}_n(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbb{M}_n(\boldsymbol{\mu})\dot{\mathbf{u}}_n(t^{k+1}; \boldsymbol{\mu}) + \mathbf{f}_n(\mathbf{u}_n(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu}) - \mathbf{g}_n(t^{k+1}; \boldsymbol{\mu}) \quad (5.47)$$

is the residual associated with the discretization of the FOM (5.1); and the left ROB $\mathbb{W}$ is chosen such that the solution of the projected nonlinear system of equations (5.46) results in the minimization in the two-norm of the discrete, nonlinear residual $\mathbf{r}_{N_h}(\mathbb{V}\mathbf{u}_n(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})$ over the approximation subspace associated with the right ROB $\mathbb{V}$. Namely, GNAT operates in the context of an iteration-dependent left ROB

$$\mathbb{W} = \frac{\partial \mathbf{r}_{N_h}}{\partial \mathbf{u}_n}(\mathbb{V}\mathbf{u}_n(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu}) = \frac{\partial \mathbf{r}_{N_h}}{\partial \mathbf{u}_{N_h}}(\mathbb{V}\mathbf{u}_n(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})\mathbb{V}, \quad (5.48)$$

where $\mathbb{J}_{N_h}(\mathbb{V}\mathbf{u}_n(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu}) = \frac{\partial \mathbf{r}_{N_h}}{\partial \mathbf{u}_{N_h}}(\mathbb{V}\mathbf{u}_n(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})$ is the Jacobian matrix of the discrete, nonlinear residual $\mathbf{r}_{N_h}(\mathbb{V}\mathbf{u}_n(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})$. In this case, solving the system of equations (5.46) using Newton's method is equivalent to solving the nonlinear, least-squares minimization problem

$$\mathbf{u}_n(t^{k+1}; \boldsymbol{\mu}) = \underset{\mathbf{x}\in\mathbb{R}^n}{\arg\min}\|\mathbf{r}_{N_h}(\mathbb{V}\mathbf{x}; \boldsymbol{\mu})\|_2 \quad (5.49)$$

using the Gauss–Newton method. For this reason, the projection method summarized by (5.46) and (5.48) is commonly referred to in the literature as the least-squares Petrov–Galerkin (LSPG) projection method [12].

For a self-adjoint FOM (5.1) characterized at the semi-discrete level by an SPD Jacobian matrix $\mathbb{J}_{N_h}(\mathbb{V}\mathbf{u}_n(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})$, the Galerkin-PMOR ($\mathbb{W} = \mathbb{V}$) approach can also be shown to minimize the discrete, nonlinear residual (5.47), but in the $\mathbb{J}_{N_h}^{-1}$-norm. However, in the general case where the Jacobian matrix $\mathbb{J}_{N_h}(\mathbb{V}\mathbf{u}_n(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})$ is not SPD – for example, when the FOM (5.1) results from the semi-discretization of the Navier–Stokes equations – the Galerkin-PMOR approach lacks the optimality property of the LSPG approach associated with the aforementioned minimization process.

The solution of the LSPG minimization problem (5.49) by the Gauss–Newton method incurs the solution of a sequence of linear, least-squares problems of the form

$$\underset{\mathbf{x}\in\mathbb{R}^n}{\min}\|\mathbb{J}_{N_h}(\mathbb{V}\mathbf{u}_n^{(p)}; \boldsymbol{\mu})\mathbb{V}\mathbf{x} + \mathbf{r}_{N_h}(\mathbb{V}\mathbf{u}_n^{(p)}; \boldsymbol{\mu})\|_2, \quad (5.50)$$

where p denotes the p-th iteration of the Gauss–Newton method. Thus, even though the dimension of the search subspace associated with the solution of the problem (5.50) is $n \ll N_h$, the computational complexity of any processing of the LSPG-PROM defined by (5.46) and (5.48) remains dependent on the dimension N_h of the original FOM (5.1). To address this issue, GNAT approximates the columns of the high-dimensional matrix-vector product $\mathbb{J}_{N_h}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}); \boldsymbol{\mu})\mathbb{V}$ and the high-dimensional

nonlinear residual $\mathbf{r}_{N_h}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})$ by projection onto the low-dimensional subspaces spanned by two bases to be determined, $\mathbb{U}_{\mathbb{J}} \in \mathbb{R}^{N_h \times m_{\mathbb{J}}}$ and $\mathbb{U}_{\mathbf{r}} \in \mathbb{R}^{N_h \times m_{\mathbf{r}}}$, respectively, where $m_{\mathbb{J}} \ll N_h$ and $m_{\mathbf{r}} \ll N_h$. This can be written as

$$
\begin{aligned}
\mathbb{J}_{N_h}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})\mathbb{V} &\approx (\widetilde{\mathbb{J}_{N_h}\mathbb{V}})(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbb{U}_{\mathbb{J}}\mathbb{J}_{m_{\mathbb{J}}}^{(p)}(\boldsymbol{\mu}), \\
\mathbf{r}_{N_h}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu}) &\approx \tilde{\mathbf{r}}_{N_h}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbb{U}_{\mathbf{r}}\mathbf{r}_{m_{\mathbf{r}}}^{(p)}(\boldsymbol{\mu}),
\end{aligned}
\tag{5.51}
$$

where $\mathbb{J}_{m_{\mathbb{J}}}^{(p)}(\boldsymbol{\mu}) \in \mathbb{R}^{m_{\mathbb{J}} \times m_{\mathbb{J}}}$ and $\mathbf{r}_{m_{\mathbf{r}}}^{(p)}(\boldsymbol{\mu}) \in \mathbb{R}^{m_{\mathbf{r}}}$. Specifically, GNAT constructs each of the bases $\mathbb{U}_{\mathbb{J}}$ and $\mathbb{U}_{\mathbf{r}}$ by computing snapshots of the matrix-vector products incurred by $\mathbb{J}_{N_h}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})\mathbb{V}$ and the nonlinear residual $\mathbf{r}_{N_h}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})$, respectively, and compressing them using the POD method.

As in EIM and DEIM, GNAT computes the reduced coordinates associated with the subspace approximations (5.51) by minimizing the interpolation error at a selected set of interpolation indices $\mathcal{I}$ determined by a greedy procedure. Specifically, GNAT uses for this purpose a gappy POD reconstruction technique [20, 13] where the interpolation error is minimized at a selected set of indices $\mathcal{I}$ in the least-squares sense. This yields

$$
\begin{aligned}
\mathbb{J}_{m_{\mathbb{J}}}^{(p)}(\boldsymbol{\mu}) &= \underset{\mathbf{x} \in \mathbb{R}^{m_{\mathbb{J}} \times m_{\mathbb{J}}}}{\arg\min} \|\mathbb{J}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})\mathbb{V} - \mathbb{U}_{\mathbb{J}}^{\mathcal{I}}\mathbf{x}\|_F \\
&= (\mathbb{U}_{\mathbb{J}}^{\mathcal{I}})^{\dagger}\mathbb{J}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})\mathbb{V}, \\
\mathbf{r}_{m_{\mathbf{r}}}^{(p)}(\boldsymbol{\mu}) &= \underset{\mathbf{x} \in \mathbb{R}^{m_{\mathbf{r}}}}{\arg\min} \|\mathbf{r}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu}) - \mathbb{U}_{\mathbf{r}}^{\mathcal{I}}\mathbf{x}\|_2 \\
&= (\mathbb{U}_{\mathbf{r}}^{\mathcal{I}})^{\dagger}\mathbf{r}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu}),
\end{aligned}
\tag{5.52}
$$

where the superscript $\mathcal{I}$ designates, as before, the restriction of an algebraic quantity to its rows associated with the indices in $\mathcal{I}$, and the superscript $\dagger$ designates the Moore–Penrose pseudo-inverse. Unlike in the EIM and DEIM, where $|\mathcal{I}|$ is chosen to be equal to the dimension m of the function to be approximated for the purpose of hyperreduction, $|\mathcal{I}|$ is governed in GNAT by the more general constraints $|\mathcal{I}| \geq m_{\mathbf{r}}$ and $|\mathcal{I}| \geq m_{\mathbb{J}}$. These constraints suffice to make the interpolation problems well-posed. However, it is noted here that: both EIM and DEIM can be modified, if desired, to use a set of indices $\mathcal{I}$ whose cardinality is larger than m; and the reconstruction in GNAT can be constrained to the case where $|\mathcal{I}| = m_{\mathbf{r}} = m_{\mathbb{J}}$, if this is deemed more practical.

Substituting the approximations defined by (5.51) and (5.52) into the linear, least-squares minimization problem (5.50) to be solved at each p-th Gauss–Newton iteration and exploiting the orthogonality property $\mathbb{U}_{\mathbb{J}}^T\mathbb{U}_{\mathbb{J}} = \mathbb{I}_{m_{\mathbb{J}}}$ transforms this problem into the hyperreduced, linear, least-squares minimization problem

$$
\begin{aligned}
\min_{\mathbf{x} \in \mathbb{R}^n} \big\| &(\mathbb{U}_{\mathbb{J}}^{\mathcal{I}})^{\dagger}\mathbb{J}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})\mathbb{V}\mathbf{x} \\
&+ \mathbb{U}_{\mathbb{J}}^T\mathbb{U}_{\mathbf{r}}(\mathbb{U}_{\mathbf{r}}^{\mathcal{I}})^{\dagger}\mathbf{r}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})\big\|_2,
\end{aligned}
\tag{5.53}
$$

where the matrices $\mathbb{Q}_{\mathbb{J}} = (\mathbb{U}_{\mathbb{J}}^{\mathcal{I}})^{\dagger} \in \mathbb{R}^{m_{\mathbb{J}} \times |\mathcal{I}|}$ and $\mathbb{Q}_{\mathbf{r}} = \mathbb{U}_{\mathbb{J}}^T\mathbb{U}_{\mathbf{r}}(\mathbb{U}_{\mathbf{r}})^{\dagger} \in \mathbb{R}^{m_{\mathbb{J}} \times |\mathcal{I}|}$ can be precomputed offline. The offline-online decomposition achievable here is the direct result

of the gappy POD-based approximation followed by the projection onto the left ROB $\mathbb{W}$ of full-order quantities (in this case, $\mathbb{J}_{N_h}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})\mathbb{V}$ and $\mathbf{r}_{N_h}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})$), in the same fashion as was achieved for EIM and DEIM. Note that given $m_{\mathbb{J}} \geq n$, the problem (5.53) has a unique solution.

The above description of GNAT shows that the computational complexity of this hyperreduction method is independent of the dimension N_h of the FOM (5.1). At each p-th Gauss–Newton iteration, this complexity has two parts. The first part, which is associated with the computation of the matrix-matrix product $\mathbb{Q}_{\mathbb{J}}\mathbb{J}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})\mathbb{V}$ and that of the matrix-vector product $\mathbb{Q}_{\mathbf{r}}\mathbf{r}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})$, is $\mathcal{O}((m_{\mathbb{J}}n + m_{\mathbb{J}})|\mathcal{I}|)$. The second part, which is associated with the solution of the linear, least-squares problem (5.53) of dimension $m_{\mathbb{J}} \times n$, has the computational complexity $\mathcal{O}(m_{\mathbb{J}}n^2)$ when performed by forming and solving directly the normal equations associated with (5.53), or using instead a QR factorization-based approach.

5.4.2.1.5 Mesh sampling

The EIM, the DEIM, and GNAT have in common a procedure for constructing a set of indices $\mathcal{I}$. Collectively, these indices constitute in general a sampling of the DOFs of the FOM (5.1) that defines the vectors $\mathbf{g}_{N_h}^{\mathcal{I}}(t; \boldsymbol{\mu})$ and $\mathbf{f}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$, or the vectors $\mathbf{r}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})$ and Jacobian matrices $\mathbb{J}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})$, and leads to efficient hyperreduction. In most cases, the construction of these sampled vectors and matrices does not require access neither to the complete mesh nor to the full-order solution $\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu})$. Instead, it requires access only to a small subset of the geometrical entities (for example, the elements of an FE mesh) related to the indices $\mathcal{I}$, and to the contributions of these entities to the aforementioned sampled vectors and matrices. For this reason, the concept of a *reduced mesh* was introduced in [13] in order to ease the implementation in practice of hyperreduction methods of the approximate-then-project type. In this concept, the reduced mesh is defined as the subset of the original mesh which contains only those geometrical entities that are essential to the correct computation of the aforementioned sampled vectors and matrices. Hence, depending on the type of the spatial discretization (i. e., FE, cell-centered or vertex-based finite volume [FV], or finite difference [FD]), the reduced mesh may represent a larger masking of the full-order solution vector described by the larger set of indices $\mathcal{I}_+$ ($|\mathcal{I}_+| \geq |\mathcal{I}|$). Specifically, it is fully described by $\mathcal{I}$ and the stencil of the semi-discretization underlying the FOM (5.1). As such, the reduced mesh describable by $\mathcal{I}_+$ enables the reuse of the same computational framework and corresponding software employed to construct the FOM (5.1), not only to efficiently compute the sampled quantities $\mathbf{g}_{N_h}^{\mathcal{I}}(t; \boldsymbol{\mu})$ and $\mathbf{f}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$, or $\mathbf{r}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})$ and $\mathbb{J}_{N_h}^{\mathcal{I}}(\mathbb{V}\mathbf{u}_n^{(p)}(t^{k+1}; \boldsymbol{\mu}); \boldsymbol{\mu})$, but also to hyperreduce the PROM (5.6) and process the resulting hyperreduced PROM.

The concept of a reduced mesh described above is illustrated in Figure 5.2, for a two-dimensional, first-order, vertex-based, FV semi-discretization based on triangular elements. The left part of this figure shows the computational mesh with the dual
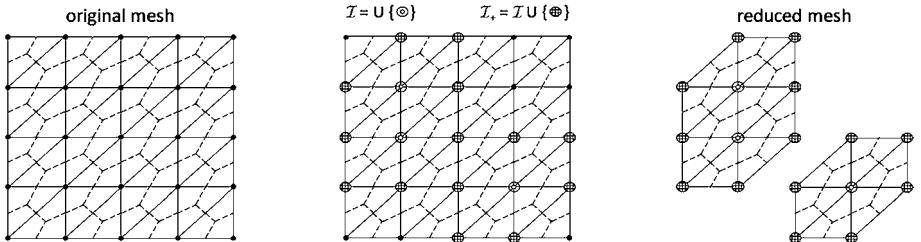
Figure 5.2: Illustration of the reduced mesh concept for a two-dimensional, first-order, vertex-based FV semi-discretization: original mesh (left), basic and enlarged sets of indices (middle), and reduced mesh (right).

cells (or control volumes) delineated using dashed lines. The middle part of this figure highlights a sampled set of indices $\mathcal{I}$ and the associated larger set of indices $\mathcal{I}_+$. The right part of Figure 5.2 shows the reduced mesh associated with $\mathcal{I}_+$. (Note that for an FE semi-discretization using the same mesh, the same set of indices $\mathcal{I}$ leads to the same reduced mesh.)

5.4.2.2 Project-then-approximate hyperreduction methods

Unlike their approximate-then-project counterparts, project-then-approximate hyperreduction methods approximate directly reduced quantities, such as the reduced, nonlinear, internal force vector $\mathbf{f}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbb{W}^T \mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ or its Jacobian $\mathbb{K}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbb{W}^T \mathbb{K}_{N_h}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})\mathbb{V}$ (5.9). Furthermore, unlike the EIM and the DEIM, the method ECSW overviewed below hyperreduces $\boldsymbol{\mu}$-dependent functions such as the reduced, external force vector $\mathbf{g}_n(t; \boldsymbol{\mu})$ and solution- and $\boldsymbol{\mu}$-dependent functions such as $\mathbf{f}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ in the same manner. For this reason, the focus is set here on the general case examplified by the reduced, nonlinear, force balance vector $\mathbf{b}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{f}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) - \mathbf{g}_n(t; \boldsymbol{\mu})$ (5.21) and its Jacobian with respect to $\mathbf{u}_n(t; \boldsymbol{\mu})$, $\mathbb{K}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$.

5.4.2.2.1 Energy-conserving sampling and weighting method
The hyperreduction method ECSW can be derived and/or interpreted using two different but related approaches. Both are combined below to describe this method in the clearest possible manner, in the context of the FE, cell-centered or vertex-based FV, or FD method.

Let $\mathcal{E} = \{e_1, e_2, \ldots, e_{n_e}\}$ denote: the set of n_e elements of a given mesh if the context is set to that of an FE or cell-centered FV semi-discretization over this mesh; the set of dual cells if the context is set to that of a vertex-based FV semi-discretization; or the set of nodes of the mesh if the context is set to that of an FD semi-discretization.

In the case of an FE semi-discretization, the evaluation of $\mathbf{b}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ and the construction of $\mathbb{K}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ are typically performed as described by the following equations:

$$\mathbf{b}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \sum_{e \in \mathcal{E}} \mathbb{W}^T \mathbb{L}_e^T \mathbf{b}_e(\mathbb{L}_e \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}), \tag{5.54}$$

$$\mathbb{K}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \sum_{e \in \mathcal{E}} \mathbb{W}^T \mathbb{L}_e^T \mathbb{K}_e(\mathbb{L}_e \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) \mathbb{L}_e \mathbb{V}, \tag{5.55}$$

where $\mathbb{L}_e$ is the $d_e \times N_h$ Boolean matrix that localizes a high-dimensional matrix defined over the entire mesh to the DOFs associated with the entity (element or dual cell) e; d_e denotes the number of DOFs associated with this entity; $\mathbf{b}_e(\mathbb{L}_e \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{d_e}$ denotes the contribution of this entity to the reduced vector $\mathbf{b}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$; and $\mathbb{K}_e(\mathbb{L}_e \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) \in \mathbb{R}^{d_e \times d_e}$ denotes the contribution of this entity to the reduced matrix $\mathbb{K}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$.

In the case of an FV or FD semi-discretization, the evaluation and construction of the above reduced quantities are performed similarly – that is,

$$\mathbf{b}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \sum_{e \in \mathcal{E}} \mathbb{W}^T \mathbb{L}_e^T \mathbf{b}_e(\mathbb{L}_{e+} \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}), \tag{5.56}$$

$$\mathbb{K}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \sum_{e \in \mathcal{E}} \mathbb{W}^T \mathbb{L}_e^T \mathbb{K}_e(\mathbb{L}_{e+} \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) \mathbb{L}_{e+} \mathbb{V}, \tag{5.57}$$

where $\mathbf{b}_e(\mathbb{L}_{e+} \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ and $\mathbb{K}_e(\mathbb{L}_{e+} \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ denote, as before, the contribution of the entity e to the reduced vector $\mathbf{b}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ and its contribution to the reduced matrix $\mathbb{K}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$, respectively. However, each of these two contributions may depend in this case on the DOFs associated not only with the entity e, but also with neighboring entities as dictated by the semi-discretization stencil. This is designated in (5.56) and (5.57) by the symbol $+$ next to the subscript e of the Boolean matrix $\mathbb{L}_{e+}$ of dimension $(d_e n_{e+}) \times n$, where n_{e+} denotes the number of entities participating in the evaluation of $\mathbf{b}_e(\mathbb{L}_{e+} \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ and $\mathbb{K}_e(\mathbb{L}_{e+} \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$, as required by the stencil of the chosen spatial discretization.

To unify the notation adopted above for an FE, FV, or FD semi-discretization, no distinction is made in the remainder of this section between $\mathbb{L}_e$ and $\mathbb{L}_{e+}$. Instead, it is assumed that if the spatial discretization is of the FV or FD type, $\mathbb{L}_e$ is to be understood as $\mathbb{L}_{e+}$.

Consider the variational setting underlying the construction of the nonlinear, force balance vector $\mathbf{b}_{N_h}(\mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbf{f}_{N_h}(\mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) - \mathbf{g}_{N_h}(t; \boldsymbol{\mu})$ and tangent stiffness matrix $\mathbb{K}_{N_h}(\mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$. In this setting, $\mathbb{W}$ can be interpreted as a matrix of test functions and each entry $[\mathbf{b}_n]_i(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ of the reduced vector $\mathbf{b}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ (5.54) can be interpreted as the virtual work of $\mathbf{b}_{N_h}(\mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ along the virtual "displacement" $[\mathbb{W}^T]_i$.

Now, let $\widetilde{\mathcal{E}} = \{\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_{\tilde{n}_e \ll n_e}\} \subset \mathcal{E}$ represent a reduced mesh obtained by sampling the elements or dual cells of the mesh represented by $\mathcal{E}$, as appropriate. The conservation on this mesh of each virtual work represented by each entry of the reduced

vector $\mathbf{b}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ can be globally expressed as

$$\underbrace{\sum_{e \in \mathcal{E}} \mathbb{W}^T \mathbb{L}_e^T \mathbf{b}_e(\mathbb{L}_e \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})}_{\mathbf{b}_n(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})} = \underbrace{\sum_{e \in \widetilde{\mathcal{E}} \subset \mathcal{E}} \xi_e^{\star} \mathbb{W}^T \mathbb{L}_e^T \mathbf{b}_e(\mathbb{L}_e \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})}_{\widetilde{\mathbf{b}}_n(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})}, \tag{5.58}$$

where the real-valued coefficients $\xi_e^{\star}$ must be introduced so that the above equality may be feasible.

Similarly, each entry $[\mathbb{K}_n]_{ij}(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ of the reduced matrix $\mathbb{K}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ (5.55) can be interpreted as the virtual work of the internal force vector $\mathbb{K}_{N_h}(\mathbb{V}\mathbf{u}_n(t;\boldsymbol{\mu}); \boldsymbol{\mu})[\mathbb{V}^T]_j^T$ along the virtual "displacement" $[\mathbb{W}^T]_i$. The conservation on the reduced mesh represented by $\widetilde{\mathcal{E}}$ of this virtual work for $i = 1, \ldots, n$ and $j = 1, \ldots, n$ can be globally expressed as

$$\underbrace{\sum_{e \in \mathcal{E}} \mathbb{W}^T \mathbb{L}_e^T \mathbb{K}_e(\mathbb{L}_e \mathbb{V} \mathbf{u}_n(t;\boldsymbol{\mu}); \boldsymbol{\mu})\mathbb{L}_e \mathbb{V}}_{\mathbf{K}_n(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})} = \underbrace{\sum_{e \in \widetilde{\mathcal{E}} \subset \mathcal{E}} \xi_e^{\star} \mathbb{W}^T \mathbb{L}_e^T \mathbb{K}_e(\mathbb{L}_e \mathbb{V} \mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})\mathbb{L}_e \mathbb{V}}_{\widetilde{\mathbf{K}}_n(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})}, \tag{5.59}$$

where $\widetilde{\mathcal{E}}$ and its associated set of real-valued coefficients $\{\xi_e^{\star}\}_{e=\widetilde{e}_1}^{\widetilde{e}_{\bar{n}_e}}$ are in principle the same as those that appear in (5.58), because $\mathbb{K}_{N_h}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ is the Jacobian of $(\mathbf{f}_{N_h}(\mathbb{V}\mathbf{u}_n(t)) - \mathbf{g}_{N_h}(t))$. In practice, these coefficients must be determined numerically such that both identities (5.58) and (5.59) hold approximately. For this reason, these identities should be rewritten as

$$\mathbf{b}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) \approx \widetilde{\mathbf{b}}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \sum_{e \in \widetilde{\mathcal{E}} \subset E} \xi_e^{\star} \mathbb{W}^T \mathbb{L}_e^T \mathbf{b}_e(\mathbb{L}_e \mathbb{V} \mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu}), \tag{5.60}$$

$$\mathbf{K}_n(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) \approx \widetilde{\mathbf{K}}_n(\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \sum_{e \in \widetilde{\mathcal{E}} \subset E} \xi_e^{\star} \mathbb{W}^T \mathbb{L}_e^T \mathbb{K}_e(\mathbb{L}_e \mathbb{V} \mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})\mathbb{L}_e \mathbb{V}. \tag{5.61}$$

Expressions (5.60) and (5.61) suggest that each of the approximations $\widetilde{\mathbf{b}}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ and $\widetilde{\mathbf{K}}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ can be interpreted as a *generalized quadrature rule*, where the elements in $\widetilde{\mathcal{E}}$ are the *quadrature points* of this rule and the real coefficients $\{\xi_e^{\star}\}_{e=1}^{\bar{n}_e}$ are its *weights*.

If the high-dimensional matrix $\mathbf{K}_{N_h}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ is SPD and $\mathbb{W} = \mathbb{V}$ (Galerkin projection), or for some other reason the matrix product $\mathbb{W}^T \mathbf{K}_{N_h}(\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})\mathbb{V}$ is SPD, then $\mathbf{K}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ is also SPD. In this case, $\widetilde{\mathbf{K}}_n(\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ should also be SPD, and therefore each coefficient $\xi_e^{\star}$ should be positive – that is, $\xi_e^{\star} \in \mathbb{R}^+$.

5.4.2.2.2 Mesh sampling and weighting

ECSW determines simultaneously the reduced mesh represented by $\widetilde{\mathcal{E}} = \{\widetilde{e}_1, \widetilde{e}_2, \ldots, \widetilde{e}_{\bar{n}_e \ll n_e}\} \subset \mathcal{E}$ and its associated set of element weights $\{\xi_e^{\star}\}_{e=\widetilde{e}_1}^{\widetilde{e}_{\bar{n}_e}}$ by training either approximation (5.60) or (5.61) – for example, the approximation (5.60) – using a set of precomputed reduced, nonlinear, force balance snapshots of the form

$$\mathbf{b}_n(\mathbf{u}_n(t^k; \boldsymbol{\mu}_l); \boldsymbol{\mu}_l) = \mathbb{W}^T \mathbf{b}_{N_h}(\mathbb{V}\mathbf{u}_n(t^k; \boldsymbol{\mu}_l); \boldsymbol{\mu}_l) = \mathbb{W}^T \mathbf{b}_{N_h}^s(\mathbb{V}\mathbf{u}_n^s), \tag{5.62}$$

for $s = 1, \ldots, n_s$, where t^k denotes the k-th sampled time instance and $\boldsymbol{\mu}_l$ the l-th parameter vector sampled in the parameter space $\mathcal{P}$; the superscript s designates a solution snapshot that is precomputed at some time instance $t^k \in [t^0, \ T]$ and for some sampled parameter point $\boldsymbol{\mu}_l \in \mathcal{P}$, and is introduced to simplify the notation; and n_s denotes the total number of precomputed snapshots.

For both convenience and computational efficiency, no snapshot $\mathbf{u}_n^s$ of the solution of the parametric, nonlinear PROM (5.6) is computed in order to evaluate (5.62). Instead, snapshots $\mathbf{u}_{N_h}^s$ of the solution of the parametric, nonlinear FOM (5.1) are first collected as in any standard PMOR method (this approach for achieving computational efficiency is identical to that adopted in the case of DEIM, where the snapshots (5.45) are used instead of their counterparts (5.44)). Next, these high-dimensional solution snapshots are converted *on-the-fly* into the form $\mathbb{V}\mathbf{u}_n^s$ using an orthogonal projection operator onto the subspace spanned by the columns of $\mathbb{V}$. Such a projection operator is denoted here by $\Pi_{\mathbb{V}}^{\perp}$.

Let $\mathbf{c}_e(\mathbf{u}_{N_h}^s) = \mathbb{L}_e^T \mathbf{b}_e(\mathbb{L}_e \mathbf{u}_{N_h}^s)$ denote the contribution of the mesh entity e to $\mathbf{b}_{N_h}(\mathbf{u}_{N_h}^s)$. Given a collection of n_s high-dimensional solution snapshots $\mathcal{S} = \{\mathbf{u}_{N_h}^s\}_{s=1}^{n_s}$, consider

$$\mathbf{b}_n(\Pi_{\mathbb{V}}^{\perp}\mathbf{u}_{N_h}^s) = \sum_{e \in \mathcal{E}} \mathbb{W}^T \mathbf{c}_e(\Pi_{\mathbb{V}}^{\perp}\mathbf{u}_{N_h}^s), \quad s = 1, \ldots, n_s.$$

The above expression can be written in matrix form as

$$\mathbb{C}\mathbf{1} = \mathbf{d},$$

where

$$\mathbb{C} = \begin{pmatrix} \mathbb{W}^T \mathbf{c}_1(\Pi_{\mathbb{V}}^{\perp}\mathbf{u}_{N_h}^1) & \cdots & \mathbb{W}^T \mathbf{c}_{n_e}(\Pi_{\mathbb{V}}^{\perp}\mathbf{u}_{N_h}^1) \\ \vdots & \ddots & \vdots \\ \mathbb{W}^T \mathbf{c}_1(\Pi_{\mathbb{V}}^{\perp}\mathbf{u}_{N_h}^{n_s}) & \cdots & \mathbb{W}^T \mathbf{c}_{n_e}(\Pi_{\mathbb{V}}^{\perp}\mathbf{u}_{N_h}^{n_s}) \end{pmatrix} \in \mathbb{R}^{(nn_s) \times n_e}, \tag{5.63}$$

$$\mathbf{d} = \begin{pmatrix} \mathbf{b}_n(\Pi_{\mathbb{V}}^{\perp}\mathbf{u}_{N_h}^1) \\ \vdots \\ \mathbf{b}_n(\Pi_{\mathbb{V}}^{\perp}\mathbf{u}_{N_h}^{n_s}) \end{pmatrix} \in \mathbb{R}^{(nn_s)}, \tag{5.64}$$

and $\mathbf{1}$ is the vector whose entries are all equal to 1. It follows that the hyperreduced ECSW approximation $\mathbf{b}_n \approx \tilde{\mathbf{b}}_n$ of the form given in (5.60) can be written in matrix form as

$$\mathbb{C}\boldsymbol{\xi}^{\star} \approx \mathbf{d}, \tag{5.65}$$

where $\boldsymbol{\xi}^{\star} \in \mathbb{R}^{n_e}$ denotes the vector of element weights extended to the entire mesh represented by $\mathcal{E}$. Therefore, this vector contains 0 in each of its rows associated with a mesh entity $e \in \mathcal{E} \backslash \bar{\mathcal{E}}$.

For large-scale high-dimensional models, the practical number of precomputed reduced, nonlinear, force balance snapshots n_s and the desired dimension of the PMOR n are such that $nn_s < n_e$. Hence, for all practical purposes, the linear system of equations (5.65) can be considered to be often underdetermined.

The result (5.65) suggests that the pair of minimal subset of sampled elements $\tilde{\mathcal{E}}$ representing the desired reduced mesh and associated vector of element weights $\boldsymbol{\xi}^\star$ that delivers sufficiently accurate hyperreduced approximations of the forms given in (5.60) and (5.61) is given by the solution of the following optimization problem:

$$\begin{cases} \boldsymbol{\xi}^\star = \arg\min_{\boldsymbol{\xi}\geq 0} \|\boldsymbol{\xi}\|_0 & \text{s.t.} \quad \|\mathbb{C}\boldsymbol{\xi} - \mathbf{d}\|_2 \leq \tau\|\mathbf{d}\|_2 & \text{if } \mathbb{W}^T\mathbf{K}_{N_h}\mathbb{V} \text{ is SPD,} \\ \boldsymbol{\xi}^\star = \arg\min \|\boldsymbol{\xi}\|_0 & \text{s.t.} \quad \|\mathbb{C}\boldsymbol{\xi} - \mathbf{d}\|_2 \leq \tau\|\mathbf{d}\|_2 & \text{otherwise,} \end{cases} \tag{5.66}$$

where $\tau \in \mathbb{R}^+$ is a small, relative tolerance that can be used to control the accuracy of the resulting hyperreduction. Unfortunately, both optimization problems described in (5.66) are NP-hard. Therefore, the solution of either of these two problems is infeasible for practical meshes.

Alternatively, *inexact* solutions to three different convex approximations of (5.66) that promote sparsity in the solution can be considered [14]:

1. Approximation A1 below, which transforms problem (5.66) into a (nonnegative) least-squares (NNLS) problem

$$\begin{cases} \boldsymbol{\xi}^\star \approx \arg\min_{\boldsymbol{\xi}\geq 0} \frac{1}{2}\|\mathbb{C}\boldsymbol{\xi} - \mathbf{d}\|_2^2 & \text{if } \mathbb{W}^T\mathbf{K}_{N_h}\mathbb{V} \text{ is SPD,} \\ \boldsymbol{\xi}^\star \approx \arg\min \frac{1}{2}\|\mathbb{C}\boldsymbol{\xi} - \mathbf{d}\|_2^2 & \text{otherwise.} \end{cases} \tag{5.67}$$

2. Approximation A2, which is based on the l_1-norm and transforms the original optimization problem into a (nonnegative) variant of the basis pursuit problem [16]

$$\begin{cases} \boldsymbol{\xi}^\star \approx \arg\min_{\boldsymbol{\xi}\geq 0} \|\boldsymbol{\xi}\|_1 & \text{s.t.} \quad \mathbb{C}\boldsymbol{\xi} = \mathbf{d} & \text{if } \mathbb{W}^T\mathbf{K}_{N_h}\mathbb{V} \text{ is SPD,} \\ \boldsymbol{\xi}^\star \approx \arg\min \|\boldsymbol{\xi}\|_1 & \text{s.t.} \quad \mathbb{C}\boldsymbol{\xi} = \mathbf{d} & \text{otherwise.} \end{cases} \tag{5.68}$$

3. Approximation A3 below, which corresponds to transforming problem (5.66) into a (nonnegative) regularized, least-squares problem

$$\begin{cases} \boldsymbol{\xi}^\star \approx \arg\min_{\boldsymbol{\xi}\geq 0} \|\mathbb{C}\boldsymbol{\xi} - \mathbf{d}\|_2^2 + \lambda\|\boldsymbol{\xi}\|_1 & \text{if } \mathbb{W}^T\mathbf{K}_{N_h}\mathbb{V} \text{ is SPD,} \\ \boldsymbol{\xi}^\star \approx \arg\min \|\mathbb{C}\boldsymbol{\xi} - \mathbf{d}\|_2^2 + \lambda\|\boldsymbol{\xi}\|_1 & \text{otherwise,} \end{cases} \tag{5.69}$$

where λ is a positive penalty parameter.

In each of these three approximations, the inexactness of the solution specifically refers to the fact that the optimal solution is not required to satisfy exactly the Karush–Kuhn–Tucker conditions, but is considered to be acceptable if it satisfies instead the following conditions:

$$\begin{cases} \|\mathbb{C}\boldsymbol{\xi} - \mathbf{d}\|_2 \leq \tau\|\mathbf{d}\|_2 & \text{and} \quad \boldsymbol{\xi} \geq 0 & \text{if } \mathbb{W}^T\mathbf{K}_N\mathbb{V} \text{ is SPD,} \\ \|\mathbb{C}\boldsymbol{\xi} - \mathbf{d}\|_2 \leq \tau\|\mathbf{d}\|_2 & & \text{otherwise.} \end{cases} \tag{5.70}$$

Each of the optimization problems (5.67), (5.68), and (5.69) can be solved using a parallel, iterative, active set algorithm equipped with (5.70) as a stopping criterion. Detailed descriptions of such algorithms can be found in [14], where it was shown that the NNLS approach (5.67) (and its active set algorithm) is by far the fastest and most computationally efficient approach for performing mesh sampling and weighting.

For completeness, Algorithm 5.4 summarizes the computation of the reduced mesh represented by $\widetilde{\mathcal{E}}$ and its associated set of element weights $\{\xi_e^{\star}\}_{e=\widetilde{e}_1}^{\widetilde{e}_{\widetilde{n}_e}}$.

Algorithm 5.4: ECSW: offline and online phases.

Offline phase

Input: $\mathcal{S}, \tau$

Output: $\widetilde{\mathcal{E}}, \{\xi_e^{\star}\}_{e=\widetilde{e}_1}^{\widetilde{e}_{\widetilde{n}_e}}$

  1: Assemble $\mathbb{C}$ from $\mathcal{S}$ using (5.63)

  2: $\mathbf{d} \leftarrow \mathbb{C}\mathbf{1}$

  3: Solve (5.67), (5.68), or (5.69) for $\boldsymbol{\xi}^{\star}$

  4: $\widetilde{\mathcal{E}} \leftarrow \text{indices}(\boldsymbol{\xi}^{\star} > 0)$

  5: $\{\xi_e^{\star}\}_{e=\widetilde{e}_1}^{\widetilde{e}_{\widetilde{n}_e}} \leftarrow \boldsymbol{\xi}^{\star}(\widetilde{\mathcal{E}})$

Online phase

Input: $\boldsymbol{\mu}^{\star}, \mathbf{u}_n(t; \boldsymbol{\mu}^{\star}), \widetilde{\mathcal{E}}, \{\xi_e^{\star}\}_{e=\widetilde{e}_1}^{\widetilde{e}_{\widetilde{n}_e}}$

Output: $\widetilde{\mathbf{b}}_n(\mathbf{u}_n(t; \boldsymbol{\mu}^{\star}); \boldsymbol{\mu}^{\star}), \widetilde{\mathbf{K}}_n(\mathbf{u}_n(t; \boldsymbol{\mu}^{\star}); \boldsymbol{\mu}^{\star})$

  1: Compute $\widetilde{\mathbf{b}}_n(\mathbf{u}_n(t; \boldsymbol{\mu}^{\star}); \boldsymbol{\mu}^{\star})$ using (5.60)

  2: Compute $\widetilde{\mathbf{K}}_n(\mathbf{u}_n(t; \boldsymbol{\mu}^{\star}); \boldsymbol{\mu}^{\star})$ using (5.61)

Remark 5.5. If the semi-discretization is of the FV or FD type and, $\forall e \in \widetilde{\mathcal{E}}$, the evaluation of the quantities $\mathbf{b}_e(\mathbb{L}_{e+}\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ and $\mathbb{K}_e(\mathbb{L}_{e+}\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ requires the localization matrices $\mathbb{L}_{e+} \forall e \in \widetilde{\mathcal{E}}$, the construction of the reduced mesh must be upgraded as follows. After $\widetilde{\mathcal{E}}$ has been computed as described above, it must be augmented with the entities that define for each $e \in \widetilde{\mathcal{E}}$ the Boolean matrix $\mathbb{L}_{e+}$ so that all hyperreduced computations can be performed completely on the upgraded reduced mesh. This process is similar to that described in Section 5.4.2.1.5, where the sample index set $\mathcal{I}$ is augmented with the entities required to construct the sampled quantities at these indices in order to form the final reduced mesh. This additional step is not required in the case of an FE semi-discretization, because in this case, the evaluations of $\mathbf{b}_e(\mathbb{L}_e\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ and $\mathbb{K}_e(\mathbb{L}_e\mathbb{V}\mathbf{u}_n(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ are local to the element e.

5.4.2.2.3 Structure-preserving property and significance

Let

$$N_h' = N_h/2,$$

$$\mathbf{u}_{N_h}(t; \boldsymbol{\mu}) = \begin{pmatrix} \dot{\mathbf{q}}_{N_h'}(t; \boldsymbol{\mu}) \\ \mathbf{q}_{N_h'}(t; \boldsymbol{\mu}) \end{pmatrix},$$

$$\mathbf{f}_{N_h}\big(\mathbf{u}_{N_h}(t; \boldsymbol{\mu}); \boldsymbol{\mu}\big) = \begin{pmatrix} \mathbf{f}_{N_h'}^{\mathrm{int}}(\mathbf{q}_{N_h'}(t; \boldsymbol{\mu}); \boldsymbol{\mu}) + \mathbf{f}_{N_h'}^{\mathrm{diss}}(\dot{\mathbf{q}}_{N_h'}(t; \boldsymbol{\mu}); \boldsymbol{\mu}) \\ -\dot{\mathbf{q}}_{N_h'}(t; \boldsymbol{\mu}) \end{pmatrix}, \qquad (5.71)$$

$$\mathbf{g}_{N_h}(t; \boldsymbol{\mu}) = \begin{pmatrix} \mathbf{f}_{c_{N_h'}}^{\mathrm{ext}}(t; \boldsymbol{\mu}) + \mathbf{f}_{nc_{N_h'}}^{\mathrm{ext}}(t; \boldsymbol{\mu}) \\ 0 \end{pmatrix},$$

$$\mathbb{M}_{N_h}(\boldsymbol{\mu}) = \begin{pmatrix} \mathbb{M}_{N_h'}(\boldsymbol{\mu}) & 0 \\ 0 & \mathbb{I}_{N_h'} \end{pmatrix},$$

where:
- $\mathbf{q}_{N_h'}(t; \boldsymbol{\mu})$ denotes the parametric, displacement/rotation vector associated with a parametric FOM resulting from the FE semi-discretization of a nonlinear, non-conservative, *second-order dynamical system* such as, for example, a nonlinear, nonconservative, structural dynamics system, and $\dot{\mathbf{q}}_{N_h'}(t; \boldsymbol{\mu})$ denotes the corresponding velocity vector.
- $\mathbf{f}_{N_h'}^{\mathrm{int}}(\mathbf{q}_{N_h'}(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ denotes the parametric, nonlinear, *true* internal force vector associated with the aforementioned FE-based FOM, and the word "true" is used here and throughout the remainder of this section to distinguish a newly defined entity from its generic counterpart introduced at the beginning of this chapter. This force vector usually derives from a parametric, nonlinear, internal potential energy $\mathcal{V}^{\mathrm{int}}(\mathbf{q}_{N_h'}(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ – that is,

$$\mathbf{f}_{N_h'}^{\mathrm{int}}(\mathbf{q}_{N_h'}(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = -\frac{\partial \mathcal{V}^{\mathrm{int}}}{\partial \mathbf{q}_{N_h'}}^{T}(\mathbf{q}_{N_h'}(t; \boldsymbol{\mu}); \boldsymbol{\mu}). \qquad (5.72)$$

- $\mathbf{f}_{N_h'}^{\mathrm{diss}}(\mathbf{q}_{N_h'}(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ denotes the parametric, nonlinear, *dissipative* force vector associated with the aforementioned FE-based FOM. Typically, this vector contains dissipative forces that remain parallel and in opposite direction to the velocity vector, depend on its modulus, and are associated with a parametric, nonlinear, dissipation function $\mathcal{D}(\dot{\mathbf{q}}_{N_h'}(t; \boldsymbol{\mu}); \boldsymbol{\mu})$ that is a homogeneous function of order m in the velocity vector – that is,

$$\dot{\mathbf{q}}_{N_h'}^{T}(t; \boldsymbol{\mu})\frac{\partial \mathcal{D}}{\partial \dot{\mathbf{q}}_{N_h'}}(\dot{\mathbf{q}}_{N_h'}(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = m\mathcal{D}(\dot{\mathbf{q}}_{N_h'}(t; \boldsymbol{\mu}); \boldsymbol{\mu}).$$

For example, $m = 1$ corresponds to the case of dry friction, $m = 2$ to viscous damping, and $m = 3$ to aerodynamic drag. Hence, the parametric, nonlinear, dissipative

force vector can be written as

$$\mathbf{f}^{\text{diss}}_{N_h'}(\dot{\mathbf{q}}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) = -\frac{\partial\mathcal{D}^T}{\partial\dot{\mathbf{q}}_{N_h'}}(\dot{\mathbf{q}}'_{N_h}(t;\boldsymbol{\mu});\boldsymbol{\mu}). \tag{5.73}$$

- $\mathbf{f}^{\text{ext}}_{c_{N_h'}}(t;\boldsymbol{\mu})$ denotes the parametric, conservative force vector deriving from a parametric, external potential $\mathcal{V}^{\text{ext}}(\mathbf{q}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu})$ – that is,

$$\mathbf{f}^{\text{ext}}_{c_{N_h'}}(t;\boldsymbol{\mu}) = -\frac{\partial\mathcal{V}^{\text{ext}}{}^T}{\partial\mathbf{q}_{N'}}(\mathbf{q}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) \Leftrightarrow \mathcal{V}^{\text{ext}}(\mathbf{q}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) = -\mathbf{q}^T_{N_h'}\mathbf{f}^{\text{ext}}_{c_{N_h'}}(t;\boldsymbol{\mu}). \tag{5.74}$$

- $\mathbf{f}^{\text{ext}}_{nc_{N_h'}}(t;\boldsymbol{\mu})$ denotes the parametric, nonconservative force vector.
- $\mathbb{M}_{N_h'}(\boldsymbol{\mu})$ denotes the parametric, true mass matrix of dimension N_h' associated with the aforementioned FOM.

In this context, the semi-discrete, parametric FOM (5.1) models a nonlinear, nonconservative, second-order dynamical system – for example, a nonlinear, nonconservative, structural dynamics system – whose governing FE equation

$$\mathbb{M}_{N_h'}(\boldsymbol{\mu})\ddot{\mathbf{q}}_{N_h'} + \mathbf{f}^{\text{int}}_{N_h'}(\mathbf{q}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) + \mathbf{f}^{\text{diss}}_{N_h'}(\dot{\mathbf{q}}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \mathbf{f}^{\text{ext}}_{c_{N_h'}}(t;\boldsymbol{\mu}) + \mathbf{f}^{\text{ext}}_{nc_{N_h'}}(t;\boldsymbol{\mu}) \tag{5.75}$$

has been rewritten in first-order form. Note that the emphasis here on the case of a nonconservative system is simply because it is more general than the particular case of a conservative system.

Let

$$\mathcal{V}(\mathbf{q}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \mathcal{V}^{\text{int}}(\mathbf{q}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) + \mathcal{V}^{\text{ext}}(\mathbf{q}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) \tag{5.76}$$

denote the parametric, nonlinear, *total* potential associated with the second-order dynamical system represented by (5.75), and let

$$\mathcal{T}(\dot{\mathbf{q}}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) = \frac{1}{2}\dot{\mathbf{q}}^T_{N_h'}(t;\boldsymbol{\mu})\mathbb{M}_{N_h'}(\boldsymbol{\mu})\dot{\mathbf{q}}_{N_h'}(t;\boldsymbol{\mu}) \tag{5.77}$$

denote its kinetic energy. For this second-order dynamical system, Hamilton's principle can be written as

$$\dot{\mathcal{T}}(\dot{\mathbf{q}}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) + \dot{\mathcal{V}}(\mathbf{q}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) = -m\mathcal{D}(\dot{\mathbf{q}}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) + \dot{\mathbf{q}}^T_{N'}\mathbf{f}^{\text{ext}}_{nc_{N_h'}}(t;\boldsymbol{\mu}), \tag{5.78}$$

where $\mathcal{T}$ is the kinetic energy defined in (5.77), $\mathcal{V}$ is the total potential defined in (5.76), (5.72), and (5.74), and $\mathcal{D}$ is the dissipation function defined in (5.73). The associated Lagrange equation of motion can be written as

$$-\frac{d}{dt}\left(\frac{\partial\mathcal{T}}{\partial\dot{\mathbf{q}}_{N_h'}}\right)(\dot{\mathbf{q}}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) - \frac{\partial\mathcal{V}^T}{\partial\mathbf{q}_{N_h'}}(\mathbf{q}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) - \frac{\partial\mathcal{D}^T}{\partial\dot{\mathbf{q}}_{N_h'}}(\dot{\mathbf{q}}_{N_h'}(t;\boldsymbol{\mu});\boldsymbol{\mu})$$
$$+ \mathbf{f}^{\text{ext}}_{nc_{N_h'}}(t;\boldsymbol{\mu}) = 0. \tag{5.79}$$

In the special context defined by (5.71), the above FE-based equation is equivalent to its counterpart given in (5.75). More importantly, equations (5.78) and (5.79) collectively define a *mathematical structure* that is referred to here as the Lagrangian structure associated with Hamilton's principle. In [24], it was shown that ECSW preserves this structure as in the context of (5.71), the kinetic energy $\widetilde{\mathcal{T}}$ based on the subspace approximation (5.17) (adjusted to the dimension n'), and the hyperreduced total potential $\widetilde{\mathcal{T}}$ and dissipative function $\widetilde{\mathcal{D}}$ obtained by the application of the ECSW approximations of the form given in (5.60) to $\mathcal{T}$ and $\mathcal{D}$, respectively, satisfy

$$\dot{\widetilde{\mathcal{T}}}(\dot{\mathbf{q}}_{n'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) + \dot{\widetilde{\mathcal{V}}}(\mathbf{q}_{n'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) = -m\widetilde{\mathcal{D}}(\dot{\mathbf{q}}_{n'}(t;\boldsymbol{\mu});\boldsymbol{\mu}) + \dot{\mathbf{q}}_{n'}^{T}\mathbf{f}_{nc_{n'}}^{\mathrm{ext}}(t;\boldsymbol{\mu}), \tag{5.80}$$

which is similar to (5.78). In (5.80), $n' = n/2$, $\mathbf{q}_{n'}$ is the vector of reduced (or generalized) coordinates associated with $\mathbf{q}_{N_h'}$, and $\mathbf{f}_{nc_{n'}}^{\mathrm{ext}}(t;\boldsymbol{\mu})$ is the reduced vector of nonconservative external forces.

Hence, for parametric, nonlinear, second-order dynamical systems, ECSW is a structure-preserving hyperreduction method. A major consequence is that in the context defined by (5.71), a preferred time integrator applied to the time discretization of the parametric, nonlinear PROM (5.6) hyperreduced by ECSW will exhibit the same, if not better, numerical stability properties as those that it exhibits when applied to the time discretization of the underlying FOM (5.1) – see [24] for the justification. Specifically, if the preferred time integrator is energy-conserving and unconditionally stable when applied to the time discretization of (5.1), it is *guaranteed* to be unconditionally stable when applied to the hyperreduction by ECSW of the parametric, nonlinear PROM (5.6). The latter result follows directly from the preservation by ECSW of the Lagrangian structure associated with Hamilton's principle.

5.5 Applications

Here, the DEIM and ECSW are illustrated with both academic and real-world, parametric and nonparametric, linear and nonlinear applications for which hyperreduction is necessary for achieving computational efficiency. As stated before, these two methods represent the state of the art of approximate-then-project and project-then-approximate hyperreduction methods, respectively. Throughout this section, a hyperreduced PROM is referred to as an HPROM, and all reported computations are performed in double precision arithmetic.

5.5.1 Hyperreduction of a parametric Helmholtz-elasticity model

Here, the two-dimensional, parametric, acoustic scattering problem graphically depicted in Figure 5.3 is considered. This problem is characterized by: a two-dimensional
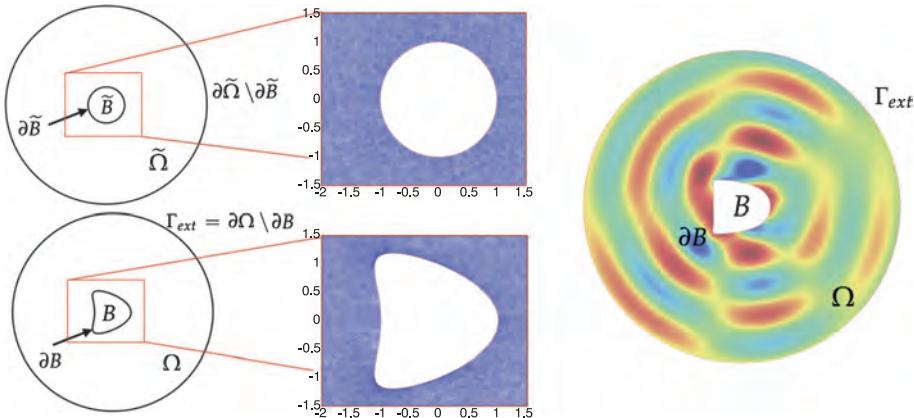
Figure 5.3: Parametric, one-way coupled, acoustic scattering problem: Variable shape scatterer, artificial boundary, computational domain and its discretization, and amplitude of a scattered wave.

obstacle B parameterized by the shape of its boundary ∂B; a computational domain Ω delimited by the parametric boundary ∂B and a nonparametric artificial boundary $\Gamma_{\text{ext}} = \partial\Omega\backslash\partial B$, where a local absorbing boundary condition is applied; and a planar incident wave parameterized by a variable direction $(\cos a \quad \sin a)^T$ and a variable wave number κ. It is modeled by an FE-based FOM that is a linear version of the second-order FOM (5.75) where the solution $\mathbf{q}_{N_h'}(t; \boldsymbol{\mu})$ is sought after in the form $\mathbf{q}_{N_h'}(t; \boldsymbol{\mu}) = \mathbf{q}_{N_h'}^{\text{amp}}(\boldsymbol{\mu})e^{-I\omega t}$, where the superscript amp designates the amplitude of a vector, I denotes the pure imaginary number ($I^2 = -1$), $\omega = \kappa c$ is a specified circular frequency, and c is the speed of sound in the medium surrounding the obstacle B; $\mathbf{f}_{N_h'}^{\text{int}}(\mathbf{q}_{N_h'}(t; \boldsymbol{\mu})) = \mathbb{K}_{N_h'}(\boldsymbol{\mu})\mathbf{q}_{N_h'}(t; \boldsymbol{\mu}) = \mathbb{K}_{N_h'}(\boldsymbol{\mu})\mathbf{q}_{N_h'}^{\text{amp}}(\boldsymbol{\mu})e^{-I\omega t}$; $\mathbf{f}_{N_h'}^{\text{diss}}(\dot{\mathbf{q}}_{N_h'}(t; \boldsymbol{\mu}); \boldsymbol{\mu}) = \mathbb{S}_{N_h'}(\boldsymbol{\mu})\dot{\mathbf{q}}_{N_h'}(t; \boldsymbol{\mu}) = -I\omega\mathbb{S}_{N_h'}(\boldsymbol{\mu})\mathbf{q}_{N_h'}^{\text{amp}}(\boldsymbol{\mu})e^{-I\omega t}$;and $\mathbf{f}_{nc_{N_h'}}^{\text{ext}}(t; \boldsymbol{\mu}) = \mathbf{f}_{N_h'}^{\text{amp}}(\boldsymbol{\mu})e^{-I\omega t}$. Hence, the aforementioned acoustic scattering problem is represented here by a linear FOM of the Helmholtz type

$$(\mathbb{K}_{N_h'}(\boldsymbol{\mu}) - \omega^2\mathbb{M}_{N_h'}(\boldsymbol{\mu}) - I\omega\mathbb{S}_{N_h'}(\boldsymbol{\mu}))\mathbf{q}_{N_h'}^{\text{amp}}(\boldsymbol{\mu}) = \mathbf{f}_{N_h'}^{\text{amp}}(\boldsymbol{\mu}), \qquad (5.81)$$

where $\mathbb{S}_{N_h'}(\boldsymbol{\mu})$ is a real-valued sparse matrix associated with the discretization of the local absorbing boundary condition and therefore has nonzero entries on Γ_{ext} only; and $\mathbf{f}_{N_h'}^{\text{amp}}(\boldsymbol{\mu})$ arises from the treatment of the displacement (Dirichlet) boundary condition on ∂B associated with the incident wave. This FOM is representative of simplified versions of Helmholtz problems that arise in many applications pertaining to sonar and radar design, medical imaging, and nondestructive testing.

Specifically, the shape of ∂B is parameterized as follows:

$$\partial B(\boldsymbol{\zeta}) = \partial\tilde{B} + \Gamma_B(\zeta_1, \zeta_2, \zeta_3, \zeta_4),$$

where $\widetilde{B}$ is a disk of center $(0, 0)$ and radius $1\,\mathrm{m}$ – and therefore $\partial\widetilde{B}$ is a circle of center $(0, 0)$ and radius $1\,\mathrm{m}$ – Γ_B is the shape deformation function defined by the following parameterized two-dimensional displacement vector [18]:

$$\mathbf{u}_{\Gamma_B}(\boldsymbol{\zeta}) = \begin{pmatrix} \zeta_1 \cos 2s + \zeta_3 \cos 4s - (\zeta_1 + \zeta_3) \\ \zeta_2 \sin s + \zeta_4 \sin 3s \end{pmatrix},$$

where $s = \tan^{-1}(\frac{y}{x}) \in [0, 2\pi]$, x and y denote the coordinates of a generic point on the circle $\partial\widetilde{B}$, and $\boldsymbol{\zeta} = (\zeta_1 \quad \zeta_2 \quad \zeta_3 \quad \zeta_4)^T \in \mathcal{P}_{\boldsymbol{\zeta}} \subset \mathbb{R}^4$.

In all cases, the artificial boundary Γ_{ext} is chosen to be the circle of center $(0, 0)$ and radius $5\,\mathrm{m}$; hence, $\forall \boldsymbol{\zeta} \in \mathcal{P}_{\boldsymbol{\zeta}}$, $\partial\Omega \backslash \partial B = \partial\widetilde{\Omega} \backslash \partial\widetilde{B} = \Gamma_{\text{ext}}$, where $\widetilde{\Omega}$ denotes the reference computational domain defined by the annular disk of internal radius equal to $1\,\mathrm{m}$ and external radius equal to $5\,\mathrm{m}$. This reference domain is discretized by a reference mesh $\mathscr{M}(\mathbf{0})$ with $n_e = 142{,}168$ linear triangular elements. For each nonzero value of the deformational vector $\mathbf{u}_{\Gamma_B}(\boldsymbol{\zeta})$, this mesh is deformed using the structural analogy method described in [37] in order to obtain the mesh $\mathscr{M}(\boldsymbol{\zeta}, \eta)$ that conforms to the parametric boundary $\partial B(\boldsymbol{\zeta})$ and nonparametric artificial boundary Γ_{ext}. To this end, two displacement DOFs $u_{j_x}^\Omega$ and $u_{j_y}^\Omega$ in the x- and y-directions, respectively, are attached to each node of $\mathscr{M}(\mathbf{0})$, in addition to the DOF q_j governed by the FOM (5.81). For each queried parameter point $\boldsymbol{\zeta} \in \mathcal{P}_{\boldsymbol{\zeta}}$, the following high-dimensional problem is constructed and solved:

$$\mathbb{K}_{N_h{}^\Omega}(\boldsymbol{\zeta}, \eta)\mathbf{u}_{N_h{}^\Omega}(\boldsymbol{\zeta}, \eta) = \mathbf{f}_{N_h{}^\Omega}(\boldsymbol{\zeta}, \eta), \tag{5.82}$$

where $\mathbb{K}_{N_h{}^\Omega}(\boldsymbol{\zeta}, \eta)$ is an FE stiffness matrix of dimension $N_h{}^\Omega$ associated with the elasticity-based structural analogy method described in [37], η is a user-defined numerical parameter of this method, $\mathbf{u}_{N_h{}^\Omega}(\boldsymbol{\zeta}, \eta)$ is the vector of displacement DOFs of $\mathscr{M}(\mathbf{0})$, and $\mathbf{f}_{N_h{}^\Omega}(\boldsymbol{\zeta}, \eta)$ arises from the treatment of the displacement (Dirichlet) boundary condition on ∂B associated with the deformation of this boundary. Then, $\mathscr{M}(\boldsymbol{\zeta}, \eta)$ is constructed by updating the position of the nodes of $\mathscr{M}(\mathbf{0})$ using the computed vector of displacement DOFs $\mathbf{u}_{N_h{}^\Omega}(\boldsymbol{\zeta}, \eta)$. This approach ensures that $\forall \boldsymbol{\zeta} \in \mathcal{P}_{\boldsymbol{\zeta}}$, the mesh topology of the reference mesh $\mathscr{M}(\mathbf{0})$ is preserved by $\mathscr{M}(\boldsymbol{\zeta}, \eta)$. Therefore, it simplifies the PMOR of the FOM (5.82) whose operators depend on $\mathscr{M}(\boldsymbol{\zeta}, \eta)$. For $n_e = 142{,}168$, $N_h{}^\Omega = 141{,}256$ and $N_h{}' = 71{,}324$.

Let $\mathcal{P}_{(\boldsymbol{\zeta}, \eta)} \subset \mathbb{R}^5$ denote the five-dimensional parameter space where a typical point $\boldsymbol{\mu}_{(\boldsymbol{\zeta}, \eta)}$ is given by $\boldsymbol{\mu}_{(\boldsymbol{\zeta}, \eta)} = (\zeta_1 \quad \zeta_2 \quad \zeta_3 \quad \zeta_4 \quad \eta)^T$, and let $\mathcal{P} \subset \mathbb{R}^7$ denote the seven-dimensional parameter space where a typical point $\boldsymbol{\mu}$ is given by $\boldsymbol{\mu} = (\zeta_1 \quad \zeta_2 \quad \zeta_3 \quad \zeta_4 \quad \eta \quad a \quad \kappa)^T$, where it is recalled that a and κ parameterize the planar incident wave direction and wave number. The considered ranges for the parameter space $\mathcal{P}$ are:

- $\zeta_1 \in [-1/2, 1/2]$, $\zeta_2 \in [-0.8, 1.2]$, $\zeta_3 \in [-0.05, 0.05]$, $\zeta_4 \in [-0.05, 0.05]$, and $\eta \in [0, 1.4]$;

– $a \in [0, \pi/6]$ and $\kappa \in [2, 4]$.

Then, for each queried parameter point $\boldsymbol{\mu} \in \mathcal{P}$, the solution of the acoustic scattering problem graphically depicted in Figure 5.3 consists in solving the one-way coupled problem defined by the FOM (5.82) and the FOM (5.81), as follows:

–   First, solve the problem (5.82) for $\boldsymbol{\mu}_{(\boldsymbol{\zeta}, \eta)} \subset \boldsymbol{\mu}$ and transform $\mathcal{M}(\mathbf{0})$ into $\mathcal{M}(\boldsymbol{\zeta}, \eta)$ using the computed $\mathbf{u}_{N_h^\Omega}(\boldsymbol{\mu}_{(\boldsymbol{\zeta}, \eta)})$.

–   Next, construct the problem (5.81) for the queried parameter point $\boldsymbol{\mu}$ and solve it to obtain $\mathbf{q}_{N_h'}^{\mathrm{amp}}(\boldsymbol{\mu})$.

Because the parametric stiffness matrix $\mathbb{K}_{N_h^\Omega}(\boldsymbol{\zeta}, \eta)$ of the parametric FOM (5.82) depends implicitly on $\boldsymbol{\mu}_{(\boldsymbol{\zeta}, \eta)}$, – and therefore, $\mathbb{K}_{N_h^\Omega}(\boldsymbol{\zeta}, \eta)$ cannot be explicitly written as an affine function of $\boldsymbol{\mu}_{(\boldsymbol{\zeta}, \eta)}$ – and because the parametric operator $\mathbb{K}_{N_h'}(\boldsymbol{\mu}) - \omega^2 \mathbb{M}_{N_h'}(\boldsymbol{\mu}) - I\omega \mathbb{S}_{N_h'}(\boldsymbol{\mu})$ governing the parametric FOM (5.81) depends on the updated mesh $\mathcal{M}(\boldsymbol{\zeta}, \eta)$ and therefore depends implicitly on $\boldsymbol{\mu}_{(\boldsymbol{\zeta}, \eta)} \subset \boldsymbol{\mu}$, the hyperreduction of any PROMs constructed for the FOM (5.82) and/or the FOM (5.81) is necessary in order to achieve computational efficiency.

The PMOR of the one-way coupled, acoustic scattering problem defined above is performed in two steps as follows. First, a PROM for the mesh motion problem (5.82) is constructed, followed by the construction of a PROM for the Helmholtz problem (5.81). In both cases, the POD and Galerkin projection methods are used for this purpose. Specifically, the LATIN hypercube sampling method is applied to sample $\mathcal{P}_{(\boldsymbol{\zeta}, \eta)} \subset \mathbb{R}^5$ in 50 points, and 50 solution snapshots are computed at these points and compressed using SVD. This leads to a PROM for the mesh motion problem of dimension $n_u = 10$. Then, the DEIM is applied to hyperreduce this PROM using seven POD basis vectors for approximating the action of an instance of the left-hand side matrix $\mathbb{K}_{N_h^\Omega}(\boldsymbol{\zeta}, \eta)$ on a vector of dimension N_h^Ω, and 21 POD basis vectors for representing a right-hand side vector of the form $\mathbf{f}_{N_h^\Omega}(\boldsymbol{\zeta}, \eta)$. A relative tolerance of 10^{-5} is used to truncate all computed POD bases [32], which is reasonable given the decay of the normalized singular values reported in Figure 5.4 for each computed snapshot matrix.

Next, $\mathcal{P} \subset \mathbb{R}^7$ is similarly sampled at 250 points in order to compute 250 solution snapshots of the Helmholtz problem (5.81). To this end, the PROM constructed for the mesh motion problem (5.82) is used for solving this problem at each sampled point and updating accordingly the position of the nodes of $\mathcal{M}(\mathbf{0})$. The 250 computed solution snapshots of (5.81) are then compressed using SVD to obtain a global ROB and associated Galerkin-PROM of dimension $n_q = 137$. Then, this PROM is hyperreduced using DEIM, 206 POD basis vectors for approximating the action of an instance of the left-hand side matrix $(\mathbb{K}_{N_h'}(\boldsymbol{\mu}) - \omega^2 \mathbb{M}_{N_h'}(\boldsymbol{\mu}) - I\omega \mathbb{S}_{N_h'}(\boldsymbol{\mu}))$ on a vector of dimension N_h', and 229 POD basis vectors for representing a right-hand side vector of the form $\mathbf{f}_{N_h'}^{\mathrm{amp}}(\boldsymbol{\mu})$. In this case, a relative tolerance of 10^{-7} is used to truncate all computed POD bases: This tolerance value is reasonable given the decay of the normalized singular
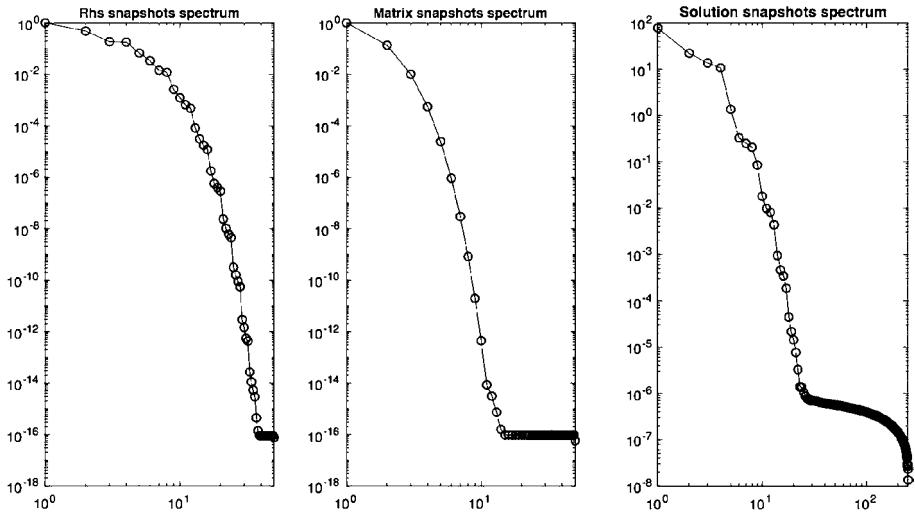
Figure 5.4: Parametric, one-way coupled, acoustic scattering problem: Decay of the normalized singular values of the matrix of snapshots of the right-hand side vector of (5.82) (left), of the action of the left-hand side matrix of (5.82) on a vector (middle), and of solutions of (5.82) (right).
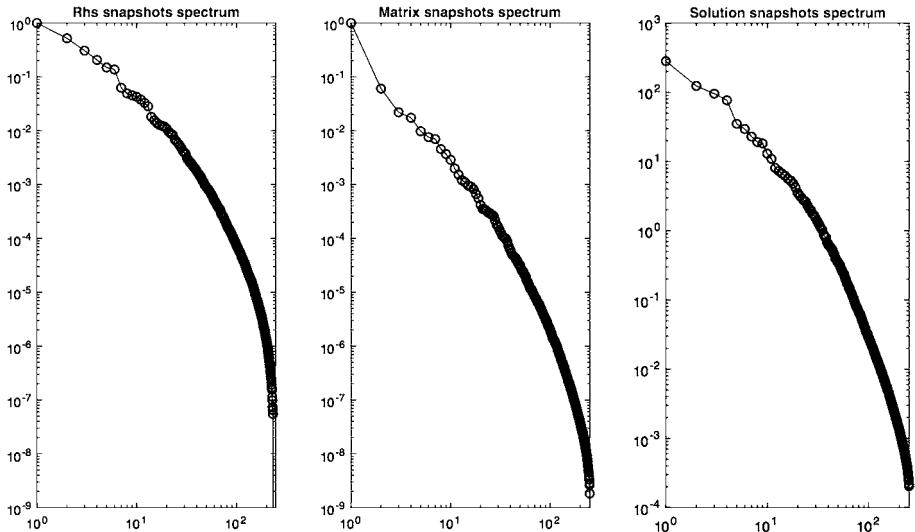
Figure 5.5: Parametric, one-way coupled, acoustic scattering problem: Decay of the normalized singular values of the matrix of snapshots of the right-hand side vector of (5.81) (left), of the action of the left-hand side matrix of (5.81) on a vector (middle), and of solutions of (5.81) (right).

values reported in Figure 5.5 for each computed snapshot matrix. Note that this decay is significantly slower than in the case of the mesh motion problem, which indicates a stronger dependence of the operators and solution of the Helmholtz problem (5.81) on
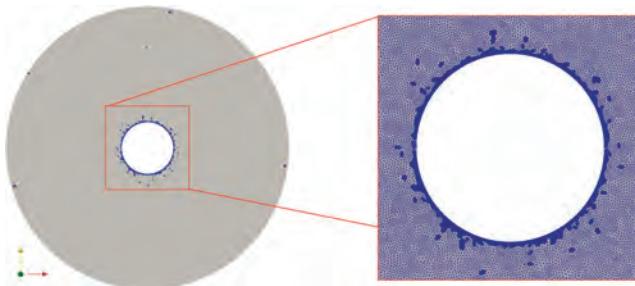
Figure 5.6: Parametric, one-way coupled, acoustic scattering problem: DEIM-based reduced mesh.

their parameters. Figure 5.6 shows the reduced mesh obtained for this one-way coupled Helmholtz problem using the DEIM. This mesh contains 1,391 elements, which corresponds to 0.98 % of the total number of elements n_e of the original mesh. The elements of this reduced mesh are located mainly around the scatterer (obstacle) – that is, in the region of the computational domain where the sensitivity of the solution of problem (5.81) to shape variations is higher.

Figures 5.7 and 5.8 display solutions of the mesh motion and Helmholtz problems computed using the DEIM-based HPROMs for several sampled points of the parameter space $\mathcal{P} \subset \mathbb{R}^7$. Figure 5.7 also reports, for 100 parameter points sampled in $\mathcal{P}_{(\zeta,\eta)} \subset \mathbb{R}^5$, the convergence of the average global relative error of the HPROM-based solution of the mesh motion problem – which is defined here with respect to the FOM-based counterpart solution, measured in the energy norm over the entire mesh, and averaged over the number of sampled parameter points – as a function of the dimension n_u of the
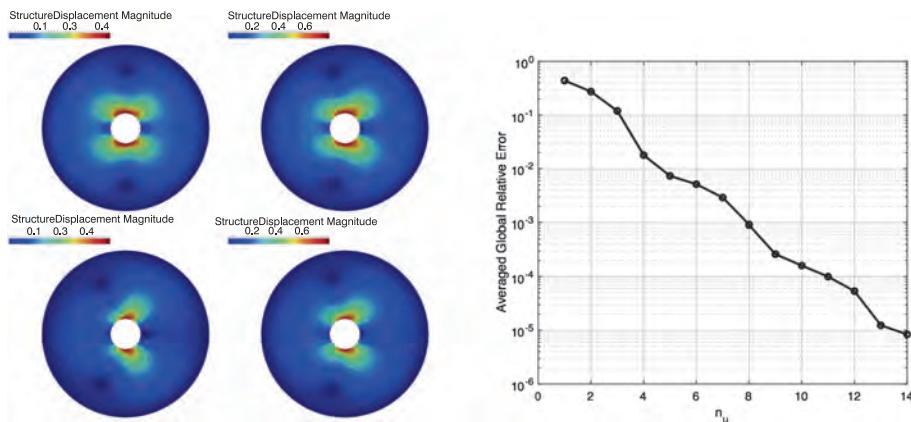
Figure 5.7: Parametric, one-way coupled, acoustic scattering problem: Visualization of the magnitude of the mesh displacement for different parameter values (left); and variation of the relative error of the mesh motion HPROM with the dimension n_u of the associated PROM (right).
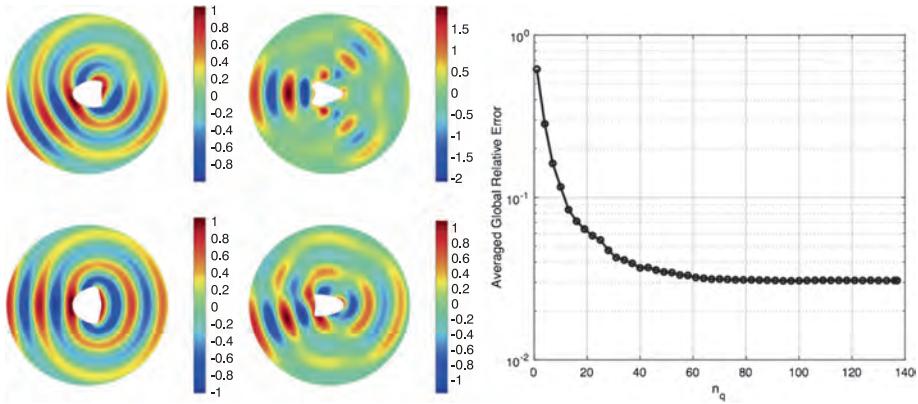
Figure 5.8: Parametric, one-way coupled, acoustic scattering problem: Visualization of the scattered wave amplitude for different parameter values (left); and variation of the relative error of the Helmholtz HPROM with the dimension n_q of the associated PROM (right).

PROM underlying the HPROM. Similarly, Figure 5.8 shows the convergence with n_q of the counterpart error associated with the solution of the Helmholtz problem. The reader can observe that in the case of the Helmholtz problem, the convergence of the aforementioned relative error is significantly slower than that of its counterpart for the mesh motion problem and eventually flattens at roughly $n_q = 80$. This behavior is characteristic of the HPROM error which tends to be dominated by the error introduced by hyperreduction, and which cannot always be decreased by increasing the dimension of the underlying PROM.

Regarding wall-clock performance, the online solution of this one-way coupled, acoustic scattering problem defined at a queried but unsampled parameter point $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^7$ can be computed in 0.43 s using the mesh motion and Helmholtz HPROMs on a workstation with a dual-core Intel Core i5 processor running at 2.8 GHz and 16 GB of memory. On the same computing platform, the online solution of the same one-way coupled problem using the FOMs (5.82) and (5.81) is 24 times slower.

5.5.2 Hyperreduction of a parametric PDE-ODE wildfire model

Next, a time-dependent, nonlinear, two-way coupled, PDE–ordinary differential equation (ODE) system describing the evolution of a wildfire in a domain representing the two-dimensional layer just above the ground surface is considered. This system models a wildfire using balance equations for energy and fuel [39, 30]. In principle, the PDE is the two-dimensional unsteady, advection-diffusion equation governing temperature. However, for demonstrative purposes the advection term is neglected here and therefore the PDE is the two-dimensional unsteady diffusion equation gov-

erning the temperature distribution. The ODE governs the time-dependent fuel supply mass fraction. Two-way coupling between the two equations is performed via non-linear source terms (see [39, 30] for further details). For this problem, the parameters of interest are: the thermal diffusivity, ψ, in m^2/s; the rate of temperature rise at the maximum burning rate, A, in K/s; the proportionality coefficient in the modified Arrhenius law, B, in K; the scaled coefficient of heat transfer to the environment, C, in K^{-1}; and the relative fuel consumption rate, C_S, in s^{-1}. They define a parameter space $\mathcal{P} \subset \mathbb{R}^5$ and are stored in a parameter vector $\boldsymbol{\mu} \in \mathcal{P}$. The parametric solutions of this problem propagate in three areas of localized combustion: a preheated area ahead of the fire; a combustion zone; and a burning region behind the fire. Their efficient computation using PROMs requires hyperreduction, because the coupling source terms are nonlinear and their dependencies on the aforementioned parameters are nonpolynomial.

The computational domain $\Omega \subset \mathbb{R}^2$ is chosen to be a square of side 1,000 m. The ranges of the individual dimensions of $\mathcal{P}$ are set as follows: $0.15 \leq \psi \leq 1.5$; $2 \leq A \leq 20$; $60 \leq B \leq 600$; $4.5 \times 10^{-5} \leq C \leq 4.5 \times 10^{-4}$; and $1.5 \times 10^{-2} \leq C_S \leq 1.5 \times 10^{-1}$. Note that for increasing values of A/C_S, the temperature in the traveling combustion wave increases; for increasing values of ψ, both the width and the speed of the combustion wave increase; and a sustained combustion requires sufficiently small values of C_S.

In order to model a fire located in a small circular region close to the center of Ω, the temperature in this chosen computational domain is initialized as

$$T(\mathbf{x}, 0) = T_c\, e^{-\|\mathbf{x}\|^2/\chi^2} + T_a, \quad \mathbf{x} \in \Omega,$$

where $T_c = 1{,}200$ K, $T_a = 300$ K, and $\chi^2 = 10^4$ m. In order to model the fuel depletion of a fully developed fire, the fuel supply mass fraction is initialized as

$$S(\mathbf{x}, 0) = 1 - e^{-\|\mathbf{x}\|^2/\chi_S^2},$$

where $\chi_S^2 = 5 \times 10^3$ m. For $t > 0$, the reaction heat spreads isotropically, heating the fuel ahead of the wave until the reaction in front of the wave can sustain itself, thus causing the spread of combustion. At the rear, the reaction ceases due to fuel depletion, thus causing the temperature to decrease due to cooling.

A two-dimensional FE model of this coupled PDE-ODE system is constructed using a triangulation of Ω with $n_e = 137{,}820$ elements and $n_n = 69{,}313$ nodes. At each node, two DOFs are attached: one for the temperature, and one for the fuel supply mass fraction, which results in a total number of $N = 138{,}626$ DOFs. Spatial approximation is performed using piecewise linear finite elements. Temporal discretization is performed using the first-order time-accurate backward Euler method, and all nonlinear terms are treated semi-implicitly. All numerical simulations reported below are conducted in the time interval [0, 750] s using the fixed time step $\Delta t = 5$ s.

PMOR is achieved using the POD-based Galerkin projection method. For this purpose, solution snapshots are computed using the parametric FOM described above

at each time step, for 100 randomly sampled points in $\mathcal{P}$. Next, these snapshots are compressed to construct an ROB and then an associated PROM of dimension $n = 12$. The DEIM is applied to hyperreduce the PROM by approximating only the nonlinear terms, as all linear terms of the FOM depend affinely on the problem parameters and therefore do not require hyperreduction. Specifically, a basis of dimension $m = 11$ is constructed using snapshots of the nonlinear (coupling) source terms computed at each time step, for 50 randomly sampled points in $\mathcal{P}$, and a set of 11 interpolation indices $\mathcal{I}$ ($|\mathcal{I}| = m = 11$) is selected. The corresponding reduced mesh $\tilde{\mathcal{E}}$ has only 45 elements – that is, 0.033 % of the number of elements in the original FE mesh. Using the resulting HPROM, the pointwise approximation of the ODE governing the fuel supply mass fraction is advanced only at the DOFs corresponding to the interpolation indices in $\mathcal{I}$, which further decreases the processing time for this application.

Figures 5.9 and 5.10 contrast the FOM-, PROM-, and HPROM-based predictions of the temperature time histories and fuel supply mass fraction at four different spatial locations, for two significantly different parameter configurations. They show that in each case, the FOM, PROM, and HPROM deliver essentially the same results, which indicates that both PMOR- and hyperreduction-induced errors are minimal. These two figures also reveal that the two different parameter configurations lead to very different physical results, which, given also the previous remark, illustrates the robustness of the global PROM and that of the associated HPROM with respect to parameter changes. Figure 5.11 reports for both parameter configurations the temperature distributions predicted at $t = 60$ s using the HPROM and the corresponding absolute errors
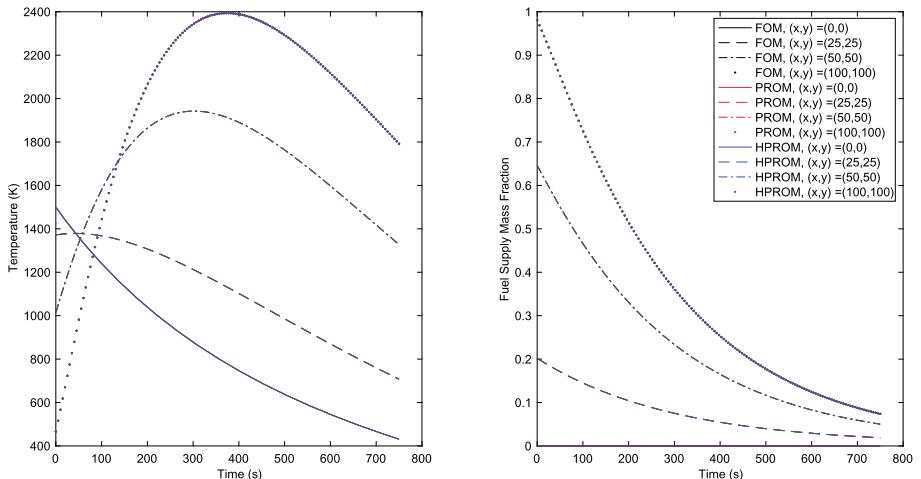
Figure 5.9: Parametric, two-way coupled, PDE-ODE wildfire problem: Temperature (left) and fuel supply mass fraction (right) predicted at four different spatial locations for $\boldsymbol{\mu} = (1.07, 91.19, 258.41, 10^{-4}, 0.0201)^T$.
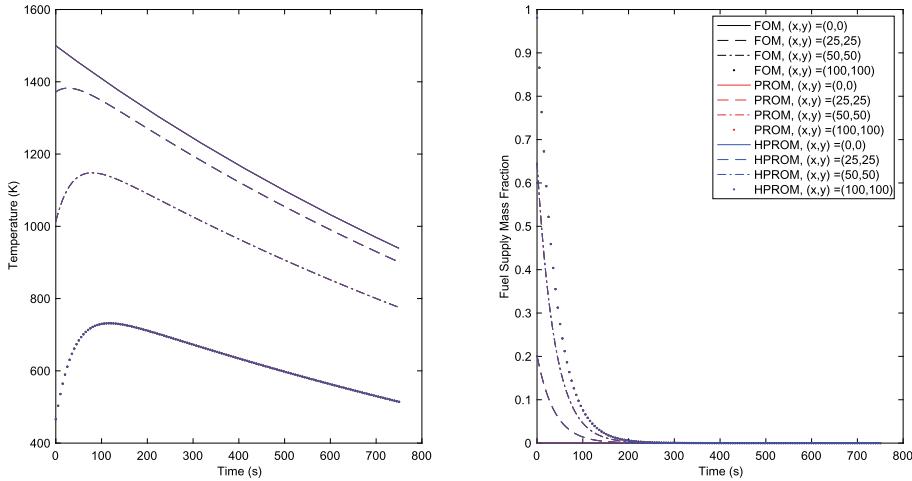
Figure 5.10: Parametric, two-way coupled, PDE-ODE wildfire problem: Temperature (left) and fuel supply mass fraction (right) predicted at four different spatial locations for $\boldsymbol{\mu} = (1.5, 50, 65, 5.5 \times 10^{-5}, 0.15)^T$.

(measured with respect to the FOM-based temperature solutions). It leads to the same conclusions as Figures 5.9 and 5.10.

Table 5.2 reports the wall-clock timings obtained on a workstation with a dual-core Intel Core i5 processor running at 2.8 GHz and 16 GB of memory, the speedup factors, and the average global relative errors of the PROM- and HPROM-based simulations. These errors are defined here with respect to the results of the FOM-based counterpart simulations, measured in the energy norm over the entire mesh, and averaged over 50 different parameter points randomly selected in $\mathcal{P}$. The reported results show that overall, the DEIM-based HPROM maintains the level of accuracy of its underlying PROM. As expected, the PROM does not accelerate the FOM-based simulation by any meaningful factor, but the HPROM delivers a speedup factor of almost 30.

Table 5.2: Parametric, two-way coupled, PDE-ODE wildfire problem: Wall-clock timings on a single core and speedup factors.

	Wall-clock time (s)	Speedup factor	Relative error
FOM	88.4	1.0	–
PROM	48.5	1.82	3.16×10^{-7}
HPROM	3.18	27.8	5.46×10^{-7}
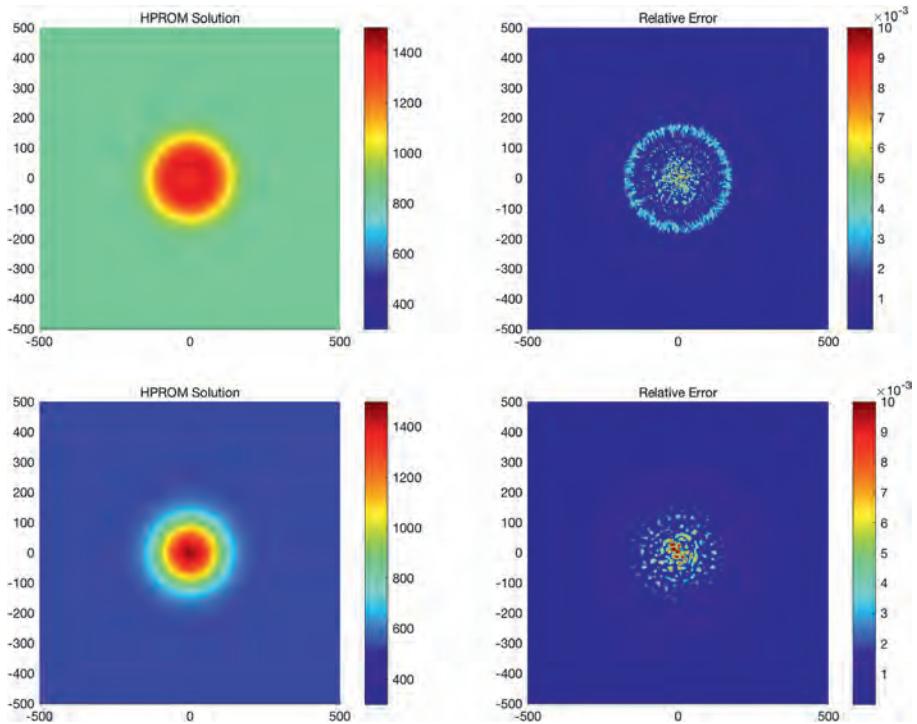
Figure 5.11: Parametric, two-way coupled, PDE-ODE wildfire problem: Temperature distributions at $t = 60$ s predicted using the HPROM (left) and corresponding relative errors (right), for $\boldsymbol{\mu} = (1.07, 91.19, 258.41, 10^{-4}, 0.0201)^T$ (top) and $\boldsymbol{\mu} = (1.5, 50, 65, 5.5 \times 10^{-5}, 0.15)^T$ (bottom).

5.5.3 Hyperreduction of nonlinear structural dynamics models

Two highly nonlinear structural dynamics problems are considered here. The first one focuses on a fast spinning top: It has the merit of being easily reproducible by the reader who is familiar with solid mechanics, and is explored to highlight the instability of the DEIM for second-order dynamical systems. The second problem focuses on an underbody blast event. Its associated FE model is representative of two families of computational models: those whose practical exploitation calls for model reduction, due to their CPU-intensive nature; and those that are difficult to reduce due to the presence of rotational DOFs. In both problems, the PROMs are constructed using the POD method based on displacement/rotation snapshots. In the first problem, the PROM is hyperreduced using both DEIM and ECSW. In the second problem, only ECSW is applied to hyperreduce the constructed PROM, due to its superior numerical stability properties. Because DEIM and ECSW operate essentially in the same fashion whether the PROM to be hyperreduced is parametric or nonparametric, both aforementioned problems are considered here in their simpler, nonparametric context (for the parametric case, the reader is referred to [31]). In all cases, ECSW is configured with the

convex approximation A1 (5.67). Furthermore, it is equipped with the NNLS algorithm developed by Lawson and Hanson [28] for solving the NNLS problem associated with this approximation because this algorithm has demonstrated an excellent track record of robustness and performance for hyperreduction [22, 24, 14].

Given a computed, time-dependent, PROM-based approximate solution $\mathbf{q}_{N_h'}(t) \approx \mathbb{V}'\mathbf{q}_{n'}(t)$, the corresponding global relative error in a direction $\diamond$ is defined here as

$$\mathbb{RE}_\diamond = \frac{\sqrt{\sum_{t\in\mathcal{P}}(\mathbf{q}_{\diamond_{N_h'}}(t) - \mathbb{V}'\mathbf{q}_{\diamond_{n'}}(t))^T(\mathbf{q}_{\diamond_{N_h'}}(t) - \mathbb{V}'\mathbf{q}_{\diamond_{n'}}(t))}}{\sqrt{\sum_{t\in\mathcal{P}}\mathbf{q}_{\diamond_{N_h'}}(t)^T\mathbf{q}_{\diamond_{N_h'}}(t)}} \times 100\,\%,$$

where the subscript $\diamond$ designates the displacement/rotation in the x-, y-, or z-direction of the global coordinate frame; $\mathbf{q}_{\diamond_{N_h'}}(t)$ is the vector of $\diamond$-displacement/rotation DOFs at time t extracted from the solution obtained using the discrete FOM of interest; $\mathbb{V}'\mathbf{q}_{\diamond_{n'}}(t)$ is the vector of $\diamond$-displacement/rotation DOFs at time t extracted from the solution obtained using the PROM or hyperreduced PROM whose performance is being assessed; and $\mathcal{P}$ is the set of timestamps used in the evaluation of $\mathbb{RE}_\diamond$ – that is,

$$\mathcal{P} = \{t \in \{t^0, t^0 + \Delta s, t^0 + 2\Delta s, \ldots\} : t \le T\},$$

where Δs denotes the sampling time interval chosen for the evaluation of the global relative error.

All computations reported herein are performed in double-precision arithmetic on a parallel Linux cluster where each computing node consists of two quad-core Intel Xeon E5345 processors running at 2.33 GHz inside a Dell Poweredge 1950 and has 16 GB of memory, and the interconnect is Cisco DDR InfiniBand.

5.5.3.1 Fast spinning top in a gravitational field

First, the hyperreduction of a geometrically nonlinear, FE-based PROM constructed for a fast spinning top in the presence of a gravitational field is considered. This hyperreduction problem was previously discussed in [22], [24], and [14]. The shape of the top is a cone with height $H = 0.1\,\text{m}$ and radius $R = 0.05\,\text{m}$. The top is assumed to be made of aluminum, which is modeled here using the Saint Venant–Kirchhoff constitutive law. This hyperelastic material law can be expressed as a linear relation between the second Piola–Kirchhoff stress tensor and the Green–Lagrange strain tensor. Hence, the internal force vector is in this case a nonlinear function $\mathbf{f}_{N_h}^{\text{int}}(\mathbf{q}_{N_h'}(t))$ of the displacement vector $\mathbf{q}_{N_h'}(t)$ due to the nonlinear kinematics (geometric nonlinearities). The Young modulus of this material is $E = 80\,\text{GPa}$, and its Poisson ratio is $\nu = 0.33$. Its density in the reference configuration is $\rho_o = 2,700\,\text{kg/m}^3$.

The top is simply supported at its apex. Initially, it is set into the position obtained by a rigid body rotation of $\frac{\pi}{3}$ rad about the x-axis, and into the spinning motion about its axis with the convected angular velocity $\Omega_z = 300$ rad/s. The gravitational acceleration, $g = 9.81$ m/s^2, generates an external body force that acts on the top in the negative z-direction.
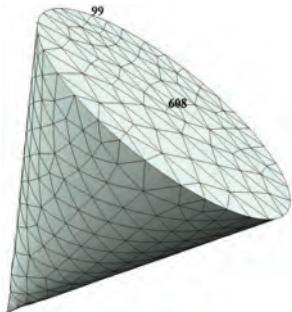
Figure 5.12: Spinning top: FE mesh and nodes 99 and 608 where the time histories of the x- and z-displacements are observed, respectively.

For this problem, an FE structural model is constructed using 4,461 10-noded tetrahedral elements with three displacement DOFs per node – which results in an FOM of dimension N_h = 13,317 DOFs. Figure 5.12 shows the constructed FE model. Time discretization is performed using the second-order accurate explicit central difference method. For this explicit time integrator and the aforementioned FOM, the maximum stability time step is $\Delta t = 1 \times 10^{-7}$ s. Using this time step, the FOM-based simulation of the first 1 second of top spinning consumes 21.8 hours of wall-clock time on 80 cores of the Linux cluster. The time-dependent solution computed during this simulation is sampled every $\Delta s = 2 \times 10^{-3}$ s for the purpose of constructing three different POD-based PROMs of dimension $n = 9$, $n = 10$, and $n = 15$. The DEIM and ECSW are applied for the hyperreduction of each of these PROMs as follows: ECSW is applied using values of the relative training tolerance τ that produce reduced meshes containing less than 1 % of the number of elements of the original FE mesh; and DEIM is applied with the set of indices $\mathcal{I}$ selected so that if one DOF is sampled at a node, then all DOFs attached to that node are also sampled – and therefore, $|\mathcal{I}| \geq m$, where m is the dimension of the ROB $\mathbb{U}$ (5.39).

All HPROM-based simulations of the dynamics of the spinning top are performed using the same explicit central difference time integrator used for computing the FOM-based time-dependent solution. However, because these simulations are not governed by the same Courant–Friedrichs–Lewy restriction as their counterpart FOM-based simulation, they can be performed – and therefore are performed – using a time step that is 20 times larger than the maximum-stability time step of that reference simulation. Furthermore, because all reduced meshes generated by DEIM and ECSW contain in this case less than 35 elements (for example, see Table 5.4 for the case of ECSW),

all HPROM-based simulations of the dynamics of the spinning top are executed on a single core of the Linux cluster.

It turns out that for this problem and chosen time integrator, all discrete HPROMs obtained using the DEIM are numerically unstable, independently of the chosen value for m; for example, see Figure 5.13, which reports the computed time histories of the x-displacement at node 99 of the FE mesh. On the other hand, all counterpart discrete HPROMs delivered by ECSW are found to be numerically stable, as anticipated by the theory presented in Section 5.4.2.2.3. Figure 5.14 – which reports the computed time histories of the x-displacement at node 99 and z-displacement at node 608 of the FE mesh – and Table 5.3 show that all discrete HPROMs constructed using ECSW deliver
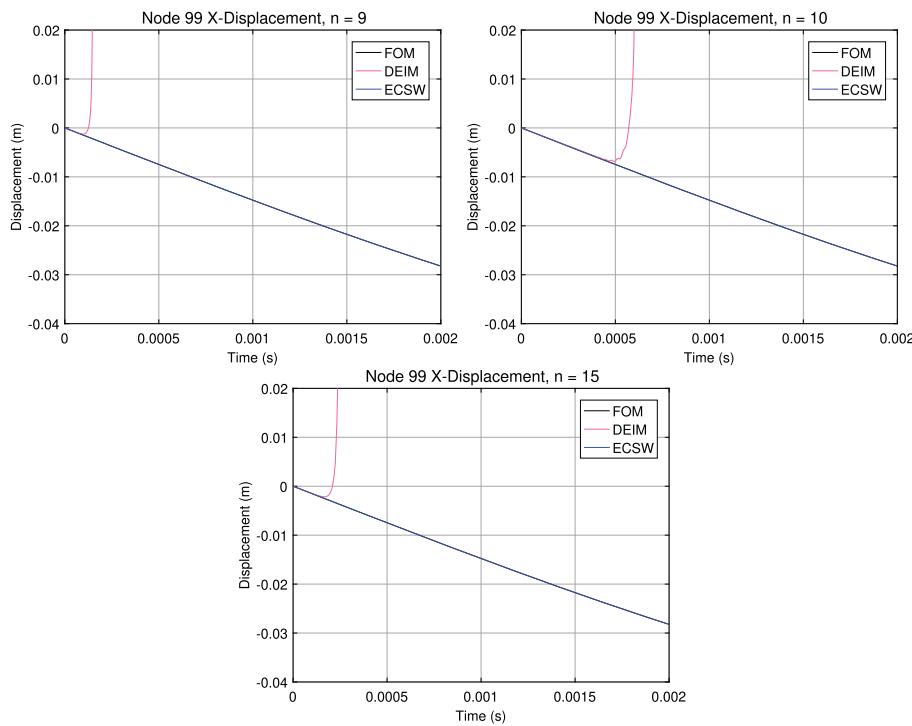
Figure 5.13: Fast spinning top: Numerical instabilities exhibited by the DEIM-based HPROMs.

Table 5.3: Fast spinning top: Global accuracy delivered by the ECSW-based HPROMs.

n	$\mathbb{RE}_x$ (%)	$\mathbb{RE}_y$ (%)	$\mathbb{RE}_z$ (%)	$\mathbb{RE}_{vel_x}$ (%)	$\mathbb{RE}_{vel_y}$ (%)	$\mathbb{RE}_{vel_z}$ (%)
9	0.012	0.0099	0.014	0.28	0.31	0.17
10	0.010	0.0083	0.012	0.24	0.26	0.16
15	0.0078	0.0064	0.0091	0.12	0.13	0.092
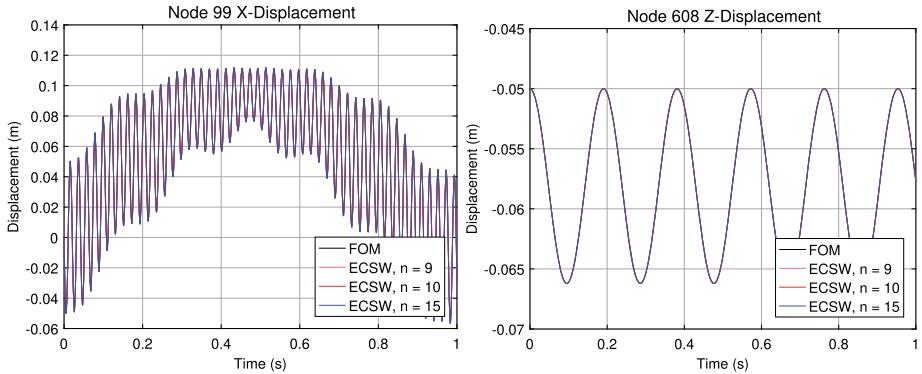
Figure 5.14: Fast spinning top: Numerical stability and local accuracy delivered by the ECSW-based HPROMs.

Table 5.4: Fast spinning top: Variations with the dimension n of the right ROB of the size of the reduced mesh generated by ECSW and the speedup factor delivered by the ECSW-based HPROM.

n	Reduced mesh (# of elements)	Speedup factor
9	12	9.25×10^3
10	15	7.66×10^3
15	34	3.21×10^3

excellent levels of local and global accuracy, respectively. Table 5.4 shows that they also deliver impressive speedups: partly because the underlying reduced meshes have very small sizes; and partly because these ECSW-based, explicit, discrete HPROMs afford a stability time step that is on average 20 times larger than the critical time step for the underlying discrete FOM.

5.5.3.2 Structural response of a vehicle frame to underbody blast

Next, the hyperreduction using ECSW of a highly nonlinear, POD-based PROM of dimension $n = 100$, constructed for the analysis of the structural response of a generic V-hull vehicle frame to a live fire blast test, is considered. The frame has a complex structure (Figure 5.15(a)) that is made predominantly of steel, which is modeled here as a nonlinear elastoplastic material. It is subjected to a gravity load, and to an external, configuration- and time-dependent pressure force due to the explosion of a 10 kg charge placed under its body, at the location shown in Figure 5.15(b). For this purpose, a structural, nonlinear, FE model of the vehicle frame is constructed using n_e = 236,995 flexible shell and rigid beam elements with six DOFs per node (Fig-
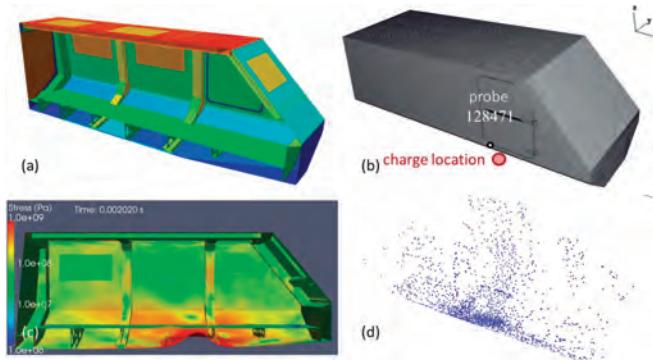
Figure 5.15: Underbody blast of a V-hull vehicle frame: (a) complex structure; (b) FE mesh, and locations of charge and a nodal probe; (c) snapshot of the structural response at $t = 0.002$ s; and (d) ECSW-generated reduced mesh.

ure 5.15(b)). This FOM features large rotations, large strains, large angular velocities – and therefore both geometrical and material nonlinearities – and a large dimension equal to $N_h = 1,399,056$ DOFs. The blast pressure loading is modeled using the CONWEP software [27].

The vehicle is assumed to be initially at rest. For the sake of diversification, time discretization is performed here using the implicit generalized-α method [17] with $\beta = 0.444$, $\gamma = 0.833$, $\alpha_f = 0.333$, $\alpha_m = 0$, and the constant time step $\Delta t = 1 \times 10^{-5}$ s. As for the previous problem, ECSW is configured with the convex approximation A1 (5.67), equipped with the NNLS algorithm [28] for solving the NNLS problem associated with this approximation, and applied to the hyperreduction of the constructed nonlinear PROM. Specifically, the relative training tolerance is set to $\tau = 0.01$. In this case, the NNLS algorithm delivers the reduced mesh $\tilde{\mathcal{E}}$ with 3,145 elements – that is, with about 1.32 % of the number of elements of the original FE mesh – as shown in Figure 5.15(d). Then, the structural response of the vehicle frame is computed three times in the time interval $[0, 10^{-2}]$ s: using the discrete FOM of dimension $N_h = 1,399,056$, the discrete PROM of dimension $n = 100$, and the ECSW-generated discrete HPROM. All three simulations are performed on a single core of the Linux cluster, using the same fixed time step $\Delta t = 1 \times 10^{-5}$ s. For the PROM- and HPROM-based predictions, the relative global errors are calculated using the same sampling time interval $\Delta s = 1 \times 10^{-5}$ s. A snapshot of the structural response of the structural system computed using the discrete FOM is shown in Figure 5.15(c).

Tables 5.5 and 5.6 report the wall-clock timings, speedup factors, and global relative errors of the PROM- and HPROM-based simulations. These results show that overall, the ECSW-based HPROM maintains the level of accuracy of its underlying PROM. This level of accuracy is high for the predicted displacement and rotation fields, reasonable for the computed velocity field, but low for the predicted angular velocity

Table 5.5: Structural response of a vehicle frame to underbody blast: Wall-clock timings on a single core and speedup factors.

	Wall-clock time (s)	Speedup factor
FOM	5.02×10^5	1.0
PROM	3.63×10^5	1.38
HPROM	3.57×10^3	141

Table 5.6: Structural response of a vehicle frame to underbody blast: Global relative errors.

	PROM	HPROM
$\mathbb{RE}_x$ (%)	3.05	3.53
$\mathbb{RE}_y$ (%)	2.46	2.72
$\mathbb{RE}_z$ (%)	3.52	3.69
$\mathbb{RE}_{rot_x}$ (%)	1.53	1.87
$\mathbb{RE}_{rot_x}$ (%)	1.63	2.04
$\mathbb{RE}_{rot_x}$ (%)	2.40	2.95
$\mathbb{RE}_{vel_x}$ (%)	12.05	13.62
$\mathbb{RE}_{vel_y}$ (%)	7.21	8.38
$\mathbb{RE}_{vel_z}$ (%)	4.81	5.70
$\mathbb{RE}_{vel_{rot_x}}$ (%)	23.03	24.89
$\mathbb{RE}_{vel_{rot_y}}$ (%)	24.60	26.63
$\mathbb{RE}_{vel_{rot_z}}$ (%)	29.92	32.40

field. The latter can be improved by including velocity snapshots in the snapshot matrix underlying the construction of the POD-based PROM. The reader can observe that as expected, the discrete PROM does not speed up the solution time of the discrete FOM by any meaningful factor. On the other hand, the HPROM reduces the solution time of the FOM by more that two orders of magnitude. While significant, this speedup factor is smaller than that achievable in the context of explicit discretizations. The reason is that like in the case of an implicit discrete FOM, the time step of an implicit discrete PROM or HPROM is limited by accuracy and not by stability considerations, and therefore cannot exceed that of the underlying discrete FOM. In other words, the speedup factor of 141 achieved in this case is purely due to model reduction and hyperreduction.

Figures 5.16 and 5.17 display the computed time histories of selected displacement/rotation and velocity/angular velocity DOFs, respectively, computed at a probe located at a node of the FE mesh in the vicinity of the explosive charge (Figure 5.15(b)). The accuracy levels demonstrated in these two figures for the constructed PROM and HPROM are consistent with the quantitative accuracy results summarized in Table 5.6.
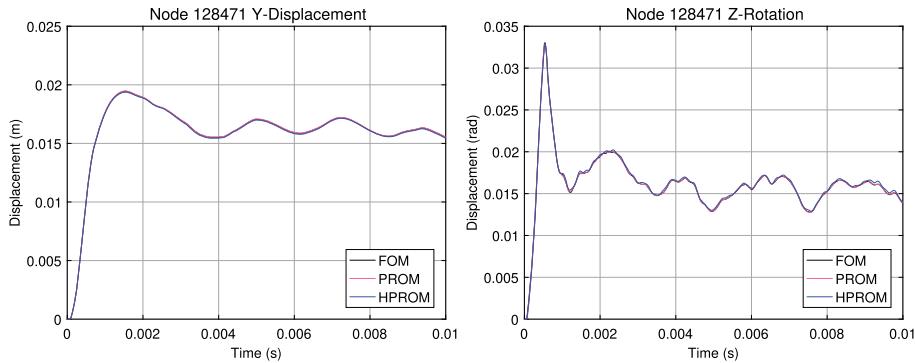
Figure 5.16: Structural response of a vehicle frame to underbody blast: Sample displacement and rotation time histories computed at a probe.
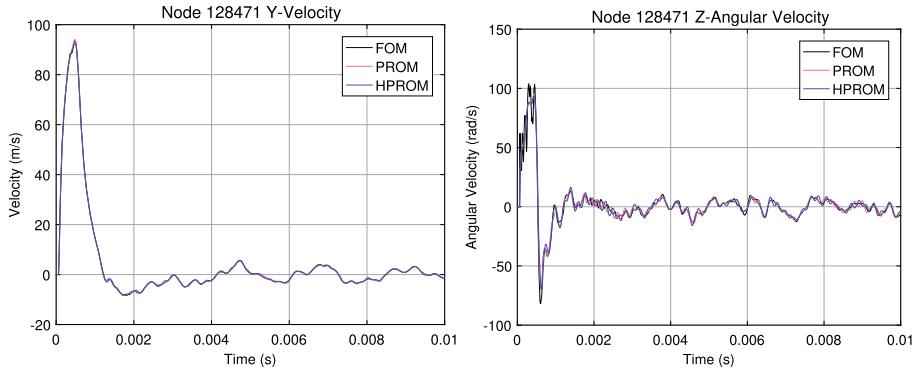
Figure 5.17: Structural response of a vehicle frame to underbody blast: Sample velocity and angular velocity time histories computed at a probe.

5.6 Summary and conclusions

For many linear and nonlinear problems, PMOR is not guaranteed to accelerate the performance of N_h-dimensional FOMs, even when the resulting PROM has a dimension $n \ll N_h$. For linear problems, this is the case for large-scale, parametric FOMs, where the computational complexity of the projections of the linear operators and source term defining the FOM – which must be repeated every time a parameter is changed – scales as $\mathcal{O}(N_h^2 n)$ and $\mathcal{O}(N_h n)$, respectively. Therefore, these repeated projections can rapidly become overwhelming. For nonlinear problems, this issue is even more severe as it arises for both parametric and nonparametric problems.

In this chapter, two divide-and-conquer approaches for addressing the computational bottlenecks outlined above have been reviewed.

The first approach is feasible and exact for: parametric, linear FOMs that admit an efficient parameter-affine representation, where efficiency in this context has been properly defined in this chapter; and parametric and nonparametric, nonlinear FOMs characterized by a low-order polynomial dependency of the internal force vector on the solution and a time-independent external force vector. This first approach consists in dividing the computation of the reduced quantities, whenever possible, into two parts: one part that is responsible for the computational bottlenecks mentioned above and can be addressed by offline precomputations; and another part that is amenable to real-time processing.

The second approach consists of a family of inexact approaches that introduce into the construction of a PROM an additional layer of approximations that enable the real-time evaluation of all of its reduced quantities. In this chapter, two methodologies belonging to this family of approaches have been discussed. The first one targets parametric, linear PROMs. It samples offline the parameter space of interest at a carefully chosen set of parameter points, and constructs at each sampled point an accurate linear PROM. Then, at each queried but unsampled parameter point, it constructs in real-time a linear PROM by interpolating on matrix manifolds the operators defining the precomputed PROMs. The second inexact methodology discussed in this chapter is known as hyperreduction. It is more comprehensive than the first one, as it is equally applicable to parametric and nonparametric, linear and nonlinear PROMs. This methodology comes at least in two flavors: approximate-then-project and project-then-approximate methods. At the time of writing this chapter, the DEIM represents the state of the art of approximate-then-project methods, and ECSW represents that of project-then-approximate methods. For second-order dynamical systems such as those arising in wave propagation, solid mechanics, and structural dynamics applications, ECSW is to date the only known hyperreduction method with provable structure-preserving and unconditional stability properties. As such, it is superior to DEIM for this important class of problems. For other dynamical systems, DEIM and ECSW typically exhibit comparable performances in terms of accuracy and computational efficiency. For such systems, they have been shown in the literature, and are shown in this chapter, to be capable of speeding up the execution time of FOMs by factors that are problem-dependent, but typically range between one and three orders of magnitude if not higher. These hyperreduction methods are robust, practical, and a must for the reduction of: parametric, linear FOMs that do not admit an efficient parameter-affine representation; parametric and nonparametric, nonlinear FOMs that are not characterized by a low-order polynomial nonlinearity in the internal force vector; and parametric, linear or nonlinear FOMs where the parameter vector may vary within a single simulation.

Bibliography

[1]   D. Amsallem, J. Cortial, and C. Farhat, Toward real-time computational-fluid-dynamics-based aeroelastic computations using a database of reduced-order information, *AIAA Journal*, **48** (9) (2010), 2029–2037.

[2]   D. Amsallem and C. Farhat, Interpolation method for adapting reduced-order models and application to aeroelasticity, *AIAA Journal*, **46** (7) (2008), 1803–1813.

[3]   D. Amsallem and C. Farhat, An online method for interpolating linear parametric reduced-order models, *SIAM Journal on Scientific Computing*, **33** (5) (2011), 2169–2198.

[4]   D. Amsallem and C. Farhat, Stabilization of projection-based reduced-order models, *International Journal for Numerical Methods in Engineering*, **91** (4) (2012), 358–377.

[5]   D. Amsallem, R. Tezaur, and C. Farhat, Real-time solution of linear computational problems using databases of parametric reduced-order models with arbitrary underlying meshes, *Journal of Computational Physics*, **326** (2016), 373–397.

[6]   S. An, T. Kim, and D. James, Optimizing cubature for efficient integration of subspace deformations, *ACM Transactions on Graphics*, **27** (5) (2008), 165.

[7]   P. Astrid, S. Weiland, K. Willcox, and T. Backx, Missing point estimation in models described by proper orthogonal decomposition, *IEEE Transactions on Automatic Control*, **53** (10) (2008), 2237–2251.

[8]   J. Barbič and D. L. James, Real-time subspace integration for st. venant-kirchhoff deformable models, *ACM Transactions on Graphics*, **24** (3) (2005), 982–990.

[9]   M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, *Comptes Rendus. Mathématique*, **339** (9) (2004), 667–672.

[10] M. Bebendorf, Y. Maday, and B. Stamm, Comparison of some reduced representation approximations, in *Reduced Order Methods for Modeling and Computational Reduction*, pp. 67–100, Springer, 2014.

[11] C. Canuto, T. Tonn, and K. Urban, A posteriori error analysis of the reduced basis method for non-affine parameterized nonlinear PDEs, *SIAM Journal on Numerical Analysis*, **47** (3) (2009), 2001–2022.

[12] K. Carlberg, C. Bou-Mosleh, and C. Farhat, Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations, *International Journal for Numerical Methods in Engineering*, **86** (2) (2011), 155–181.

[13] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem, The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows, *Journal of Computational Physics*, **242** (2013), 623–647.

[14] T. Chapman, P. Avery, P. Collins, and C. Farhat, Accelerated mesh sampling for the hyper reduction of nonlinear computational models, *International Journal for Numerical Methods in Engineering*, **109** (12) (2017), 1623–1654.

[15] S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM Journal on Scientific Computing*, **32** (5) (2010), 2737–2764.

[16] S. S. Chen, D. L. Donoho, and M. A. Saunders, Atomic decomposition by basis pursuit, *SIAM Journal on Scientific Computing*, **20** (1) (1998), 33–61.

[17] J. Chung and G. Hulbert, A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method, *Journal of Applied Mechanics*, **60** (2) (1993), 371–375.

[18] D. Colton and R. Kress, *Inverse Acoustic and Electromagnetic Scattering Theory*, Springer-Verlag, Berlin, 2012.

[19] J. L. Eftang, M. A. Grepl, and A. T. Patera, A posteriori error bounds for the empirical interpolation method, *Comptes Rendus de L'Académie des Sciences. Series 1, Mathematics*, **348** (9–10) (2010), 575–579.

[20] R. Everson and L. Sirovich, Karhunen-Loeve procedure for gappy data, *Journal of the Optical Society of America. A, Online*, **12** (8) (1995), 1657–1664.

[21] C. Farhat, P. Avery, T. Chapman, and J. Cortial, Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency, *International Journal for Numerical Methods in Engineering*, **98** (9) (2014), 625–662.

[22] C. Farhat, P. Avery, T. Chapman, and J. Cortial, Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency, *International Journal for Numerical Methods in Engineering*, **98** (9) (2014), 625–662.

[23] C. Farhat, A. Bos, R. Tezaur, T. Chapman, P. Avery, and C. Soize, A stochastic projection-based hyperreduced order model for model-form uncertainties in vibration analysis, in *2018 AIAA Non-Deterministic Approaches Conference*. Kissimmee, Florida, January 8–12, 2018.

[24] C. Farhat, T. Chapman, and P. Avery, Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models, *International Journal for Numerical Methods in Engineering*, **102** (2015), 1077–1110.

[25] M. A. Grepl, Certified reduced basis methods for nonaffine linear time-varying and nonlinear parabolic partial differential equations, *Mathematical Models and Methods in Applied Sciences*, **22** (3) (2012), 1793-6314.

[26] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, *ESAIM: Mathematical Modelling and Numerical Analysis*, **41** (3) (2007), 575–605.

[27] D. W. Hyde, *Conventional Weapons Program (ConWep)*. U.S Army Waterways Experimental Station, Vicksburg, MS, USA, 1991.

[28] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, vol. 161, SIAM, 1974.

[29] Y. Maday, N. C. Nguyen, A. T. Patera, and G. S. H. Pau, A general multipurpose interpolation procedure: the magic points, *Communications on Pure and Applied Analysis*, **8** (1) (2009), 383–404.

[30] J. Mandel, L. S. Bennethum, J. D. Beezley, J. L. Coen, C. C. Douglas, M. Kim, and A. Vodacek, A wildland fire model with data assimilation, *Mathematics and Computers in Simulation*, **79** (3) (2008), 584–606.

[31] A. Paul-Dubois-Taine and D. Amsallem, An adaptive and efficient greedy procedure for the optimal training of parametric reduced-order models, *International Journal for Numerical Methods in Engineering*, **102** (5) (2015), 1262–1292.

[32] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, vol. 92, Springer, 2016.

[33] D. Ryckelynck, A priori hyperreduction method: an adaptive approach, *Journal of Computational Physics*, **202** (1) (2005), 346–366.

[34] R. Schaback, A practical guide to radial basis functions. *Electronic Resource*, **11** (2007).

[35] L. Sirovich, Turbulence and the dynamics of coherent structures. i. coherent structures, *Quarterly of Applied Mathematics*, **45** (3) (1987), 561–571.

[36] C. Soize and C. Farhat, A nonparametric probabilistic approach for quantifying uncertainties in low-dimensional and high-dimensional nonlinear models, *International Journal for Numerical Methods in Engineering*, **109** (6) (2017), 837–888.

[37] K. Stein, T. E. Tezduyar, and R. Benney, Automatic mesh update with the solid-extension mesh moving technique, *Computer Methods in Applied Mechanics and Engineering*, **193** (21) (2004), 2019–2032.

[38] P. Tiso and D. Rixen, Discrete empirical interpolation method for finite element structural dynamics, *Topics in Nonlinear Dynamics*, **1** (2013), 203–212.

[39] R. O. Weber, G. N. Mercer, H. S. Sidhu, and B. F. Gray, Combustion waves for gases (Le = 1) and solids (Le → ∞). *Proceedings of the Royal Society A. Mathematical, Physical and Engineering Sciences*, **453** (1960) (1997), 1105–1118.

[40] K. Willcox and J. Peraire, Balanced model reduction via the proper orthogonal decomposition, *AIAA Journal*, **40** (11) (2002), 2323–2330.

[41] D. Wirtz, D. C. Sorensen, and B. Haasdonk, A posteriori error estimation for DEIM reduced nonlinear dynamical systems, *SIAM Journal on Scientific Computing*, **36** (2) (2014), A311–A338.

Andreas Buhr, Laura Iapichino, Mario Ohlberger, Stephan Rave,
Felix Schindler, and Kathrin Smetana

6 Localized model reduction for parameterized problems

Abstract: In this contribution we present a survey of concepts in localized model order reduction methods for parameterized partial differential equations. The key concept of localized model order reduction is to construct local reduced spaces that have only support on part of the domain and compute a global approximation by a suitable coupling of the local spaces. In detail, we show how optimal local approximation spaces can be constructed and approximated by random sampling. An overview of possible conforming and nonconforming couplings of the local spaces is provided and corresponding localized a posteriori error estimates are derived. We introduce concepts of local basis enrichment, which includes a discussion of adaptivity. Implementational aspects of localized model reduction methods are addressed. Finally, we illustrate the presented concepts for multiscale, linear elasticity, and fluid-flow problems, providing several numerical experiments.

Keywords: localized model reduction, reduced basis method, randomized training, a posteriori error estimation, basis enrichment, online adaptivity, parameterized systems, multiscale problems

MSC 2010: 65Y15, 65N30, 65N55, 65N15, 35J20, 35J25

6.1 Introduction

Projection-based model order reduction has become a mature technique for simulations of large classes of parameterized systems; for an introduction, we refer to the text books and survey [14, 50, 97, 15] and to Chapters 1 to 4 of this volume of *Model order reduction*. However, especially for large-scale and multiscale problems the "stan-

Andreas Buhr, Mario Ohlberger, Stephan Rave, Felix Schindler, Mathematics Münster, Einsteinstr. 62, D-48149 Münster, Germany
Laura Iapichino, Department of Mathematics and Computer Science, TU Eindhoven, Eindhoven, The Netherlands
Kathrin Smetana, University of Twente, Faculty of Electrical Engineering, Mathematics & Computer Science, Zilverling, P.O. Box 217, 7500 AE Enschede, The Netherlands

dard" model order reduction approach as described for instance in Chapter 1 of this volume of *Model order reduction* exhibits several limitations: curse of parameter dimensionality in the sense that many parameters require prohibitively large reduced spaces, no topological flexibility, and possibly high computational costs and storage requirements in the offline stage for instance due to large computational domains. Localized model order reduction methods, which combine approaches from model order reduction, multiscale methods, and/or domain decomposition techniques, overcome or significantly mitigate those limitations. As a further advantage, they allow using reduced spaces of different dimensions in different parts of the computational domain and accommodate (local) changes of the geometry and the partial differential equation (PDE) in the online stage. The key idea of localized model order reduction is to construct local reduced spaces on (unions of) subdomains of the decomposed computational domain and couple the local reduced spaces across interfaces either in a conforming or in a nonconforming manner. In this chapter we investigate localized model order reduction for linear coercive elliptic parameterized problems; inf-sup stable problems have for instance been considered in [60] and parabolic and nonlinear problems will be briefly discussed at the end of this chapter.

We discuss both conforming and nonconforming localized approximations. Prominent examples for a conforming localization for nonparametric PDEs are the partition of unity method [11], the generalized finite element method (GFEM) [8, 7, 11, 10], and component mode synthesis (CMS) [59, 12, 17], [52, 64].

A combination of domain decomposition and reduced basis methods has first been considered in the reduced basis element method (RBEM) [76, 77, 74], where the local reduced basis approximations are coupled by Lagrange multipliers in a nonconforming manner. The reduced basis hybrid method [62] extends the RBEM by additionally considering a coarse finite element (FE) discretization on the whole domain to account for continuity of normal stresses in the context of Stokes equations. Alternatively, a nonconforming coupling can be realized, say, by penalization as in the local reduced basis discontinuous Galerkin approach [66], the localized reduced basis multiscale (LRBMS) method [5, 90, 87], the discontinuous Galerkin (DG) RBEM [6], or the generalized multiscale discontinuous Galerkin method [29]. The static condensation reduced basis element (scRBE) method [61, 60, 37, 103] combines intra-element reduced basis approximations similar to the RBEM with coupling techniques from CMS, resulting in a conforming approximation. A similar approach is pursued by the ArbiLoMod [20] that also allows for arbitrary (nonparametric) local changes of the underlying equations and/or the geometry.

In the context of the proper generalized decomposition method (Chapter 3 of this volume of *Model order reduction*) a domain decomposition strategy has been proposed in [55], and in [94] hierarchical model reduction [112, 94, 92, 102] has been combined with an iterative substructuring method.

Concerning the generation of local approximation spaces we focus on empirical training (see for instance [37, 10, 103]), i. e., local reduced spaces generated from lo-

cal solutions of the PDE, and adaptive basis enrichment. In detail, we present local approximation spaces that are optimal in the sense of Kolmogorov and can be constructed by solving a local so-called transfer eigenvalue problem on the space of local solutions of the PDE. Optimal local approximation spaces for subdomains have first been proposed in [10] and for interfaces and parameterized PDEs in [103]. We will also show how those optimal approximation spaces can be approximated by random sampling [21]. A localizable a posteriori error estimator is crucial for an adaptive enrichment of the local reduced spaces where the reduced approximation is not accurate enough. Such an adaptive basis enrichment is one way to approach "optimal" computational complexity within outer loop applications such as optimal control, inverse problems, or Monte Carlo methods. With this respect, we will also present a framework for localized residual-based error control [20, 100] as well as localized a posteriori error estimation based on flux reconstruction [39, 90].

Naturally, the presented methods for localized model reduction share a lot of features with domain decomposition techniques and multiscale methods. We particularly refer to domain decomposition and preconditioning techniques with multiscale coarse spaces such as [1, 45, 42] or the more recent contributions [106, 43, 47]. In the context of the FETI-DP iterative substructuring method we refer to [82, 67]. For multiscale problems there has been a tremendous development of suitable numerical methods in the last two decades, including the multiscale FE method (MsFEM) [54, 35, 36, 49], the heterogeneous multiscale method [114, 115, 85, 2], the variational multiscale method [56, 58, 72], or the more recent local orthogonal decomposition [81, 48]. Model reduction can be used to accelerate the solution of localized problems which occur in multiscale methods; see, e. g., [3, 4]. Similar to the methods presented in this chapter the generalized MsFEM (GMsFEM) [34, 31, 30] relies on a Galerkin projection on local subspaces, but in contrast uses ideas from multiscale methods to construct the local bases. A connection between multiscale methods and domain decomposition has recently been investigated in [69–71].

This chapter is organized as follows. In Section 6.2 we introduce the problem setting and basic notation for localized model order reduction of coercive variational problems. Concepts for conforming and nonconforming coupling of approximation spaces are presented in Section 6.3. Section 6.4 deals with the preparation of local approximation spaces. Particularly, the construction of optimal local approximation spaces and their approximation via random sampling is presented and illustrated with numerical experiments for linear elasticity. In Section 6.5 we present two abstract frameworks for localized a posteriori error estimation and give exemplifications for conforming and nonconforming localized model reduction approaches. Localized a posteriori error estimators are the key ingredient for basis enrichment strategies and online adaptivity, which are presented in Section 6.6. Computational aspects are discussed in Section 6.7 and numerical experiments for multiscale problems and fluid flow are presented in Section 6.8. We conclude by showing possible extensions to parabolic and nonlinear problems in Section 6.9.

6.2 Parameterized partial differential equations and localization

Let $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, be a large, bounded domain with Lipschitz boundary. Let us further introduce a Hilbert space V such that $[H_0^1(\Omega)]^z \subset V \subset [H^1(\Omega)]^z$, $z = 1, \ldots, d$, and denote by V' the dual space of V. Moreover, we introduce the compact set of admissible parameters $\mathcal{P} \subset \mathbb{R}^p$, $p \in \mathbb{N}$. We consider the following variational problem.

Definition 6.1 (Parameterized coercive problem in variational form). For any parameter $\boldsymbol{\mu} \in \mathcal{P}$ find $u(\boldsymbol{\mu}) \in V$, such that

$$a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \text{for all } v \in V. \tag{6.1}$$

Here, $f(\cdot; \boldsymbol{\mu}) \in V'$ and $a(\cdot, \cdot; \boldsymbol{\mu}) : V \times V \to \mathbb{R}$ denote parametric linear and bilinear forms, the latter being continuous and coercive with respect to the norm $\|\cdot\|_V$ induced by the inner product $(\cdot, \cdot)_V : V \times V \to \mathbb{R}$. That is, there exist constants $0 < \alpha \leq \alpha(\boldsymbol{\mu}) \leq \gamma(\boldsymbol{\mu}) \leq \gamma$, such that for any $\boldsymbol{\mu} \in \mathcal{P}$,

$$a(v, w; \boldsymbol{\mu}) \leq \gamma(\boldsymbol{\mu}) \, \|v\|_V \, \|w\|_V \quad \text{for all } v, w \in V,$$
$$a(v, v; \boldsymbol{\mu}) \geq \alpha(\boldsymbol{\mu}) \, \|v\|_V^2 \qquad \text{for all } v \in V.$$

Let us denote the energy norm of u for a parameter $\boldsymbol{\mu}$ as $\|u\|_{\boldsymbol{\mu}} := a(u, u; \boldsymbol{\mu})^{1/2}$. Problem (6.1) thus admits a unique solution for all $\boldsymbol{\mu} \in \mathcal{P}$ owing to the Lax–Milgram theorem. Examples for (6.1) include elliptic multiscale problems, incompressible fluid flow, and linear elasticity, as detailed in the following. We will consider Neumann boundary conditions on Γ_N and Dirichlet boundary conditions on Γ_D, where Γ_N, Γ_D are nonoverlapping and $\Gamma_N \cup \Gamma_D = \partial\Omega$. To simplify notations, homogenous boundary conditions on $\partial\Omega$ will be prescribed in most places.

Example 6.2 (Parametric elliptic multiscale problems). With $V = H_0^1(\Omega)$, the pressure equation in the context of two-phase flow in porous media (obtained through Darcy's law) reads as follows: Given a collection of sources and sinks $q \in L^2(\Omega)$ and a parametric and possibly highly heterogeneous permeability field $\kappa : \mathcal{P} \to L^\infty(\Omega)^{d \times d}$, find for each $\boldsymbol{\mu} \in \mathcal{P}$ the global pressure $u(\boldsymbol{\mu}) \in V$, such that

$$-\nabla \cdot (\kappa(\boldsymbol{\mu}) \nabla u(\boldsymbol{\mu})) = q \quad \text{in a weak sense in } V'. \tag{6.2}$$

If the smallest eigenvalue of $\kappa(\boldsymbol{\mu})$ is bounded from below away from zero for all $\boldsymbol{\mu} \in \mathcal{P}$, we can consider this to be an example of Definition 6.1 by setting $a(u, v; \boldsymbol{\mu}) := \int_\Omega (\kappa(\boldsymbol{\mu}) \nabla u) \cdot \nabla v \, dx$ and $f(v; \boldsymbol{\mu}) := \int_\Omega qv \, dx$. In the context of instationary two-phase flow, (6.2) needs to be solved in each time step for varying total mobilities (modeled by the parametric nature of κ), while the permeability field κ typically resolves fine geological structures and thus requires a very fine computational grid compared to the size of Ω (see [90] and the references therein).

Example 6.3 (Incompressible fluid flow). The Stokes and Navier–Stokes equations represent a model of the flow motion for a viscous Newtonian incompressible fluid. In the steady case it can be formulated as follows:

$$
\begin{cases}
-\nu\Delta\mathbf{y} + \delta(\mathbf{y}\cdot\nabla)\mathbf{y} + \nabla p = \mathbf{f} & \text{in } \Omega, \\
\nabla\cdot\mathbf{y} = 0 & \text{in } \Omega, \\
\mathbf{y} = \mathbf{g}_D & \text{on } \Gamma_D, \\
-p\mathbf{n} + \nu\dfrac{\partial\mathbf{y}}{\partial\mathbf{n}} = \mathbf{g}_N & \text{on } \Gamma_N,
\end{cases}
\tag{6.3}
$$

where $(\mathbf{y}, p)$ are the velocity and the pressure fields defined on the computational domain Ω. The first equation expresses the linear momentum conservation and the second one the mass conservation, which is also called the continuity equation. Here $\mathbf{f}$ denotes a forcing term per unit mass, and $\mathbf{g}_D$ and $\mathbf{g}_N$ are the functions addressing the Dirichlet and Neumann boundary conditions respectively on Γ_D and Γ_N. The parameter $\nu = \sigma/\rho$ denotes the kinematic viscosity, with ρ being the density and σ the viscosity of the fluid. Navier–Stokes equations correspond to the case $\delta = 1$; here we consider only $\delta = 0$, the convective term is neglected, obtaining the steady Stokes equations, which provide a model in the case of slow motion of fluids with very high viscosity.

We denote the functional spaces for velocity and pressure fields by $X = (H^1_{0,\Gamma_D}(\Omega))^d$, $Q = L^2(\Omega)$, respectively, where $H^1_{0,\Gamma_D}(\Omega) = \{y \in H^1(\Omega) : y|_{\Gamma_D} = 0\}$. Moreover, for simplicity, we assume that $\mathbf{g}_D = 0$ (otherwise the lift function is required). The corresponding weak form of the Stokes equations (6.3) reads as follows: Find $(\mathbf{y}, p) \in X \times Q$ such that

$$
\nu\int_\Omega \nabla\mathbf{y} : \nabla\mathbf{w}\, d\Omega - \int_\Omega p\,\nabla\cdot\mathbf{w}\, d\Omega = \int_\Omega \mathbf{f}\cdot\mathbf{w}\, d\Omega + \int_{\Gamma_N} \mathbf{g}_N\cdot\mathbf{w}\, d\Gamma, \quad \forall\mathbf{w} \in X,
$$

$$
\int_\Omega q\,\nabla\cdot\mathbf{y}\, d\Omega = 0, \quad \forall q \in Q.
\tag{6.4}
$$

In a parameterized setting, the input-parameter vector $\boldsymbol{\mu}$ may characterize either the geometrical configuration or physical properties, boundary data, and sources of the problem.

Denoting by V the product space given by $V = X \times Q$, defining by $u(\boldsymbol{\mu}) = (\mathbf{y}(\boldsymbol{\mu}), p(\boldsymbol{\mu})) \in V$, and defining $v = (\mathbf{w}, q)$, the parameterized abstract formulation (6.4) can be rewritten in the following form: Find $u(\boldsymbol{\mu}) = (\mathbf{y}(\boldsymbol{\mu}), p(\boldsymbol{\mu})) \in V$ s.t.

$$
a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}), \quad \forall v \in V,
\tag{6.5}
$$

where

$$
a(u, v; \boldsymbol{\mu}) = \nu\int_\Omega \nabla\mathbf{y} : \nabla\mathbf{w}\, d\Omega - \int_\Omega p\,\nabla\cdot\mathbf{w}\, d\Omega - \int_\Omega q\,\nabla\cdot\mathbf{y}\, d\Omega,
\tag{6.6}
$$

$$
f(v; \boldsymbol{\mu}) = \int_\Omega \mathbf{f}\cdot\mathbf{w}\, d\Omega + \int_{\Gamma_N} \mathbf{g}_N\cdot\mathbf{w}\, d\Gamma.
\tag{6.7}
$$

Example 6.4 (Linear elasticity). We assume that $\Omega \subset \mathbb{R}^3$ represents an isotropic homogeneous material and we consider the following linear elastic boundary value problem: Find the displacement vector $u(\boldsymbol{\mu})$ and the Cauchy stress tensor $\boldsymbol{\sigma}(u(\boldsymbol{\mu}))$ such that

$$
\begin{aligned}
-\nabla \cdot \boldsymbol{\sigma}(u(\boldsymbol{\mu})) &= \mathbf{G}(\boldsymbol{\mu}) && \text{in } \Omega, \\
\boldsymbol{\sigma}(u(\boldsymbol{\mu})) \cdot \boldsymbol{n} &= 0 && \text{on } \Gamma_N, \\
u(\boldsymbol{\mu}) &= \boldsymbol{g}_D && \text{on } \Gamma_D,
\end{aligned}
\tag{6.8}
$$

where the body force $\boldsymbol{G} : \mathcal{P} \to \mathbb{R}^3$ accounts for gravity. We can express for a linear elastic material the Cauchy stress tensor as $\boldsymbol{\sigma}(u(\boldsymbol{\mu})) = E(\boldsymbol{\mu})\,\boldsymbol{C} : \boldsymbol{\varepsilon}(u(\boldsymbol{\mu}))$, where $\boldsymbol{C}$ is the fourth-order stiffness tensor, $\boldsymbol{\varepsilon}(u(\boldsymbol{\mu})) = 0.5(\nabla u(\boldsymbol{\mu}) + (\nabla u(\boldsymbol{\mu}))^T)$ is the infinitesimal strain tensor, and the colon operator : is defined as $\boldsymbol{C} : \boldsymbol{\varepsilon}(u(\boldsymbol{\mu})) = \sum_{k,l=1}^{3} \boldsymbol{C}_{ijkl}\boldsymbol{\varepsilon}_{kl}(u(\boldsymbol{\mu}))$. Moreover, $E : \mathcal{P} \to L^{\infty}(\Omega)$ denotes Young's modulus, which is assumed to be piecewise constant on Ω and satisfy $E(\boldsymbol{\mu}) \geq E_0 > 0$ for a constant $E_0 \in \mathbb{R}^+$. Therefore, the stiffness tensor can be written as

$$
\boldsymbol{C}_{ijkl} = \frac{\nu}{(1+\nu)(1-2\nu)}\delta_{ij}\delta_{kl} + \frac{1}{2(1+\nu)}(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}), \quad 1 \leq i,j,k,l \leq 3,
$$

where δ_{ij} denotes the Kronecker delta; we choose Poisson's ratio $\nu = 0.3$. The corresponding variational formulation of (6.8) then reads as follows: For any $\boldsymbol{\mu} \in \mathcal{P}$ find $u(\boldsymbol{\mu}) \in V = \{v \in [H^1(\Omega)]^3 \; : \; v = 0 \text{ on } \Gamma_D\}$ such that

$$
a(u(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \forall v \in V.
\tag{6.9}
$$

Here, the bilinear and linear forms $a(\cdot, \cdot; \boldsymbol{\mu}) : [H^1(\Omega)]^3 \times [H^1(\Omega)]^3 \to \mathbb{R}$ and $f(\cdot; \boldsymbol{\mu}) : [H^1(\Omega)]^3 \to \mathbb{R}$ are defined as

$$
a(w, v; \boldsymbol{\mu}) := \int_{\Omega} E(\boldsymbol{\mu})\,\frac{\partial w^i}{\partial x_j}\boldsymbol{C}_{ijkl}\frac{\partial v^k}{\partial x_l} \quad \text{and} \quad f(v; \boldsymbol{\mu}) := \int_{\Omega} \boldsymbol{G}(\boldsymbol{\mu}) \cdot v - a(\widehat{\boldsymbol{G}}(\boldsymbol{\mu}), v; \boldsymbol{\mu}),
$$

where $\widehat{\boldsymbol{G}}(\boldsymbol{\mu}) \in [H^1(\Omega)]^3$ denotes a suitable lifting function of the possibly nonhomogeneous Dirichlet boundary conditions.

To obtain approximate solutions of (6.1) we presume we have an appropriate grid-based numerical method at hand (the full-order model [FOM]), yielding a high- (but finite-)dimensional approximation space V_h. We consider conforming continuous Galerkin (CG) FEs, where $V_h \subset V$, and nonconforming discontinuous Galerkin or finite volume (FV) schemes, where $V_h \not\subset V$ (in which case we require broken Sobolev spaces for our analysis, see Section 6.3.2.2). As a starting point for localized model reduction we require the FOM space to be decomposable into "local" spaces, which we will make more precise shortly. While a localizing space decomposition could in general stem from any clustering of the degrees of freedom of V_h (see for instance [24]), we are particularly interested in local approximation spaces which are associated with a domain decomposition of the physical domain.

Definition 6.5 (Nonoverlapping domain decomposition). We call a finite collection of $M \in \mathbb{N}$ open polygonal subdomains $\mathcal{T}_H := \{\Omega_1, \Omega_2, \ldots, \Omega_M\}$ a nonoverlapping domain decomposition of the physical domain Ω, if $\bigcup_{m=1}^{M} \overline{\Omega_m} = \overline{\Omega}$ and $\Omega_m \cap \Omega_{m'} = \emptyset$ for $1 \leq m, m' \leq M$, $m \neq m'$. We collect in $\mathcal{T}_H^v$, $\mathcal{T}_H^e$, and $\mathcal{T}_H^\gamma$ the sets of all vertices, edges, and facets (which we will denote interfaces from now on),[1] respectively, associated with the partition $\mathcal{T}_H$ and define $H := \max_{m=1}^{M} \operatorname{diam} \Omega_m$. Moreover, we denote by $\Gamma := (\bigcup_{m=1}^{M} \partial \Omega_m) \setminus \partial \Omega$ the whole interface of the decomposition $\mathcal{T}_H$. Note that $\mathcal{T}_H^e = \emptyset$ for $d = 2$ and $\mathcal{T}_H^e = \mathcal{T}_H^\gamma = \emptyset$ for $d = 1$. Each of the sets $\mathcal{T}_H$, $\mathcal{T}_H^v$, $\mathcal{T}_H^e$, and $\mathcal{T}_H^\gamma$ can be decomposed into elements associated with the domain boundary and inner elements, and we collect the latter in $\mathring{\mathcal{T}}_H$, $\mathring{\mathcal{T}}_H^v$, $\mathring{\mathcal{T}}_H^e$, and $\mathring{\mathcal{T}}_H^\gamma$, respectively. For instance, for each two adjacent subdomains $\Omega_m, \Omega_{m'} \in \mathcal{T}_H$, there exists a shared interface $\Gamma_{m,m'} \in \mathring{\mathcal{T}}_H^\gamma$, while for all boundary subdomains $\Omega_m \in \mathring{\mathcal{T}}_H$ there exists at least one boundary interface $\Gamma_{m,\partial\Omega} \in \mathcal{T}_H^\gamma \setminus \mathring{\mathcal{T}}_H^\gamma$.

We can thus think of the domain decomposition as a usual grid, but without the requirements of $\mathcal{T}_H$ to actually resolve any data functions of the PDE. Given such a domain decomposition, we can abstractly define a localizing space decomposition.

Definition 6.6 (Localizing space decomposition). Let the FOM space V_h be a finite-dimensional Hilbert space with inner product and induced norm $\| \cdot \|_{V_h}^2 = (\cdot, \cdot)_{V_h}$. We call the direct sum decomposition of V_h into subdomain spaces, interface spaces, edge spaces, and vertex spaces,

$$V_h = \bigoplus_{m=1}^{M} V_h^m \oplus \bigoplus_{\gamma \in \mathcal{T}_H^\gamma} V_h^\gamma \oplus \bigoplus_{e \in \mathcal{T}_H^e} V_h^e \oplus \bigoplus_{v \in \mathcal{T}_H^v} V_h^v, \tag{6.10}$$

a localizing space decomposition.

Note that such a decomposition is not unique and can always be found. Since the reduced space shall inherit this localizing decomposition, its purpose will be three-fold: (i) offline, it allows for an independent and localized generation of the local reduced approximation spaces (compare Section 6.4), (ii) it allows to define and alter the physical domain Ω online, given that local approximation spaces for certain reference subdomains have been prepared offline, and (iii) it allows to adapt a local approximation space online (by adding basis functions or changing the local grid), while only requiring an update of local and neighboring prepared quantities (compare Section 6.6). For actual examples of space decompositions we refer to Section 6.3.

Abstractly, we do not impose any further assumptions on the FOM as well as the reduced-order model (ROM). However, given the (bi)linearity of a and f, the computational benefits of the localizing space decomposition are apparent (and are made

---

1 Note that to simplify notation we denote both the upper bound of the continuity constant and the local interfaces with γ, expecting that the respective meaning will be clear from the context.

more precise throughout the rest of this chapter). Since we allow for nonconforming approximations, in general we need to consider discrete counterparts of a and f which are only defined on the FOM space V_h and not necessarily on V, where we again refer to the following sections for examples.

Definition 6.7 (Locally decomposed full order model (FOM)). Let V_h be locally decomposable as in (6.10), and let $a_h(\cdot,\cdot;\boldsymbol{\mu}) : V_h \times V_h \to \mathbb{R}$ and $f_h(\cdot;\boldsymbol{\mu}) \in V_h'$ denote discrete variants of a and f, respectively, which are continuous and coercive with respect to the inner product of V_h. For each $\boldsymbol{\mu} \in \mathcal{P}$, find $u_h(\boldsymbol{\mu}) \in V_h$ such that

$$a_h(u_h(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) = f_h(v_h; \boldsymbol{\mu}) \quad \text{for all } v_h \in V_h. \tag{6.11}$$

The idea of projection-based localized model order reduction is to consider a local reduced approximation space for each element of the localizing space decomposition (6.10), in order to obtain a similarly decomposed reduced space $V_N \subset V_h$:

$$V_N = \bigoplus_{m=1}^{M} V_N^m \oplus \bigoplus_{\gamma \in \mathcal{T}_H^\gamma} V_N^\gamma \oplus \bigoplus_{e \in \mathcal{T}_H^e} V_N^e \oplus \bigoplus_{v \in \mathcal{T}_H^v} V_N^v, \tag{6.12}$$

with reduced subdomain spaces $V_N^m \subset V_h^m$, reduced interface spaces $V_N^\gamma \subset V_h^\gamma$, reduced edge spaces $V_N^e \subset V_h^e$, and reduced vertex spaces $V_N^v \subset V_N^v$. Similar to standard projection-based model order reduction, we obtain the ROM simply by Galerkin projection of the locally decomposed FOM (6.11) onto this locally decomposed reduced space.

Definition 6.8 (Locally decomposed reduced-order model (ROM)). Given a locally decomposed reduced space as in (6.12), for each $\boldsymbol{\mu} \in \mathcal{P}$, find $u_N(\boldsymbol{\mu}) \in V_N$ such that

$$a_h(u_N(\boldsymbol{\mu}), v_N; \boldsymbol{\mu}) = f_h(v_N; \boldsymbol{\mu}) \quad \text{for all } v_N \in V_N. \tag{6.13}$$

The main questions remain: (i) how to choose good local reduced approximation spaces to guarantee accurate and at the same time efficient reduced-order approximations, (ii) how to benefit from the localization of V_N, that is, how to obtain an offline-online decomposed scheme and in particular how to couple these local reduced approximation spaces, and (iii) how to adaptively enrich these local reduced approximation spaces online, if required. These topics will be answered throughout the remainder of this chapter, starting with examples of how to obtain localized FOMs from standard discretization schemes and how to couple the resulting local reduced spaces.

Therefore, we introduce local grids $\tau_h(\Omega_m)$ on each subdomain $\Omega_m \subset \mathcal{T}_H$, where we presume to resolve all data functions of the underlying PDE. As an analytical tool, we also define the global fine grid by $\tau_h = \bigcup_{\Omega_m \in \mathcal{T}_H} \tau_h(\Omega_m)$, which is usually not required in practical computations. For simplicity, we require the local grids of two subdomains $\Omega_m, \Omega_m' \in \mathcal{T}_H$ to match along the shared interface $\gamma_{m,m'} \in \mathring{\mathcal{T}}_H^\gamma$ and denote by $\tau_h^\gamma(\gamma_{m,m'})$ the corresponding set of all facets of τ_h which lie on $\gamma_{m,m'}$. Finally, we require that Γ does not cut any grid cells.

6.3 Coupling local approximation spaces

6.3.1 Conforming approach

There are various ways to couple local reduced spaces such that we obtain a conforming approximation, such as the partition of unity method [11] or the GFEM [8, 7, 11, 10]. However, in this section we focus on the decomposition into interface spaces and intra-element spaces, where the coupling is performed via the coupling or interface modes spanning the interface space.

6.3.1.1 The multidomain problem and the Steklov–Poincaré interface equation

First, we introduce local Hilbert spaces $H_0^1(\Omega_m) \subset V^m \subset H^1(\Omega_m)$, $m = 1, \ldots, M$, which are supposed to respect the boundary conditions on $\partial\Omega$, the local spaces $V_0^m := \{v \in V^m : v|_{\partial\Omega_m \setminus \partial\Omega} = 0\}$, and the trace space Λ associated with Γ. Moreover, we introduce local parameter-dependent bilinear and linear forms $a_m(\cdot, \cdot; \boldsymbol{\mu}) : V^m \times V^m \to \mathbb{R}$ and $f_m(\cdot; \boldsymbol{\mu}) \in V^{m'}$, $\boldsymbol{\mu} \in \mathcal{P}$, $m = 1, \ldots, M$, and the inner product $(\cdot, \cdot)_{V^m} : V^m \times V^m \to \mathbb{R}$. We may then state the variational form (6.1) equivalently as follows (see for instance [98]): For any $\boldsymbol{\mu} \in \mathcal{P}$ find $u_m(\boldsymbol{\mu}) \in V^m$, $m = 1, \ldots, M$ such that

$$a_m(u_m(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f_m(v; \boldsymbol{\mu}) \qquad \forall v \in V_0^m, \tag{6.14a}$$

$$u_m(\boldsymbol{\mu}) = u_{m'}(\boldsymbol{\mu}) \qquad \text{on } \Gamma_{m,m'}, \tag{6.14b}$$

$$\sum_{m=1}^{M} a_m(u_m(\boldsymbol{\mu}), \mathcal{E}_m\zeta; \boldsymbol{\mu}) = \sum_{m=1}^{M} f_m(\mathcal{E}_m\zeta; \boldsymbol{\mu}) \quad \forall \zeta \in \Lambda, \tag{6.14c}$$

where $\mathcal{E}_m : \Lambda \to V^m$, $m = 1, \ldots, M$, are linear and continuous extension operators.

The formulation (6.14) can then be used to derive an equation that solely acts on functions on the interface but nevertheless uniquely defines the solution $u(\boldsymbol{\mu})$ of (6.1). To that end, we introduce a parameter-dependent lifting operator $\mathcal{E}_{\Gamma\to\Omega}(\boldsymbol{\mu}) : \Lambda \to V$, where $\mathcal{E}_{\Gamma\to\Omega}(\boldsymbol{\mu})\zeta$ is defined as the minimizer of $\inf_{v(\boldsymbol{\mu})\in V} a(v(\boldsymbol{\mu}), v(\boldsymbol{\mu}); \boldsymbol{\mu})$ subject to $v(\boldsymbol{\mu})|_{\Gamma} = \zeta$. Note that we then also have

$$a_m(\mathcal{E}_{\Gamma\to\Omega}(\boldsymbol{\mu})\zeta, v; \boldsymbol{\mu}) = 0 \quad \forall v \in V_0^m \quad \text{and} \quad \mathcal{E}(\boldsymbol{\mu})_{\Gamma\to\Omega}\zeta = \zeta \quad \text{on } \Gamma \cap \partial\Omega_m. \tag{6.15}$$

Then, we can rewrite the solution $u(\boldsymbol{\mu})$ as

$$u(\boldsymbol{\mu}) = \mathcal{E}_{\Gamma\to\Omega}(\boldsymbol{\mu})(u(\boldsymbol{\mu})|_{\Gamma}) + \sum_{m=1}^{M} u_m^f(\boldsymbol{\mu}), \tag{6.16}$$

where $u_m^f(\boldsymbol{\mu}) \in V_0^m$ solves

$$a_m(u_m^f(\boldsymbol{\mu}), v; \boldsymbol{\mu}) = f_m(v; \boldsymbol{\mu}) \quad \forall v \in V_0^m, \ m = 1, \ldots, M. \tag{6.17}$$

Inserting (6.16) into (6.14c) and choosing $\mathcal{E}_m = \mathcal{E}_{\Gamma \to \Omega_m}(\boldsymbol{\mu})$ yields the Steklov–Poincaré interface equation: Find $u(\boldsymbol{\mu})|_\Gamma \in \Lambda$ such that

$$
\begin{aligned}
\sum_{m=1}^{M} & a_m(\mathcal{E}_{\Gamma \to \Omega_m}(\boldsymbol{\mu})(u(\boldsymbol{\mu})|_\Gamma), \mathcal{E}_{\Gamma \to \Omega_m}(\boldsymbol{\mu})\zeta; \boldsymbol{\mu}) \\
& = \sum_{m=1}^{M} [f_m(\mathcal{E}_{\Gamma \to \Omega_m}(\boldsymbol{\mu})\zeta; \boldsymbol{\mu}) - a_m(u_m^f(\boldsymbol{\mu}), \mathcal{E}_{\Gamma \to \Omega_m}(\boldsymbol{\mu})\zeta; \boldsymbol{\mu})] \quad \forall \zeta \in \Lambda.
\end{aligned}
\tag{6.18}
$$

Let us note that the Steklov–Poincaré interface equation and its discrete, algebraic analogon, the Schur complement system, are at the base of iterative substructuring methods (see [98, 110]), which have been combined with the reduced basis method in [80].

We may finally define a space associated with the interface V^Γ as $V^\Gamma = \{\mathcal{E}_{\Gamma \to \Omega}(\boldsymbol{\mu})\zeta \in V : \zeta \in \Lambda\}$ and obtain the decomposition $V = (\bigoplus_{m=1}^{M} V_0^m) \oplus V^\Gamma$. This decomposition is a-orthogonal thanks to (6.15).

While the computation of the (harmonic) lifting operators is inherently local (see (6.15)), the Steklov–Poincaré interface equation is posed on the whole interface Γ. To localize the latter we decompose V^Γ as we will describe next.

6.3.1.2 A conforming, localized reduced-order approximation

First, we determine basis functions associated with the vertices $v \in \mathcal{T}_H^v$. One common approach [52, 64, 20] is to require that a basis function $\psi^v \in V_h \cap [H_0^1(\bigcup_{v \subset \overline{\Omega}_m} \overline{\Omega}_m)]^z$, $z = 1, \ldots, d$, associated with some vertex v of the coarse mesh $\mathcal{T}_H$ satisfies for all Ω_m, $m = 1, \ldots, M$,

$$
(\psi^v, w)_{V^m} = 0 \quad \forall w \in V_{h;0}^m \quad \text{and} \quad \psi^v(\mathbf{x}^v) = 1, \ \psi^v(\mathbf{x}^{v'}) = 0, v \neq v'. \tag{6.19}
$$

Here, $\mathbf{x}^v$ are the (global) coordinates of the vertex v and $V_{h;0}^m := \{v \in V_h^m : v = 0 \text{ on } \partial\Omega_m \setminus \Gamma_N\}$. To uniquely define ψ^v we need to prescribe the respective values on Γ. We may for instance require ψ^v to be linear on the respective edges or bilinear on the respective interfaces (see, e.g., [20]). For multiscale problems in two space dimensions with a permeability $\kappa(\mathbf{x}_1, \mathbf{x}_2; \bar{\boldsymbol{\mu}})$ it has been suggested in [54] to prescribe

$$
\psi^v(\mathbf{x}_1, \mathbf{x}_2^v) := \left(\int_{\mathbf{x}_1^{v'}}^{\mathbf{x}_1} \frac{ds}{\kappa(s, \mathbf{x}_2^v; \bar{\boldsymbol{\mu}})} \right) \Bigg/ \left(\int_{\mathbf{x}_1^{v'}}^{\mathbf{x}_1^v} \frac{ds}{\kappa(s, \mathbf{x}_2^v; \bar{\boldsymbol{\mu}})} \right) \tag{6.20}
$$

on a horizontal edge $[\mathbf{x}_1^{v'}, \mathbf{x}_1^v] \times \{\mathbf{x}_2^v\}$ in a uniform rectangular coarse grid $\mathcal{T}_H$.

Next, we assume that we have given sets of discrete edge basis functions $\{\chi_k^e\}_{k=1}^{N_{h;0}^e} \in V_h|_e$ and discrete interface basis functions $\{\chi_k^\gamma\}_{k=1}^{N_{h;0}^\gamma} \in V_h|_\gamma$ defined on the respective edge $e \in \mathcal{T}_H^e$ or interface $\gamma \in \mathcal{T}_H^\gamma$. Here, we set $N_{h;0}^e := \dim(V_h|_{e\backslash\partial e})$ and $N_{h;0}^\gamma := \dim(V_h|_{\gamma\backslash\partial\gamma})$ as we require that χ_k^e and χ_k^γ are zero on the boundary of the edge and interface, respectively. Furthermore, we define $\Lambda_{N_{h;0}^e}^e := \text{span}\{\chi_1^e, \ldots, \chi_{N_{h;0}^e}^e\}$ and $\Lambda_{N^\gamma}^\gamma := \text{span}\{\chi_1^\gamma, \ldots, \chi_{N_{h;0}^\gamma}^\gamma\}$.

Similarly as for the vertices we may then define associated basis functions that have support on the union of subdomains that share the respective edge or interface: Find $\psi_k^\gamma \in V_h \cap [H_0^1(\bigcup_{\gamma \subset \overline{\Omega}_m} \overline{\Omega}_m)]^z, z = 1, \ldots, d, \gamma \in \mathcal{T}_H^\gamma, k = 1, \ldots, N_{h;0}^\gamma$, such that

$$(\psi_k^\gamma, w)_{V^m} = 0 \quad \forall w \in V_{h;0}^m \quad \text{and} \quad \psi_k^\gamma|_\gamma = \chi_k^\gamma. \tag{6.21}$$

Likewise, we find $\psi_k^e \in V_h \cap [H_0^1(\bigcup_{e \subset \overline{\Omega}_m} \overline{\Omega}_m)]^z, z = 1, \ldots, d, e \in \mathcal{T}_H^e, k = 1, \ldots, N_{h;0}^e$, such that

$$(\psi_k^e, w)_{V^m} = 0 \quad \forall w \in V_{h;0}^m \quad \text{and} \quad \psi_k^e|_e = \chi_k^e. \tag{6.22}$$

Again, we need to provide the value of ψ_k^e on the interfaces sharing the edge $e \in \mathcal{T}_H^e$ in order to uniquely define ψ_k^e. Similarly to above we may require that ψ_k^e is linear on the respective interfaces as suggested for instance in [20] or define a function which takes into account also the coefficient function.

Note that if the interfaces are mutually disjoint, which is for instance the case if we associate the subdomains $\Omega_m, m = 1, \ldots, M$, with the components of a structure, only the basis functions $\psi_k^\gamma, k = 1, \ldots, N_{h;0}^\gamma$ ($d = 3$), or $\psi_k^e, k = 1, \ldots, N_{h;0}^e$ ($d = 2$), are needed. Here, the values of the basis functions on the boundary of the interfaces or edges are determined by the boundary conditions on $\partial\Omega$ (see for instance [61, 60, 37, 103]).

For $N^\gamma \ll N_{h;0}^\gamma$ and $N^e \ll N_{h;0}^e$ we may now define the reduced space associated with Γ as follows:

$$V_N^\Gamma := \bigoplus_{v \in \mathcal{T}_H^v} \text{span}\{\psi^v\} \oplus \bigoplus_{e \in \mathcal{T}_H^e} \text{span}\{\psi_1^e, \ldots, \psi_{N^e}^e\} \oplus \bigoplus_{\gamma \in \mathcal{T}_H^\gamma} \text{span}\{\psi_1^\gamma, \ldots, \psi_{N^\gamma}^\gamma\}. \tag{6.23}$$

Such reduced interface spaces are for instance employed in (adaptive) CMS [52, 64], the scRBE method [61, 60, 37, 103] for mutually disjoint interfaces, or in the ArbiLoMod [20]. Subspaces of V_Γ^N are considered in certain multiscale methods. For example, in the MsFEM of Hou and Wu [54], the reduced space is spanned by the basis functions $\psi^v, v \in \mathcal{T}_H^v$. For further relations between CMS, the MsFEM, and the GFEM we refer, e. g., to [52].

Recall that the basis functions associated with the vertices, edges, and interfaces have all been computed with respect to an inner product that does not depend on the parameter (see (6.19), (6.21), and (6.22)). Therefore, we finally assume that we have also given reduced spaces $V_{N;0}^m := \text{span}\{\zeta_1^m, \ldots, \zeta_{N^m}^m\} \subset V_{h;0}^m, m = 1, \ldots, M$, that will account

for parameter variations. In detail, we obtain approximations $\tilde{\psi}_k^*(\boldsymbol{\mu})$, $* = v, e, \gamma$ by solving

$$\text{find } \tilde{b}_k^*(\boldsymbol{\mu}) \in V_{N;0}^m : \quad a_m(\psi_k^* + \tilde{b}_k^*(\boldsymbol{\mu}), w; \boldsymbol{\mu}) = 0 \quad \forall w \in V_{N;0}^m \tag{6.24}$$

and setting $\tilde{\psi}_k^*(\boldsymbol{\mu}) = \psi_k^* + \tilde{b}_k^*(\boldsymbol{\mu})$, $* = v, e, f$. Finally, we define $\tilde{b}^m(\boldsymbol{\mu}) \in V_{N;0}^m$ as the solution of

$$\text{find } \tilde{b}^m(\boldsymbol{\mu}) \in V_{N;0}^m : \quad a_m(\tilde{b}^m(\boldsymbol{\mu}), w; \boldsymbol{\mu}) = f(w, \boldsymbol{\mu}) \quad \forall w \in V_{N;0}^m. \tag{6.25}$$

Note that both $\tilde{b}_k^*(\boldsymbol{\mu})$, $* = v, e, \gamma$ and $\tilde{b}^m(\boldsymbol{\mu})$ can be interpreted as intra-element reduced basis approximations; compare to Chapters 1 and 4 of this volume of *Model order reduction*. The corresponding reduced spaces $V_{N;0}^m$, $m = 1, \ldots, M$, can for instance be constructed from solutions $b_k^*(\boldsymbol{\mu}), b^m(\boldsymbol{\mu}) \in V_{h;0}^m$, $* = v, e, f$, of

$$a_m(\psi_k^* + b_k^*(\boldsymbol{\mu}), w; \boldsymbol{\mu}) = 0 \quad \forall w \in V_{h;0}^m \tag{6.26}$$

and

$$a_m(b^m(\boldsymbol{\mu}), w; \boldsymbol{\mu}) = f(w, \boldsymbol{\mu}) \quad \forall w \in V_{h;0}^m, \tag{6.27}$$

respectively, via a standard greedy algorithm or a POD.[2] Let us also remark that for instance in the scRBE method for the approximation of each basis function ψ_k^*, $* = v, e, \gamma$, a different reduced basis space is considered, to further reduce the size of problems (6.24), (6.25). Finally, we define the reduced spaces

$$V_N = \bigoplus_{m=1}^M V_{N;0}^m \oplus V_N^\Gamma \tag{6.28}$$

and

$$V_N^\Gamma(\boldsymbol{\mu}) := \bigoplus_{v \in \mathcal{T}_H^v} \text{span}\{\tilde{\psi}^v(\boldsymbol{\mu})\} \oplus \bigoplus_{e \in \mathcal{T}_H^e} \text{span}\{\tilde{\psi}_1^e(\boldsymbol{\mu}), \ldots, \tilde{\psi}_{N^e}^e(\boldsymbol{\mu})\}$$
$$\oplus \bigoplus_{\gamma \in \mathcal{T}_H^\gamma} \text{span}\{\tilde{\psi}_1^\gamma(\boldsymbol{\mu}), \ldots, \tilde{\psi}_{N^\gamma}^\gamma(\boldsymbol{\mu})\}. \tag{6.29}$$

The global reduced approximation $u_N(\boldsymbol{\mu})$ can then be computed by performing a Galerkin projection onto the reduced space $V_N^\Gamma(\boldsymbol{\mu})$ or a Petrov–Galerkin approximation using $V_N^\Gamma(\boldsymbol{\mu})$ as a trial and V_N^Γ as a test space (see, e. g., [38, 100]). Instead of eliminating the volume degrees of freedom via (6.24), (6.25), $u_N(\boldsymbol{\mu})$ can also directly be determined by performing a Galerkin projection onto V_N (see for instance [20]).

---

2 Note that in actual practice one would construct the reduced bases only on a certain number $< M$ of reference domains; see for instance [61].

Similarly, for CMS and a fixed parameter a Galerkin projection onto V_N may be performed to compute the reduced solution; here, the reduced space $V_{N;0}^m$ is constructed from an eigenvalue problem and does not account for parameter variations (see, e. g., [52]). Finally, in the reduced basis-domain decomposition–FE (RDF) method [63] the reduced space V_N is chosen as a direct sum of $\bigoplus_{m=1}^M V_{N;0}^m$ and standard FE spaces defined on the interface or on a (small) area around the interface. Here, the intra-element reduced spaces $V_{N;0}^m$ are constructed via a greedy algorithm considering a parameterized linear combination of standard Lagrange basis functions or Fourier modes as boundary conditions. Then, a Galerkin projection on V_N is performed to compute $u_N(\boldsymbol{\mu})$.

6.3.2 Nonconforming approach

With the term *nonconforming approach* we want to classify a set of alternative techniques to solve the reduced problem on the global computational domain. A first approach consists in considering a global system of equations given by local parameterized problems and additional equations ensuring the matching between the different subdomains through the use of Lagrange multipliers. This approach has been used for solving elliptic equations in [76, 77] and Stokes equations in [74, 62].

Another approach consists in coupling local FOM spaces by interior penalty (IP) bilinear forms, inspired by discontinuous Galerkin FEM. Here, we refer to the discontinuous Galerkin RBEM [27, 6, 93] and the local reduced basis discontinuous Galerkin approach [66]. A discontinuous Galerkin approach with local POD modes has been presented in [41]. In the context of multiscale problems (cf. Example 6.2), the generalized multiscale discontinuous Galerkin method has been proposed in [29, 30] and used for solving the heat problem with phase change in [107]. We will present the LRBMS method in Section 6.3.2.2. LRBMS has been introduced in [5] and analyzed in [89, 90] for elliptic and in [87] for parabolic problems. Applications to the simulation of two-phase flow in porous media have been addressed in [65] and to battery simulation with resolved electrodes in [86].

6.3.2.1 Nonconforming coupling based on Lagrange multipliers

We want to reformulate the problem (6.14), with the idea that exact coincidence of the traces of the discrete functions (equation (6.14b)) is generally too stringent, and may, in fact, lead to imposing $u_m = 0$ on the internal interfaces; thus, the gluing process can be done in a dual way through Lagrange multipliers. We assume that local basis functions are computed in each subdomain Ω_m, $m = 1, \ldots, M$, by solving local parameterized variational problems coming from the original problem (6.1) with proper boundary conditions along the boundaries which correspond to internal ones in the

original domain. The choice of the boundary conditions is strongly related to the problem aimed to be solved. Thus, local reduced spaces are defined via these local solutions and denoted by $V_N^m, m = 1, \ldots, M$. Possible ways to construct V_N^m are presented in Section 6.4. If two or more subdomains are characterized by the same type of parameter and the same type of boundary conditions, the same local reduced space can be associated to those subdomains. For simplicity we consider different spaces for each different subdomain.

We define the following operator:

$$\mathcal{L}^{m,m'}(u(\boldsymbol{\mu}), \psi) = \int_{\Gamma_{m,m'}} (u(\boldsymbol{\mu})|_{\Omega_m} - u(\boldsymbol{\mu})|_{\Omega_{m'}})\psi ds = 0, \quad \forall \psi \in W_{m,m'}, \tag{6.30}$$

where $m, m' \in \{1, \ldots, M\}$, $\Gamma_{m,m'}$ is the interface between two adjacent subdomains denoted with the indices m and m', respectively, and $W_{m,m'}$ is the Lagrange multiplier space defined on this interface. Typical choices for the latter are low-order polynomial spaces [76, 62] or spaces constructed from snapshots (and their derivatives) [77].

A basis for $W_{m,m'}$ can then for instance be provided by the characteristic Lagrange polynomials $\psi_q, q = 1, \ldots, Q_{m,m'}$, associated with the $Q_{m,m'}$ nodes of $\Gamma_{m,m'}$.

If we suppose that Ω has $M - 1$ internal interfaces, $\Gamma_{m,m+1}, m = 1, \ldots, M - 1$, the reduced global problem of this approach reads as follows: Find $u_N(\boldsymbol{\mu}) \in V_N^1 \times \cdots \times V_N^M$, $\lambda_N \in W_{m,m+1}, m = 1, \ldots, M - 1$, such that

$$\begin{cases} a(u_N(\boldsymbol{\mu}), v_N, \boldsymbol{\mu}) + \sum_{i=1}^{M-1} \mathcal{L}^{m,m+1}(v_N, \lambda_N) = f(w, \boldsymbol{\mu}) \forall v_N \in V_N^1 \times \cdots \times V_N^M, \\ \mathcal{L}^{m,m+1}(u_N(\boldsymbol{\mu}), \psi) = 0 \quad m = 1, \ldots, M - 1, \forall \psi \in W_{m,m+1}. \end{cases} \tag{6.31}$$

6.3.2.2 Nonconforming coupling based on interior penalties

We demonstrate how to obtain a localized FOM by applying ideas from IP discontinuous Galerkin schemes with respect to the domain decomposition $\mathcal{T}_H$ in the context of the parametric multiscale Example 6.2. To define the localized FOM, we presume we are given a discretization on the full global grid τ_h (which is not used in actual computations), which we make precise by specifying the approximation space with an associated inner product and discrete variants of a and f. As a common ground for the analysis of conforming as well as nonconforming schemes, we introduce the broken Sobolev space $H^s(\tau_h(\omega)) := \{v \in L^2(\omega) \mid v|_t \in H^s(t) \ \forall t \in \tau_h(\omega)\}$, for a given grid $\tau_h(\omega)$ of some domain $\omega \subseteq \Omega$ and $s \geq 1$, and associated broken gradient operator $\nabla_h : H^1(\tau_h(\omega)) \rightarrow L^2(\omega)^d$ by $(\nabla_h v)|_t := \nabla(v|_t)$ on all $t \in \tau_h$ for $v \in H^1(\tau_h(\omega))$.

Example 6.9 (Continuous Galerkin (CG) FEM). The CG FEM scheme for the conforming approximation of Example 6.2 with respect to the full global grid τ_h is given by the

conforming approximation space of order $k \geq 1$,

$$V_h^{\mathrm{CG}}(\tau_h) := \{v \in V \mid v|_t \in \mathbb{P}_k(t) \ \forall t \in \tau_h\} \subset V \subset H^1(\tau_h),$$

where $\mathbb{P}_k(\omega)$ for any $\omega \subset \Omega$ denotes the space of all polynomials defined on ω of degree at most $k \geq 0$; the bilinear form $(\cdot, \cdot)^{\mathrm{CG}} : H^1(\tau_h) \times H^1(\tau_h) \rightarrow \mathbb{R}$, given by $(u, v)^{\mathrm{CG}} := \int_\Omega \nabla_h u \cdot \nabla_h v \, dx$, as the inner product on $V_h^{\mathrm{CG}}(\tau_h)$ (where we note that its restriction to $V \subset H^1(\tau_h)$ coincides with the V-inner product); and the discrete bilinear form $a_h^{\mathrm{CG}}(\cdot, \cdot; \boldsymbol{\mu}) : H^1(\tau_h) \times H^1(\tau_h) \rightarrow \mathbb{R}$ and linear functional $f_h^{\mathrm{CG}} : H^1(\tau_h) \rightarrow \mathbb{R}$, given by

$$a_h^{\mathrm{CG}}(u, v; \boldsymbol{\mu}) := \int_\Omega (\kappa(\boldsymbol{\mu}) \nabla_h u) \cdot \nabla_h v \, dx \quad \text{and} \quad f_h^{\mathrm{CG}}(v) := \int_\Omega q v \, dx$$

(again noting that their respective restrictions to V coincide with a and f).

The definition of the nonconforming scheme is more involved. We denote the set of faces of τ_h by τ_h^γ and to each face $\sigma \in \tau_h^\gamma$, we assign a unique normal $n_\sigma \in \mathbb{R}^d$ pointing away from t^+, where the face may be either an inner face $\sigma \in \tau_h^\gamma$, given by the intersection of two grid elements $t^+, t^- \in \tau_h$, $\sigma = \overline{t^+ \cap t^-}$, or a boundary face $\sigma \in \tau_h^\gamma$, given by $\sigma = \overline{t^+ \cap \partial\Omega}$ for some $t^+ \in \tau_h$. Since functions in the broken Sobolev space are two-valued on grid faces, we introduce the mean $\langle \cdot \rangle$ and jump $[\cdot]$ on a boundary face by $\langle v \rangle := [v] := v|_{t^+}$ and by $\langle v \rangle := \frac{1}{2}(v|_{t^+} + v|_{t^-})$ and $[v] := v|_{t^+} - v|_{t^-}$, respectively, on any other face.

Considering the family of IP discontinuous Galerkin schemes, we present the symmetric variant for ease of notation, and refer to the symmetric weighted variant [40], which is particularly well suited for multiscale problems with highly varying or high-contrast coefficients.

Example 6.10 (IP discontinuous Galerkin FEM). The symmetric IP discontinuous Galerkin FEM scheme for the nonconforming approximation of Example 6.2 with respect to the full global grid τ_h is given by the nonconforming approximation space of order $k \geq 1$,

$$V_h^{\mathrm{DG}}(\tau_h) := \{v \in L^2(\Omega) \mid v|_t \in \mathbb{P}_k(t) \ \forall t \in \tau_h\} \subset H^1(\tau_h);$$

the bilinear form $(\cdot, \cdot)^{\mathrm{DG}} : H^1(\tau_h) \times H^1(\tau_h) \rightarrow \mathbb{R}$, given by

$$(u, v)^{\mathrm{DG}} := (u, v)^{\mathrm{CG}} + \sum_{\sigma \in \tau_h^\gamma} (u, v)_\sigma^p \quad \text{with} \quad (u, v)_\sigma^p := \int_\sigma h_\sigma^{-1} [u][v] \, ds,$$

as inner product on $V_h^{\mathrm{DG}}(\tau_h)$, where h_σ is a positive number associated with each face $\sigma \in \tau_h^\gamma$, e. g., $h_\sigma := \mathrm{diam}(\sigma)$ for $d \geq 2$ and $h_\sigma := \min\{\mathrm{diam}(t^+), \mathrm{diam}(t^-)\}$ for $d = 1$; and the linear functional $f_h^{\mathrm{DG}} : H^1(\tau_h) \rightarrow \mathbb{R}$ given by $f_h^{\mathrm{DG}}(v) := f_h^{\mathrm{CG}}(v)$ and the discrete bilinear form $a_h^{\mathrm{DG}}(\cdot, \cdot; \boldsymbol{\mu}) : H^2(\tau_h) \times H^2(\tau_h) \rightarrow \mathbb{R}$, given by

$$a_h^{\mathrm{DG}}(u, v; \boldsymbol{\mu}) := a_h^{\mathrm{CG}}(u, v; \boldsymbol{\mu}) + \sum_{\sigma \in \tau_h^\gamma} a_\sigma(u, v; \boldsymbol{\mu})$$

with the face bilinear form a_σ for any $\sigma \in \tau_h^\gamma$ given by

$$a_\sigma(v, u; \boldsymbol{\mu}) := a_\sigma^c(v, u; \boldsymbol{\mu}) + a_\sigma^c(u, v; \boldsymbol{\mu}) + (u, v)_\sigma^p w_\sigma$$

with $a_\sigma^c(u, v; \boldsymbol{\mu}) := \int_\sigma -\langle (\kappa(\boldsymbol{\mu})\nabla_h v) \cdot n_\sigma \rangle [u] \, ds$ and a user-dependent penalty weight $w_\sigma > 0$, such that a_h^{DG} is continuous and coercive with respect to the above inner product.

The main idea of an *IP localized FOM* is to consider the restriction of either of the above discretization schemes to each subdomain of the domain decomposition, and to again couple those with IP techniques along the interfaces of the subdomain. We thus choose $* \in \{CG, DG\}$ and obtain the localized FOM space in the sense of Definition 6.6 as a direct sum of subdomain spaces (with empty interface, edge, and vertex spaces)

$$V_h := \bigoplus_{m=1}^M V_h^m, \quad \text{with} \quad V_h^m := \{v|_{\Omega_m} \mid v \in V^*\},$$

with associated inner product $(\cdot, \cdot)_{V_h} : V_h \times V_h \to \mathbb{R}$ given by

$$(u, v)_{V_h} := \sum_{m=1}^M (u|_{\Omega_m}, v|_{\Omega_m})^* + \sum_{\Gamma' \in \mathring{\mathcal{T}}_H^\gamma} \sum_{\sigma \in \tau_h^\gamma(\Gamma')} (u, v)_\sigma^p.$$

We also define the linear functional $f_h : V_h \to \mathbb{R}$ by $f_h := f_h^*$ and, in a similar manner as above, the nonconforming bilinear form $a_h(\cdot, \cdot; \boldsymbol{\mu}) : V_h \times V_h \to \mathbb{R}$ by

$$a_h(u, v; \boldsymbol{\mu}) := \sum_{m=1}^M a_h^*(u|_{\Omega_m}, v|_{\Omega_m}; \boldsymbol{\mu}) + \sum_{\Gamma' \in \mathring{\mathcal{T}}_H^\gamma} \sum_{\sigma \in \tau_h^\gamma(\Gamma')} a_\sigma(u, v; \boldsymbol{\mu}).$$

We have thus fully specified a localized FOM in the sense of Definition 6.7 and comment on two special cases: for $* = CG$ and a trivial domain decomposition of a single subdomain, $\mathcal{T}_H = \{\Omega\}$, we obtain the above standard CG FEM, while for $* = DG$ the resulting FOM coincides with the above standard symmetric IP discontinuous Galerkin FEM.

To make the coupling more precise, we may rearrange the above terms to obtain a localization of a_h with respect to the domain decomposition in the sense of

$$a_h(u, v; \boldsymbol{\mu}) = \sum_{m=1}^M a_h^m(u, v; \boldsymbol{\mu}) + \sum_{\Gamma' \in \mathring{\mathcal{T}}_H^\gamma} a_h^{\Gamma'}(u, v; \boldsymbol{\mu}),$$

with the subdomain and interface bilinear forms

$$a_h^m(u, v; \boldsymbol{\mu}) := a_h^*(u|_{\Omega_m}, v|_{\Omega_m}; \boldsymbol{\mu}) + \sum_{\Gamma' \in \mathring{\mathcal{T}}_H^\gamma \cap \Omega_m} \sum_{\sigma \in \tau_h^\gamma(\Gamma')} a_\sigma(u|_{\Omega_m}, v|_{\Omega_m}; \boldsymbol{\mu}),$$

$$a_h^{\Gamma'}(u, v; \boldsymbol{\mu}) := \sum_{\sigma \in \tau_h^\gamma(\Gamma')} \{a_\sigma(u|_{\Omega^+}, v|_{\Omega^-}; \boldsymbol{\mu}) + a_\sigma(u|_{\Omega^-}, v|_{\Omega^+}; \boldsymbol{\mu})\},$$

respectively, for all $1 \leq m \leq M$ and all $\Gamma' \in \mathring{\mathcal{T}}_H^\gamma$, with the subdomains $\Omega^+, \Omega^- \in \mathcal{T}_H$ sharing the interface Γ'.

Now given a local reduced space $V_N^m \subset V_h^m$ for each subdomain we obtain the locally decomposed broken reduced space in the sense of (6.12) by

$$V_N = \bigoplus_{m=1}^M V_N^m \subset V_h.$$

Using the above decomposition of a_h into subdomain and interface contributions, we can readily observe that the locally decomposed ROM can be offline-online decomposed by local computations, namely, by projection of the subdomain bilinear forms $a_h^m(\cdot, \cdot; \boldsymbol{\mu})$ onto $V_N^m \times V_N^m$ and the interface bilinear forms $a_h^{\Gamma'}(\cdot, \cdot; \boldsymbol{\mu})$ onto $V_N^m \times V_N^n$, with $1 \leq m, n \leq M$, such that $\Omega^+ = \Omega_m$ and $\Omega^- = \Omega_n$, respectively.

We thus obtain a sparse matrix representation of the resulting reduced system, with a sparsity pattern which coincides with the one from standard IP discontinuous Galerkin schemes.

6.4 Preparation of local approximation spaces

Both couplings that yield a conforming and nonconforming approximation require either reduced spaces $\Lambda_{N^\gamma}^\gamma \subset V_h|_\gamma$ for interfaces and/or edges $\Lambda_{N^e}^e \subset V_h|_e$ (Section 6.3.1) or reduced spaces V_N^m (Section 6.3.2) or both. As the generation of edge basis functions can be done analogously to the construction of interface basis functions we restrict ourselves to the latter in order to simplify notation. To fix the setting we thus consider the task of finding a suitable reduced space on either a subdomain $\Omega_m \subsetneq \Omega_{\text{out}} \subset \Omega$ with $\text{dist}(\Gamma_{\text{out}}, \partial\Omega_m) \geq \rho > 0$, $\Gamma_{\text{out}} := \partial\Omega_{\text{out}} \setminus \partial\Omega$ or an interface $\Gamma_{m,m'} \subset \partial\Omega_m$, where $\text{dist}(\Gamma_{\text{out}}, \Gamma_{m,m'}) \geq \rho > 0$. Possible geometric configurations of the oversampling domain Ω_{out} are illustrated in Figure 6.1.
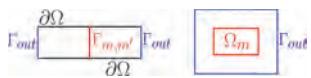
Figure 6.1: Illustration of possible decompositions of Ω_{out} with respect to $\Gamma_{m,m'}$ or Ω_m.

We will first briefly discuss in Section 6.4.1 reduced spaces that are spanned by *polynomials or solutions of "standard" eigenvalue problems* and are thus related to the spectral element method or hp-FEM. Subsequently, in Section 6.4.2 we will present reduced spaces that are generated from local solutions of the PDE, are thus of *empirical* nature, and are optimal in the sense of Kolmogorov. We will also show how those optimal basis

functions can be efficiently and accurately approximated by means of random sampling.

6.4.1 Polynomial-based local approximation spaces

CMS as introduced in [59, 12] relies on free vibration modes or eigenmodes of local, constrained eigenvalue problems [59, 12, 17, 52, 64, 51] for the approximation within subdomains. To couple the modes at the interfaces a reduced interface space spanned by eigenmodes is employed [59, 12, 17, 52, 64, 51].

A combination of domain decomposition and reduced basis methods has first been considered in the RBEM [76]. Here, inspired by the mortar spectral element method [16], the Lagrange multiplier space $W_{m,m'}$ as defined in Section 6.3.2 is chosen as a low-order polynomial space. The reduced basis hybrid method [62] extends the RBEM by additionally considering a coarse FE discretization on the whole domain to account for continuity of normal stresses and also employs a low-order polynomial Lagrange multiplier space on the interface. For the scRBE method a reduced interface space spanned by the eigenvectors of a discrete generalized eigenvalue problem based on the Laplacian has been suggested in [61, 60] and eigenmodes of a singular Sturm–Liouville eigenproblem have been used in [37]. Finally, in the RDF method [63] a standard FE space is considered on the interface or on a (small) area around the interface.

6.4.2 Local approximation spaces based on empirical training

In this subsection we are concerned with local approximation spaces that are constructed from local solutions of the PDE; those approaches are often called *empirical*. Basis functions on the interfaces selected from local snapshots are for instance suggested in [37], where an empirical pairwise training procedure for interface reduction within the scRBE context is developed, and within a heterogeneous domain decomposition method in [83]. Local approximation spaces that are optimal in the sense of Kolmogorov have been introduced for subdomains within the GFEM in [10] for parameter-independent PDEs and for interfaces within static condensation procedures [103] for parameterized PDEs. While the authors of [103] introduce and analyze a spectral greedy algorithm to deal with parameter variations, [109] suggests using a POD making use of the hierarchical approximate POD [53]. Those optimal local spaces both allow for a rigorous a priori theory and yield a rapidly (and often exponentially) convergent approximation; in certain cases the superalgebraic convergence can be proved [10]. Recently, in [109, 108] the results in [10, 9, 103] have been generalized from linear differential operators whose associated bilinear form is coercive to elliptic, inf-sup stable ones. In [21] it has been shown that those optimal local approximation

spaces can be efficiently approximated by transferring methods from randomized numerical linear algebra [46]; the local approximation spaces in [21] are constructed from local solutions of the PDE with random boundary conditions. Local reduced spaces generated from random snapshots have also been suggested in [20, 37] and methods from randomized linear algebra have been exploited in the FETI-2λ domain decomposition method in [113] and in [23] for the generalized multiscale FEM.

We will first present the optimal local approximation spaces as introduced in [10, 103] for a fixed parameter $\mathcal{P} = \{\bar{\boldsymbol{\mu}}\}$, subsequently discuss their approximation via random sampling, and conclude this subsection with the discussion of the general case $\mathcal{P} \neq \{\bar{\boldsymbol{\mu}}\}$. To simplify notation we will omit $\bar{\boldsymbol{\mu}}$ as long as it is fixed.

6.4.2.1 Optimal local approximation spaces for $\mathcal{P} = \{\bar{\boldsymbol{\mu}}\}$

To enable maximum flexibility regarding the shape of Ω on the user's side, we assume that we do not have any a priori knowledge of the shape of Ω when constructing the ROM. We thus know that the global solution u satisfies the considered PDE locally on Ω_{out} but suppose that the trace of u on $\partial\Omega_{\text{out}}$ is *unknown* to us. Therefore, we aim at approximating all local solutions u_{loc} of

$$a_{\text{loc}}(u_{\text{loc}}, v) = f_{\text{loc}}(v) \quad \forall v \in V_{\text{loc}}, \tag{6.32}$$

with *arbitrary Dirichlet boundary conditions on* Γ_{out}. Here, the Hilbert space V_{loc} is defined such that $[H_0^1(\Omega_{\text{out}})]^z \subset V_{\text{loc}} \subset [H^1(\Omega_{\text{out}})]^z$, $z = 1, \ldots, d$, respecting the boundary conditions on $\partial\Omega$, and $a_{\text{loc}} : [H^1(\Omega_{\text{out}})]^z \times [H^1(\Omega_{\text{out}})]^z \to \mathbb{R}$, $f_{\text{loc}} : V_{\text{loc}} \to \mathbb{R}$ are local bilinear and linear forms. We will first restrict ourselves to the case $f_{\text{loc}} = 0$, $g_D = 0$, and $\partial\Omega_m \cap \Gamma_D = \emptyset$; the general case will be dealt with at the end of this subsection. We may then define the space of all local solutions of the PDE as

$$\mathcal{H} := \left\{ w \in \left[H^1(\Omega_{\text{out}}) \right]^z : w \text{ solves } (6.32), w = 0 \text{ on } \Gamma_D \cap \partial\Omega_{\text{out}} \right\}, \quad z = 1, \ldots, d. \tag{6.33}$$

As suggested in [10, 103] we introduce a transfer operator $\mathcal{T} : \mathcal{S} \to \mathcal{R}$ for Hilbert spaces $\mathcal{S}$ and $\mathcal{R}$, where $\mathcal{S} = \{w|_{\Gamma_{\text{out}}} : w \in \mathcal{H}\}$. We define $\mathcal{T}$ for interfaces or subdomains, respectively, for $w \in \mathcal{H}$ as

$$\mathcal{T}(w|_{\Gamma_{\text{out}}}) = (w - P_{\Omega_{\text{out}}}(w))|_{\Gamma_{m,m'}} \quad \text{or} \quad \mathcal{T}(w|_{\Gamma_{\text{out}}}) = (w - P_{\Omega_m}(w))|_{\Omega_m} \tag{6.34}$$

and set $\mathcal{R} = \{v|_{\Gamma_{m,m'}} : v = w - P_{\Omega_{\text{out}}}(w), w \in \mathcal{H}\}$ or $\mathcal{R} = \{(w - P_{\Omega_m}w)|_{\Omega_m} : w \in \mathcal{H}\}$. Here, P_D, $D \subset \Omega_{\text{out}}$, denotes an orthogonal projection onto the kernel of the bilinear form; for further details see [21, 103]. In the case of heat conduction we would for instance subtract the mean value of the respective function on D. Note that subtracting this projection is necessary to prove compactness of the transfer operator $\mathcal{T}$. The key argument to show compactness of $\mathcal{T}$ is Caccioppoli's inequality, which estimates the

energy norm of a function in $\mathcal{H}$ on Ω_m in terms of the L^2-norm on Ω_{out} of the respective function. Using the Hilbert–Schmidt theorem and Theorem 2.2 in [95, Chapter 4] it can then be shown that certain eigenfunctions of $\mathcal{T}^*\mathcal{T}$ span the optimal local approximation space, where $\mathcal{T}^* : \mathcal{R} \to \mathcal{S}$ denotes the adjoint operator of $\mathcal{T}$. As we aim at approximating $\mathcal{H}$ and thus a whole set of functions, the concept of optimality of Kolmogorov [68] is used: A subspace $\mathcal{R}_n \subset \mathcal{R}$ of dimension at most n for which holds

$$d_n(\mathcal{T}(\mathcal{S}); \mathcal{R}) = \sup_{\psi \in \mathcal{S}} \inf_{\zeta \in \mathcal{R}_n} \frac{\|\mathcal{T}\psi - \zeta\|_{\mathcal{R}}}{\|\psi\|_{\mathcal{S}}}$$

is called an optimal subspace for $d_n(\mathcal{T}(\mathcal{S}); \mathcal{R})$, where the Kolmogorov n-width $d_n(\mathcal{T}(\mathcal{S}); \mathcal{R})$ is defined as

$$d_n(\mathcal{T}(\mathcal{S}); \mathcal{R}) := \inf_{\substack{\mathcal{R}_n \subset \mathcal{R} \\ \dim(\mathcal{R}_n)=n}} \sup_{\psi \in \mathcal{S}} \inf_{\zeta \in \mathcal{R}_n} \frac{\|\mathcal{T}\psi - \zeta\|_{\mathcal{R}}}{\|\psi\|_{\mathcal{S}}}.$$

We summarize the findings about the optimal local approximation spaces in the following theorem.

Theorem 6.11 (Optimal local approximation spaces [10, 103]). *The optimal approximation space for $d_n(\mathcal{T}(\mathcal{S}); \mathcal{R})$ is given by*

$$\mathcal{R}_n := \text{span}\{\chi_1^{sp}, \ldots, \chi_n^{sp}\}, \quad \text{where } \chi_j^{sp} = \mathcal{T}\phi_j, \ j = 1, \ldots, n, \tag{6.35}$$

and λ_j are the largest n eigenvalues and ϕ_j the corresponding eigenfunctions that satisfy the following transfer eigenvalue problem: Find $(\phi_j, \lambda_j) \in (\mathcal{S}, \mathbb{R}^+)$ such that

$$(\mathcal{T}\phi_j, \mathcal{T}w)_{\mathcal{R}} = \lambda_j(\phi_j, w)_{\mathcal{S}} \quad \forall w \in \mathcal{S}. \tag{6.36}$$

Moreover, we have

$$d_n(\mathcal{T}(\mathcal{S}); \mathcal{R}) = \sup_{\xi \in \mathcal{S}} \inf_{\zeta \in \mathcal{R}_n} \frac{\|\mathcal{T}\xi - \zeta\|_{\mathcal{R}}}{\|\xi\|_{\mathcal{S}}} = \sqrt{\lambda_{n+1}}. \tag{6.37}$$

Remark 6.12. We emphasize that the optimal space $\mathcal{R}_n$ is optimal in the sense of Kolmogorov for the approximation of the range of $\mathcal{T}$ and *not* necessarily for the approximation of $u(\boldsymbol{\mu})$. Moreover, we remark that χ_i^{sp} are the left singular vectors and $\sqrt{\lambda_i}$ the singular values of $\mathcal{T}$.

Next, for $f_{\text{loc}} \neq 0$ but still $g_D = 0$ we solve the following problem: Find $u_{\text{loc}}^f \in V_{\text{loc}}$ such that $a_{\text{loc}}(u_{\text{loc}}^f, v) = f_{\text{loc}}(v)$ for all $v \in V_{\text{loc}}$ and augment the space $\mathcal{R}_n$ with either $u_{\text{loc}}^f|_{\Omega_m}$ or $u_{\text{loc}}^f|_{\Gamma_{m,m'}}$. To take nonhomogeneous Dirichlet boundary conditions into account one can proceed for instance with a standard lifting approach, adjusting f_{loc} accordingly. Note that for homogeneous boundary conditions we proceed very similarly to above, prescribing "arbitrary" boundary conditions on Γ_{out} and homogeneous

boundary conditions on $\partial\Omega \cap \partial\Omega_{\text{out}}$. The optimal local approximation spaces for subdomains are then defined as

$$\mathcal{R}_n^+ := \text{span}\{\chi_1^{sp}, \ldots, \chi_n^{sp}, u_{\text{loc}}^f|_{\Omega_m}\} \oplus \ker(a_m(\cdot, v)) \tag{6.38}$$

and similarly for interfaces as

$$\mathcal{R}_n^+ := \text{span}\{\chi_1^{sp}, \ldots, \chi_n^{sp}, u_{\text{loc}}^f|_{\Gamma_{m,m'}}\} \oplus \ker(a_m(\cdot, v))|_{\Gamma_{m,m'}}. \tag{6.39}$$

Here, $\ker(a_m(\cdot, v))$ denotes the kernel of the mapping $a_m(\cdot, v) : [H^1(\Omega_m)]^z \to \mathbb{R}$, $z = 1, \ldots, d$, $v \in V_0^m$, for the bilinear form a_m defined in Section 6.3.1. In the case $\partial\Omega_m \cap \Gamma_D \neq \emptyset$, all modifications in this subsection involving the kernel of the bilinear form are waived.

The result in (6.37) can be exploited to derive an a priori error bound for the approximation error between the solution $u(\bar{\boldsymbol{\mu}})$ of (6.1) still for a fixed reference parameter $\bar{\boldsymbol{\mu}}$ and the optimal static condensation approximation $u^n(\bar{\boldsymbol{\mu}})$ as stated in the following proposition.

Proposition 6.13 (A priori error bound [103]). *Assume that the interfaces $\gamma \in \mathcal{T}_H^\gamma$ are mutually disjoint, that all interfaces have the same geometry, and that each Ω_m, $m = 1, \ldots, M$, has exactly two interfaces. Let $u(\bar{\boldsymbol{\mu}})$ be the (exact) solution of (6.1) for a fixed parameter $\bar{\boldsymbol{\mu}}$. Moreover, let $u_{n_+}(\bar{\boldsymbol{\mu}})$ be the static condensation approximation defined in Section 6.3.1, where we employ the optimal interface space $\mathcal{R}_n^+$ for each $\gamma \in \mathcal{T}_H^\gamma$ and assume that the error due to the intra-element reduced basis approximation is zero. Then, we have the following a priori error bound:*

$$\frac{\||u(\bar{\boldsymbol{\mu}}) - u_{n_+}(\bar{\boldsymbol{\mu}})\||_{\bar{\boldsymbol{\mu}}}}{\||u(\bar{\boldsymbol{\mu}})\||_{\bar{\boldsymbol{\mu}}}} \leq \#\gamma \max_{\gamma \in \mathcal{T}_H^\gamma}(C_\gamma \sqrt{\lambda_{n+1}^\gamma}), \tag{6.40}$$

where $\#\gamma$ denotes the number of interfaces in $\mathcal{T}_H^\gamma$ and λ_{n+1}^γ is the $(n+1)$-th eigenvalue of (6.36) for the interface $\gamma \in \mathcal{T}_H^\gamma$. The constant C_γ depends only on the subdomains that share the interface γ and neither on Ω nor on $u(\bar{\boldsymbol{\mu}})$.

To define reduced interface spaces $\Lambda_{N^\gamma}^\gamma$, $\gamma \in \mathcal{T}_H^\gamma$, and reduced spaces V_N^m, $m = 1, \ldots, M$, we approximate (6.36) with FEs. To that end, we introduce a conforming FE space $V_{h;\text{loc}} \subset V_{\text{loc}}$, the FE source space $S := \{v|_{\Gamma_{\text{out}}} : v \in V_h\}$ of dimension N_S, and the FE range space $R := \{(v - P_{\Omega_{\text{out}}}(v))|_{\Gamma_{m,m'}} : v \in V_h\}$ or $R := \{(v - P_{\Omega_m})|_{\Omega_m} : v \in V_h\}$ with $\dim(R) = N_R$. We may then define the discrete transfer operator $T : S \to R$ for $w \in \mathcal{H}_h = \{w \in V_h|_{\Omega_{\text{out}}} : a_{\text{loc}}(w, \varphi) = 0 \, \forall \varphi \in V_{h;\text{loc}}, \ w = 0 \text{ on } \Gamma_D \cap \partial\Omega_{\text{out}}\}$ as

$$T(w|_{\Gamma_{\text{out}}}) = (w - P_{\Omega_{\text{out}}}(w))|_{\Gamma_{m,m'}} \quad \text{or} \quad T(w|_{\Gamma_{\text{out}}}) = (w - P_{\Omega_m}(w))|_{\Omega_m}. \tag{6.41}$$

In order to define a matrix form of the transfer operator we introduce degree of freedom mappings $\mathbb{B}_{S \to V_h|_{\Omega_{\text{out}}}} \in \mathbb{R}^{\dim(V_h|_{\Omega_{\text{out}}}) \times N_S}$ and $\mathbb{B}_{V_h|_{\Omega_{\text{out}}} \to R} \in \mathbb{R}^{N_R \times \dim(V_h|_{\Omega_{\text{out}}})}$ that map

the degrees of freedom of S to the degrees of freedom of $V_h|_{\Omega_{\text{out}}}$ and the degrees of freedom of $V_h|_{\Omega_{\text{out}}}$ to the degrees of freedom of R, respectively. Moreover, we introduce the stiffness matrix $\mathbb{A}_{\text{loc}}$ obtained from the FE discretization of (6.32), where we assume that in the rows associated with the Dirichlet degrees of freedom the nondiagonal entries are zero and the diagonal entries equal one. By denoting by $\boldsymbol{\zeta}$ the FE coefficients of $\zeta \in S$ and by defining $\mathbb{P}_D$ as the matrix of the orthogonal projection on the kernel of the bilinear form on $D \subset \Omega_{\text{out}}$, we obtain the following matrix representation $\mathbb{T} \in \mathbb{R}^{N_R \times N_S}$ of the transfer operator for subdomains

$$\mathbb{T}\,\boldsymbol{\zeta} = (1 - \mathbb{P}_{\Omega_m})\mathbb{B}_{V_h|_{\Omega_{\text{out}}} \to R}\,\mathbb{A}^{-1}\mathbb{B}_{S \to V_h|_{\Omega_{\text{out}}}}\,\boldsymbol{\zeta} \tag{6.42}$$

and interfaces

$$\mathbb{T}\,\boldsymbol{\zeta} = \mathbb{B}_{V_h|_{\Omega_{\text{out}}} \to R}(1 - \mathbb{P}_{\Omega_{\text{out}}})\mathbb{A}^{-1}\mathbb{B}_{S \to V_h|_{\Omega_{\text{out}}}}\,\boldsymbol{\zeta}. \tag{6.43}$$

Finally, we denote by $\mathbb{M}_S$ the inner product matrix of S and by $\mathbb{M}_R$ the inner product matrix of R. Then, the FE approximation of the transfer eigenvalue problem reads as follows: Find the eigenvectors $\boldsymbol{\zeta}_j \in \mathbb{R}^{N_S}$ and the eigenvalues $\lambda_j \in \mathbb{R}_0^+$ such that

$$\mathbb{T}^t \mathbb{M}_R \mathbb{T}\,\boldsymbol{\zeta}_j = \lambda_j\,\mathbb{M}_S\,\boldsymbol{\zeta}_j. \tag{6.44}$$

The coefficients of the FE approximation of the basis functions $\{\chi_{h,1}^{sp}, \ldots, \chi_{h,n}^{sp}\}$ of the discrete optimal local approximation space

$$R_n := \operatorname{span}\{\chi_{h,1}^{sp}, \ldots, \chi_{h,n}^{sp}\} \tag{6.45}$$

are then given by $\boldsymbol{\chi}_{h,j}^{sp} = \mathbb{T}\,\boldsymbol{\zeta}_j$, $j = 1, \ldots, n$. Adding the representation of the right-hand side, the boundary conditions, and a basis of the kernel of the bilinear form yields the optimal spaces $\Lambda_{N^\gamma}^\gamma$ and V_N^m.

Note that in actual practice we would not assemble the matrix $\mathbb{T}$. Instead one may solve the PDE locally N_S times prescribing the basis functions of S as Dirichlet boundary conditions on Γ_{out} and subsequently assemble and solve the transfer eigenvalue problem. Alternatively, one may pass $\mathbb{T}$ implicitly to the Lanczos method. For instance, the implicitly restarted Lanczos method as implemented in ARPACK [73] requires $\mathcal{O}(n)$ local solutions of the PDE in each iteration and applications of the adjoint T^*. In the next subsection we will show how methods from randomized linear algebra [46, 32, 78, 79] can be used to compute an approximation of the optimal local approximation spaces. However, beforehand, we conclude this subsection with some numerical experiments on the transfer eigenvalues and thus via Proposition 6.13 on the convergence behavior of the relative approximation error.

To this end, we present the simplified model for a ship stiffener from [103]: We consider $\overline{\Omega}_{\text{out}} = \bar{\Omega}_1 \cup \bar{\Omega}_2$ and $\Gamma_{m,m'} = \Gamma_{1,2} = \bar{\Omega}_1 \cap \bar{\Omega}_2$, where Ω_2 is depicted in Figure 6.2, Ω_1 is just a shifted version of Ω_2, and the part of Γ_{out} in Ω_2 is indicated in yellow in

Figure 6.2: Mesh in the subdomain Ω_2 for the ship stiffener. The part of Γ_{out} in Ω_2 is indicated in yellow and on the opposite we have the interface $\Gamma_{m,m'}$. In the red shaded areas Young's modulus may be varied between 1 and 20 and in the gray areas we consider $E(\boldsymbol{\mu}) \equiv 1$.

Figure 6.2. We allow $E(\boldsymbol{\mu})$ to vary in the red areas of the subdomains between 1 and 20 and prescribe $E(\boldsymbol{\mu}) \equiv 1$ in the gray areas; we choose $\boldsymbol{G}(\boldsymbol{\mu}) = (0, 0, 0)^T$.

In detail, we consider $\Omega_1 = (-0.7, 0.7) \times (-0.05, 0.05) \times (-0.6, 0.6)$, $\Omega_2 = (0.7, 2.1) \times (-0.05, 0.05) \times (-0.6, 0.6)$, and $\Gamma_{\text{out}} = \{-0.7\} \times (-0.05, 0.05) \times (-0.6, 0.6) \cup \{2.1\} \times (-0.05, 0.05) \times (-0.6, 0.6)$. We employ a conforming linear FE space associated with the mesh depicted in Figure 6.2, resulting in $N = 13{,}125$ degrees of freedom per subdomain and an FE interface space of dimension $N_\Gamma = 375$. Finally, we equip both S and R with a lifting inner product based on the lifting operator $\mathcal{E}_{\Gamma \to \Omega_m}(\bar{\boldsymbol{\mu}})$ defined in Section 6.3.1.1; for further details we refer to [103].

We consider different values for Young's modulus (ratios) E_i^r, $i = 1, 2$, in the red areas of the subdomains and observe in Figure 6.3 for the ship stiffener application an exponential convergence of order $\approx e^{-n}$ of the eigenvalues $\lambda_n(\boldsymbol{\mu})$ and thus the static condensation approximation. We emphasize that we observe in Figure 6.3 that the eigenvalues associated with the stiffened plate ($E_1^r = E_2^r = 20$) decay fastest, while we see the slowest decay for the nonstiffened plate ($E_1^r = E_2^r = 1$). This is consistent with the expectation that stiffening the plate decreases the deflection of the plate, eliminating the higher eigenmodes. Moreover, an inspection of the optimal interface modes reveals many "classical" mode shapes such as bending or torsional modes of beams and demonstrates again the physical significance of the optimal modes. Also for beams of different shapes, including an I-beam with a crack and thus an irregular domain, an exponential convergence of the transfer eigenvalues and the physical significance of the transfer eigenmodes can be observed; for further details see [103].
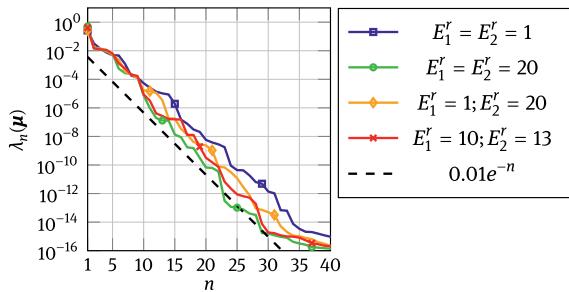
Figure 6.3: Eigenvalues $\lambda_n(\boldsymbol{\mu})$ for different Young modulus ratios E_i^r in Ω_i, $i = 1, 2$.

6.4.2.2 Randomized training

In order to compute an efficient approximation R_n^{rand} of R_n the adaptive randomized range approximation algorithm, Algorithm 6.1, as suggested in [21] iteratively enhances the reduced space with applications of T to a random function until a certain convergence criterion is satisfied.

Algorithm 6.1: Adaptive randomized range approximation.

   Input : Operator T, target accuracy tol, number of test vectors n_t, maximum
             failure probability $\varepsilon_{\text{algofail}}$
   Output: space R_n^{rand} with property $P(\|T - P_{R_n^{\text{rand}}}T\| \leq \text{tol}) > (1 - \varepsilon_{\text{algofail}})$

1 **Initialize:** $B \leftarrow \emptyset$, $M \leftarrow \{TD_S^{-1}\boldsymbol{r}_1, \ldots, TD_S^{-1}\boldsymbol{r}_{n_t}\}$

2 **Compute error estimator factors:**

3 $\varepsilon_{\text{testfail}} \leftarrow \varepsilon_{\text{algofail}}/N_T$; $c_{\text{est}} \leftarrow \left[\sqrt{2\lambda_{\min}^{\mathbb{M}_S}}\,\text{erf}^{-1}(\sqrt[n]{\varepsilon_{\text{testfail}}})\right]^{-1}$

4 **while** $(\max_{t \in M} \|t\|_R) \cdot c_{\text{est}} > \text{tol}$ **do**

5      $B \leftarrow B \cup (TD_S^{-1}\boldsymbol{r})$

6      $B \leftarrow \text{orthonormalize}(B)$

7   **orthogonalize test vectors:** $M \leftarrow \{t - P_{\text{span}\{B\}}t \mid t \in M\}$

8 **return** $R_n^{\text{rand}} = \text{span}\{B\}$

In detail, in each iteration on line 5 we draw a new random vector $\boldsymbol{r} \in \mathbb{R}^{N_S}$ whose entries are independent and identically distributed random variables with standard normal distribution. Then, we employ the mapping $D_S^{-1}: \mathbb{R}^{N_S} \to S$ to define a unique FE function in S whose coefficients are the components of $\boldsymbol{r}$. Subsequently, we apply the transfer operator T to $D_S^{-1}\boldsymbol{r}$, meaning that we solve the PDE locally on Ω_{out} with random boundary conditions and restrict the solution to Ω_m or $\Gamma_{m,m'}$; the resulting function is added to the set of basis functions B. Finally, the basis B is orthonormalized. Note that the orthonormalization is numerically challenging, as the basis functions are nearly linear dependent when span$\{B\}$ is already a good approximation of the range of T; in [21] using the numerically stable Gram–Schmidt with adaptive reiteration from [19] is suggested. The main loop of the algorithm is terminated when the following a posteriori norm estimator is smaller than the desired tolerance tol.

Proposition 6.14 (A probabilistic a posteriori norm estimator [21]). *Let $\boldsymbol{r}_i, i = 1, \ldots, n_t$, be n_t random normal test vectors and $\lambda_{\min}^{\mathbb{M}_S}$ and $\lambda_{\max}^{\mathbb{M}_S}$ the smallest and largest eigenvalues of the matrix of the inner product in S. Then, the a posteriori norm estimator*

$$\Delta(n_t, \varepsilon_{\text{testfail}}) := c_{\text{est}}(n_t, \varepsilon_{\text{testfail}}) \max_{i \in 1, \ldots, n_t} \left\|(T - P_{R_n^{\text{rand}}}T)\, D_S^{-1}\,\boldsymbol{r}_i\right\|_R \tag{6.46}$$

satisfies

$$P\{\|T - P_{R_n^{\mathrm{rand}}} T\| \leq \Delta(n_t, \varepsilon_{\mathrm{testfail}})\} \geq (1 - \varepsilon_{\mathrm{testfail}}), \tag{6.47}$$

where $c_{\mathrm{est}}(n_t, \varepsilon_{\mathrm{testfail}}) := 1/[(2\lambda_{\min}^{\mathbb{M}_S})^{1/2} \, \mathrm{erf}^{-1}(\sqrt[n]{\varepsilon_{\mathrm{testfail}}})]$. *Additionally, we have*

$$P\left\{ \frac{\Delta(n_t, \varepsilon_{\mathrm{testfail}})}{\|T - P_{R_n^{\mathrm{rand}}} T\|} \leq c_{\mathrm{eff}}(n_t, \varepsilon_{\mathrm{testfail}}) \right\} \geq 1 - \varepsilon_{\mathrm{testfail}},$$

where the constant $c_{\mathrm{eff}}(n_t, \varepsilon_{\mathrm{testfail}})$ *is defined as*

$$c_{\mathrm{eff}}(n_t, \varepsilon_{\mathrm{testfail}}) := \left[Q^{-1}\left(\frac{N_T}{2}, \frac{\varepsilon_{\mathrm{testfail}}}{n_t} \right) \frac{\lambda_{\max}^{\mathbb{M}_S}}{\lambda_{\min}^{\mathbb{M}_s}} (\mathrm{erf}^{-1}(\sqrt[n]{\varepsilon_{\mathrm{testfail}}}))^{-2} \right]^{1/2}$$

and Q^{-1} *is the inverse of the upper normalized incomplete gamma function.*

The constant $c_{\mathrm{est}}(n_t, \varepsilon_{\mathrm{testfail}})$ is calculated on line 3 using N_T, which denotes the rank of operator T. In practice N_T is unknown and an upper bound for N_T such as $\min(N_S, N_R)$ can be used instead. Note that the term $(\max_{t \in M} \|t\|_R) \cdot c_{\mathrm{est}}(n_t, \varepsilon_{\mathrm{testfail}})$ is the norm estimator (6.46). The test vectors are reused for all iterations.

To finally analyze the failure probability of Algorithm 6.1 we first note that after N_T steps we have $R_n^{\mathrm{rand}} = \mathrm{range}(T)$ and thus $\|T - P_{R_n^{\mathrm{rand}}} T\| = 0$, yielding the termination of Algorithm 6.1. Using the fact that the a posteriori error estimator defined in (6.46) is therefore executed at most N_T times combined with the probability for one estimate to fail in (6.47) and a union bound argument we infer that the failure probability for the whole algorithm is $\varepsilon_{\mathrm{algofail}} \leq N_T \varepsilon_{\mathrm{testfail}}$.

Remarkably, the convergence behavior of the reduced space R_n^{rand} is only slightly worse than the rate $\sqrt{\lambda_{n+1}}$, which is achieved by the optimal local approximation spaces defined in Theorem 6.11.

Proposition 6.15 (A priori error bound [21]). *Let* $\lambda_{\max}^{\mathbb{M}_R}$ *and* $\lambda_{\min}^{\mathbb{M}_R}$ *denote the largest and smallest eigenvalues of the inner product matrix* $\mathbb{M}_R$ *and let* R_n^{rand} *be the outcome of Algorithm* 6.1. *Then, for* $n \geq 4$ *we have*

$$\mathbb{E}\|T - P_{R_n^{\mathrm{rand}}} T\| \leq C_{R,S} \min_{\substack{k+p=n \\ k \geq 2, p \geq 2}} \left[\left(1 + \sqrt{\frac{k}{p-1}}\right) \sqrt{\lambda_{k+1}} + \frac{e\sqrt{n}}{p} \left(\sum_{j>k} \lambda_j\right)^{\frac{1}{2}} \right], \tag{6.48}$$

where $C_{R,S} = (\lambda_{\max}^{\mathbb{M}_R}/\lambda_{\min}^{\mathbb{M}_R})^{1/2} (\lambda_{\max}^{\mathbb{M}_S} \lambda_{\min}^{\mathbb{M}_S})^{1/2}$.

It can be observed in numerical experiments that the a priori bound in Proposition 6.15 is sharp in terms of the predicted convergence behavior as we will show now for a test case from [21]. Moreover, we will investigate the performance of Algorithm 6.1 also for a test case from [21]. To that end, let $\widehat{\Omega}_m = (-0.5, 0.5) \times (-0.25, 0.25) \times (-0.5, 0.5)$ and $\Omega_m = (-0.5, 0.5) \times (-0.5, 0.5) \times (-0.5, 0.5)$ be the subdomains on which we aim

to construct a local approximation space, $\widehat{\Omega}_{\text{out}} = (-2, 2) \times (-0.25, 0.25) \times (-2, 2)$ and $\Omega_{\text{out}} = (-2, 2) \times (-0.5, 0.5) \times (-2, 2)$ the corresponding oversampling domains, and $\widehat{\Gamma}_{\text{out}} = \{-2, 2\} \times (-0.25, 0.25) \times (-2, 2) \cup (-2, 2) \times (-0.25, 0.25) \times \{-2, 2\}$ and $\Gamma_{\text{out}} = \{-2, 2\} \times (-0.5, 0.5) \times (-2, 2) \cup (-2, 2) \times (-0.5, 0.5) \times \{-2, 2\}$ the respective outer boundaries. On $\partial\widehat{\Omega}_{\text{out}} \setminus \widehat{\Gamma}_{\text{out}}$ and $\partial\Omega_{\text{out}} \setminus \Gamma_{\text{out}}$ we prescribe homogeneous Neumann boundary conditions and we suppose that $\widehat{\Omega}_{\text{out}}$ and Ω_{out} do not border the Dirichlet boundary of Ω. For the FE discretization we use a regular mesh with hexahedral elements and a mesh size $h = 0.1$ in each space direction and a corresponding conforming FE space with linear FE resulting in $\dim(V_h|_{\widehat{\Omega}_{\text{out}}}) = 30{,}258$, $\dim(R) = N_R = 2{,}172$, $\dim(S) = N_S = 2{,}880$ for $\widehat{\Omega}_{\text{out}}$ and $V_h|_{\Omega_{\text{out}}} = 55{,}473$, $N_R = 3{,}987$, and $N_S = 5{,}280$ for Ω_{out}.[3] We equip the source space S with the L^2-inner product and the range space R with the energy inner product. Finally, for all results in this subsection we computed the statistics over 1,000 samples.

Analyzing the convergence behavior of $\mathbb{E}(\|T - P_{R_{k+p}^{\text{rand}}} T\|)$ on $\widehat{\Omega}_{\text{out}}$ for a growing number of randomly generated basis functions k and a (fixed) oversampling parameter $p = 2$ in Figure 6.4a we see that until $k \approx 75$ the a priori bound reproduces the convergence behavior of $\mathbb{E}(\|T - P_{R_{k+p}^{\text{rand}}} T\|)$ perfectly. We may thus conclude that the a priori bound in (6.48) seems to be sharp regarding the convergence behavior of $\mathbb{E}(\|T - P_{R_{k+p}^{\text{rand}}} T\|)$ in the basis size k. We also observe that the a priori bound is rather pessimistic as it overestimates $\mathbb{E}(\|T - P_{R_{k+p}^{\text{rand}}} T\|)$ by a factor of more than 100; this is mainly due to the square root of the conditions of the inner product matrices.

Regarding the performance of Algorithm 6.1 on Ω_{out} we first observe in Figure 6.4b that the actual error $\|T - P_{R_n^{\text{rand}}} T\|$ lies below the target tolerance tol for all 1,000 samples for $n_t = 10$, which holds also true for all other considered values of n_t. Here, we prescribe $\varepsilon_{\text{algofail}} = 10^{-10}$ and use 3,993 as an upper bound for N_T. We see in Figure 6.4b that increasing the number of test vectors n_t from 5 to 10 or from 10 to 20 increases the ratio between the median of the actual error $\|T - P_{R_n^{\text{rand}}} T\|$ and the target accuracy tol significantly – for the former by more than one magnitude – while an increase from $n_t = 40$ to $n_t = 80$ has hardly any influence; similar results have been obtained in [21] for heat conduction and a Helmholtz problem. This can be explained by the scaling of the effectivity of the employed a posteriori error estimator, which is of the order of 1,000 for $n_t = 5$ and of the order of 10 for $n_t \geq 20$. Regarding the choice of n_t it seems that for the present test case a value of about 20 is in the sweet spot. We thus infer that for the present test case only very few local solutions in addition to the optimal amount are required, demonstrating that Algorithm 6.1 performs nearly optimally in terms of computational complexity for the current problem.

---

3 Note that although in theory we should subtract the orthogonal projection on the six rigid body motions from the FE basis functions, in actual practice we avoid that by subtracting the orthogonal projection from the harmonic extensions only.

(a) Convergence behavior on $\widehat{\Omega}_{\text{out}}$

(b) Median for varying n_t on Ω_{out}
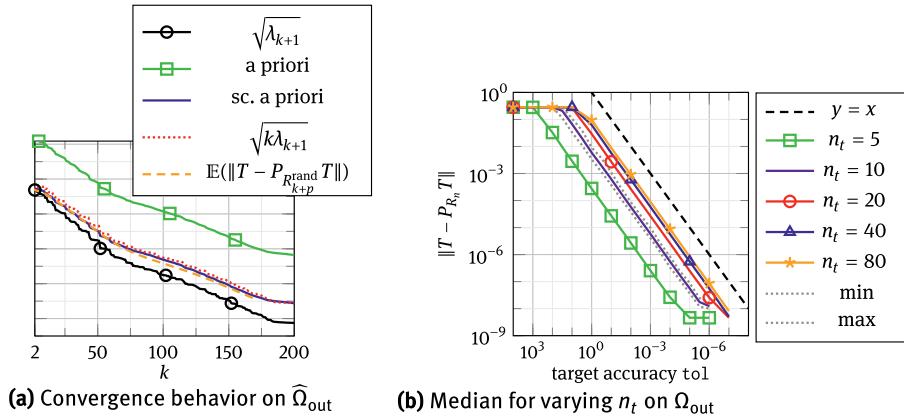
Figure 6.4: (a) Comparison of the convergence behavior of $\sqrt{\lambda_{k+1}}$, $\sqrt{k\lambda_{k+1}}$, $\mathbb{E}(\|T - P_{R^{k+p}}T\|)$, the a priori error bound (6.48), and the a priori error bound of (6.48) scaled with a constant such that its value for $k = 2$ equals the one of $\mathbb{E}(\|T - P_{R^{\text{rand}}_{k+p}}T\|)$ (sc. a priori) for increasing k for and $p = 2$ for the oversampling domain $\widehat{\Omega}_{\text{out}}$. (b) Median of the projection error $\|T - P_{R^{\text{rand}}_n}T\|$ for a decreasing target accuracy tol for a varying number of test vectors n_t and the minimal and maximal values for $n_t = 10$ on Ω_{out}.

6.4.2.3 The general setting $\mathcal{P} \neq \{\bar{\boldsymbol{\mu}}\}$

The processes in Sections 6.4.2.1 and 6.4.2.2 yield for every $\boldsymbol{\mu} \in \mathcal{P}$ the local approximation space $R_n^+(\boldsymbol{\mu})$ for this specific parameter $\boldsymbol{\mu} \in \mathcal{P}$; $R_n^+(\boldsymbol{\mu})$ can also be generated by some other process, where we require that

$$\|T(\boldsymbol{\mu}) - P_{R_n(\boldsymbol{\mu})}T(\boldsymbol{\mu})\| \le \frac{\varepsilon}{2C_1(\mathcal{T}_H, \boldsymbol{\mu})} \tag{6.49}$$

possibly only at high probability and that $R_n^+(\boldsymbol{\mu})$ is defined as the direct sum of $R_n(\boldsymbol{\mu})$, the kernel of the bilinear form, and representations of nonhomogeneous Dirichlet boundary conditions and the right-hand side. We abuse notation in this subsection by omitting henceforth the remark that the estimate may only hold in a probabilistic sense. The constant $C_1(\mathcal{T}_H, \boldsymbol{\mu})$ has to be chosen in such a manner that if one uses the parameter-dependent spaces $R_n^+(\boldsymbol{\mu})$ to define $u^n(\boldsymbol{\mu})$, we have

$$\frac{\|\|u(\boldsymbol{\mu}) - u_{n_+}(\boldsymbol{\mu})\|\|_{\boldsymbol{\mu}}}{\|\|u(\boldsymbol{\mu})\|\|_{\boldsymbol{\mu}}} \le \frac{\varepsilon}{2}. \tag{6.50}$$

The spectral greedy algorithm as introduced in [103][4] constructs *one* (quasi-optimal) parameter-independent approximation space R_N which approximates those

---

4 For a generalization to a setting where the discrete parameter set describes different geometries such as a beam with or without a crack we refer to [101].

parameter-dependent spaces $R_n^+(\boldsymbol{\mu})$ with a given accuracy on a finite-dimensional training set $\Xi \subset \mathcal{P}$. In the spectral greedy algorithm we exploit the fact that we expect that the local spaces $R_n^+(\boldsymbol{\mu})$, and in particular the spectral modes that correspond to the largest eigenvalues, are not affected too much by a variation in the parameter thanks to the expected very rapid decay of the higher eigenfunctions in the interior of Ω_{out}.

The spectral greedy as described in Algorithm 6.2 then proceeds as follows. After the initialization we compute for all $\boldsymbol{\mu} \in \Xi$ the parameter-dependent spaces $R_n^+(\boldsymbol{\mu})$ such that we have (6.49). Note that for a decomposition $\mathcal{T}_H$ with mutually disjoint interfaces (also called ports), where each Ω_m, $m = 1, \ldots, M$, has exactly two interfaces and all interfaces have the same geometry, we have the following a priori error bound [103] for the error between $u(\boldsymbol{\mu})$ and the continuous port-reduced static condensation approximation $u_{n_+}(\boldsymbol{\mu})$ corresponding to the *parameter-dependent* optimal interface space $\mathcal{R}_n^+(\boldsymbol{\mu})$:

$$\frac{\|u(\boldsymbol{\mu}) - u_{n_+}(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}}{\|u(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}} \leq \#\gamma c_1(\boldsymbol{\mu}) c_2(\boldsymbol{\mu}) \max_{\gamma \in \mathcal{T}_H^\gamma} \left(C_{\gamma,1}(\Omega_\gamma, \boldsymbol{\mu}) \sqrt{\lambda_{\gamma, n+1}(\boldsymbol{\mu})} \right). \tag{6.51}$$

Here, the constant $C_{\gamma,1}(\Omega_\gamma, \boldsymbol{\mu})$ depends only on the subdomains that share γ and not on Ω or on $u(\boldsymbol{\mu})$. Moreover, $c_1(\boldsymbol{\mu})$ and $c_2(\boldsymbol{\mu})$ are chosen such that we have $c_1(\boldsymbol{\mu}) \|\cdot\|_{\bar{\boldsymbol{\mu}}} \leq \|\cdot\|_{\boldsymbol{\mu}} \leq c_2(\boldsymbol{\mu}) \|\cdot\|_{\bar{\boldsymbol{\mu}}}$ for all $\boldsymbol{\mu} \in \mathcal{P}$ and a fixed reference parameter $\bar{\boldsymbol{\mu}} \in \mathcal{P}$. Choosing $C_1(\mathcal{T}_H, \boldsymbol{\mu}) =$

Algorithm 6.2: Spectral greedy [103].

> **Input** : train sample $\Xi \subset \mathcal{P}$, tolerance ε
>
> **Output:** set of chosen parameters Ξ_N, local approximation space R_N

1 **Initialize** $N \leftarrow \dim(\ker(a_m(\cdot, v)))$,
  $\Xi_N \leftarrow \emptyset, R_N \leftarrow \ker(a_m(\cdot, v))$ or $R_N \leftarrow \ker(a_m(\cdot, v))|_{\Gamma_{m,m'}}$

2 **foreach** $\boldsymbol{\mu} \in \Xi$ **do**

3 $\quad$ Compute $R_n^+(\boldsymbol{\mu})$ such that $\|T(\boldsymbol{\mu}) - P_{R_n^+(\boldsymbol{\mu})} T(\boldsymbol{\mu})\| \leq \frac{\varepsilon}{2C_1(\mathcal{T}_H, \boldsymbol{\mu})}$.

4 **while** *true* **do**

5 $\quad$ **if** $\max_{\boldsymbol{\mu} \in \Xi} E(S(R_n^+(\boldsymbol{\mu})), R_N) \leq \varepsilon/(\varepsilon + 2C_2(\mathcal{T}_H, \boldsymbol{\mu}) c_1(\boldsymbol{\mu}) c_2(\boldsymbol{\mu}))$ **then**

6 $\quad\quad$ **return**

7 $\quad$ $\boldsymbol{\mu}^* \leftarrow \arg\max_{\boldsymbol{\mu} \in \Xi} E(S(R_n^+(\boldsymbol{\mu})), R_N)$

8 $\quad$ $\Xi_{N+1} \leftarrow \Xi_N \cup \boldsymbol{\mu}^*$

9 $\quad$ $\kappa \leftarrow \arg\sup_{\rho \in S(R_n^+(\boldsymbol{\mu}^*))} \inf_{\zeta \in R_N} \|\rho - \zeta\|_R$

10 $\quad$ $R_{N+1} \leftarrow R_N + \text{span}\{\kappa\}$

11 $\quad$ $N \leftarrow N + 1$

12 **return** Ξ_N, R_N

$\#\gamma c_1(\boldsymbol{\mu})c_2(\boldsymbol{\mu}) \max_{\gamma \in \mathcal{T}_H^\gamma} C_{\gamma,1}(\Omega_\gamma, \boldsymbol{\mu})$ and $\sqrt{\lambda_{\gamma,n+1}(\boldsymbol{\mu})} \leq \varepsilon/2$ yields a reduced space $R_n^+(\boldsymbol{\mu})$ that satisfies the requirements stated in the beginning for every $\boldsymbol{\mu} \in \Xi$. Although precise estimates for $C_{\gamma,1}(\Omega_\gamma, \boldsymbol{\mu})$ can be obtained, setting $C_{\gamma,1}(\Omega_\gamma, \boldsymbol{\mu}) = 1$ yields in general good results as another value would just result in rescaling ε; for further details see [103]. After having collected all functions on $\Gamma_{m,m'}$ or Ω_m that are essential to obtain a good approximation for all local solutions $u_{\mathrm{loc}}(\boldsymbol{\mu})$ of the PDE evaluated on $\Gamma_{m,m'}$ or Ω_m, $\boldsymbol{\mu} \in \Xi$, we must select a suitable basis from those functions. This is realized in an iterative manner on lines 5–14.

In each iteration we first identify on line 7 the reduced space $R_n^+(\boldsymbol{\mu}^*)$ that maximizes the deviation

$$E(S(R_n^+(\boldsymbol{\mu})), R_N) := \sup_{\xi \in S(R_n^+(\boldsymbol{\mu}))} \inf_{\zeta \in R_N} \|\xi - \zeta\|_R, \quad \boldsymbol{\mu} \in \Xi,$$

where possible choices of $S(R_n^+(\boldsymbol{\mu})) \subset R_n^+(\boldsymbol{\mu})$ will be discussed below. Subsequently, we determine on line 9 the function $\kappa \in S(R_n^+(\boldsymbol{\mu}^*))$ that is worst approximated by the space R_N and enhance R_N with the span of κ. The spectral greedy algorithm terminates if for all $\boldsymbol{\mu} \in \Xi$ we have

$$\max_{\boldsymbol{\mu} \in \Xi} E(S(R_n^+(\boldsymbol{\mu})), R_N) \leq \varepsilon/(\varepsilon + 2C_2(\mathcal{T}_H, \boldsymbol{\mu})c_1(\boldsymbol{\mu})c_2(\boldsymbol{\mu})) \tag{6.52}$$

for a constant $C_2(\mathcal{T}_H, \boldsymbol{\mu})$, which can in general be chosen equal to one. We emphasize that both $C_1(\mathcal{T}_H, \boldsymbol{\mu})$ and $C_2(\mathcal{T}_H, \boldsymbol{\mu})$ do in general only depend on the *number* of faces or subspaces on which the respective reduced space R_N is used and *not on the precise decomposition* of Ω; see (6.51). A slight modification of the stopping criterion (6.52) and a different scaling of ε in the threshold for the a priori error bound on line 3 allows to prove that after termination of the spectral greedy for a decomposition $\mathcal{T}_H$ with mutually disjoint interfaces, where each Ω_m, $m = 1, \ldots, M$, has exactly two interfaces and all interfaces have the same geometry, we have [103]

$$\|u(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_{\boldsymbol{\mu}} / \|u(\boldsymbol{\mu})\|_{\boldsymbol{\mu}} \leq \varepsilon. \tag{6.53}$$

Here, $u_N(\boldsymbol{\mu})$ is the continuous port-reduced static condensation approximation corresponding to $\mathcal{R}_N$, $\mathcal{R}_N$ being the continuous outcome of the spectral greedy.

Choice of the subset $S(R_n^+(\boldsymbol{\mu}))$

First, we emphasize that in contrast to the standard greedy as introduced in [111] we have an ordering of the basis functions in $R_n^+(\boldsymbol{\mu})$ in terms of their approximation properties thanks to the transfer eigenvalue problem; the sorting of the basis functions in terms of their approximation properties is implicitly saved in their norms as $\|\chi_j^{sp}(\boldsymbol{\mu})\|_R^2 = \lambda_j(\boldsymbol{\mu})$, $j = 1, \ldots, n$. To obtain local approximation spaces R_N that yield a (very) good approximation $u^N(\boldsymbol{\mu})$ already for moderate N it is therefore desirable that

the spectral greedy algorithm selects the lower eigenmodes sooner rather than later during the while loop. As suggested in [103] we thus propose to consider

$$S(R_n^+(\boldsymbol{\mu})) := \{\zeta(\boldsymbol{\mu}) \in R_n^+(\boldsymbol{\mu}) : \|\zeta(\boldsymbol{\mu})\|_{R_n^+(\boldsymbol{\mu})} \leq 1\} \tag{6.54}$$

$$\text{with } \|\zeta(\boldsymbol{\mu})\|_{R_n^+(\boldsymbol{\mu})} := \left(\sum_{i=1}^{n_+}(\zeta_i(\boldsymbol{\mu}))^2\right)^{1/2},$$

where $\zeta(\boldsymbol{\mu}) = \sum_{i=1}^{n_+} \zeta_i(\boldsymbol{\mu})\chi_i(\boldsymbol{\mu})$, $n_+ := \dim(R_n^+(\boldsymbol{\mu}))$, and here and henceforth $\{\chi_i(\boldsymbol{\mu})\}_{i=1}^{n_+}$ denotes the orthonormal basis of $R_n^+(\boldsymbol{\mu})$. Note that we are therefore considering a weighted norm in $R_n^+(\boldsymbol{\mu})$. The deviation $E(S(R_n^+(\boldsymbol{\mu})), R_N)$ can then be computed by solving the following eigenvalue problem: Find $(\varrho_j(\boldsymbol{\mu}), \sigma_j(\boldsymbol{\mu})) \in (\mathbb{R}^{n_+}, \mathbb{R}^+)$ such that

$$\mathbb{Z}(\boldsymbol{\mu})\varrho_j(\boldsymbol{\mu}) = \sigma_j(\boldsymbol{\mu})\varrho_j(\boldsymbol{\mu}),$$

$$\text{where } \mathbb{Z}_{i,l}(\boldsymbol{\mu}) := \left(\chi_l(\boldsymbol{\mu}) - \sum_{k=1}^{N}(\chi_l(\boldsymbol{\mu}), \chi_k)_R \chi_k, \chi_i(\boldsymbol{\mu}) - \sum_{k=1}^{N}(\chi_i(\boldsymbol{\mu}), \chi_k)_R \chi_k\right)_R$$

and χ_k denotes the orthonormal basis of R_N. We thus obtain $E(S(R_n^+(\boldsymbol{\mu})), R_N) = \sqrt{\sigma_1(\boldsymbol{\mu})}$, for all $\boldsymbol{\mu} \in \Xi$, and $\kappa = \sum_{i=1}^{n_+} \varrho_1(\boldsymbol{\mu}^*)\chi_i(\boldsymbol{\mu}^*)$ at each iteration.

Note that were we to consider the norm $\| \cdot \|_R$ in (6.54) the sorting of the spectral basis $\chi_i(\boldsymbol{\mu})$ of $R_n^+(\boldsymbol{\mu})$ in terms of approximation properties is neglected in the while loop of Algorithm 6.2; for further explanations see [103].

Finally, we compare in Figure 6.5 the spectral modes generated by the spectral greedy algorithm, Algorithm 6.2, numerically with other interface modes, demonstrating the superior convergence of the former. In detail, we compare the relative error of the port-reduced static condensation approximation for interface spaces comprising "Legendre polynomial"-type functions[5] [37], empirical port modes constructed by a pairwise training algorithm[6] [37, 38], and the spectral modes. To that end, we consider
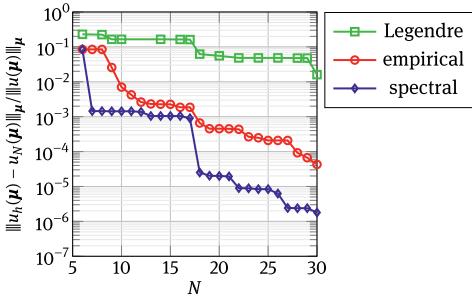
Figure 6.5: $\|u_h(\boldsymbol{\mu}) - u^N(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}/\|u(\boldsymbol{\mu})\|_{\boldsymbol{\mu}}$ for the Legendre, empirical, and spectral interface basis functions for the solid beam.

---

5 Note that each component of the displacement is the solution of a scalar singular Sturm–Liouville eigenproblem.
6 Following the notation in [38] we have chosen $N_{\text{samples}} = 500$ and $\gamma = 3$ in the pairwise training algorithm.

a domain Ω which consists of two identical solid beams, each of whom is associated with a subdomain Ω_i, $i = 1, 2$. Here, we choose $\Omega_1 = (-0.5, 0.5) \times (-0.5, 0.5) \times (0, 5)$, $\Omega_2 = (-0.5, 0.5) \times (-0.5, 0.5) \times (5, 10)$, and $\Gamma_{out} = \Gamma_1 \cup \Gamma_2$, with $\Gamma_1 = (-0.5, 0.5) \times (-0.5, 0.5) \times \{0\}$ and $\Gamma_2 = (-0.5, 0.5) \times (-0.5, 0.5) \times \{10\}$. The underlying FE discretization has $N = 3,348$ degrees of freedom per subdomain and $N_\Gamma = 108$ degrees of freedom per interface. We require $E(\boldsymbol{\mu})$ to be uniform within each subdomain, the constant varying in $[1, 10]$, and choose for $\boldsymbol{G}(\boldsymbol{\mu}) \in \mathbb{R}^3$ the admissible set of parameters to be $[-1, 1] \times [-1, 1] \times [-1, 1]$. Finally, we equip both S and R again with a lifting inner product. Within the spectral greedy we have considered 200 parameter values sampled from the uniform distribution over $\mathcal{P}$ and $\varepsilon = 1 \cdot 10^{-6}$. On average the interface spaces $R_n^+(\boldsymbol{\mu})$ have a size of 13.65 and the resulting parameter-independent port space R_N has a size of 56.

In the online stage we consider $E(\boldsymbol{\mu}) \equiv 1$ in both components, $\boldsymbol{G} = (0, 0, 0)^T$, and prescribe $\boldsymbol{g}_{D,1} = (0, 0, 0)^T$ at Γ_1 and $\boldsymbol{g}_{D,2} = (1, 1, 1)^T$ at Γ_2. We observe that the Legendre modes perform by far the worst, demonstrating that including information on the solution manifold in the basis construction procedure can significantly improve the approximation behavior. We remark that the Legendre modes will perform even worse in the case of less regular behavior on the interface, which further justifies the need for problem-specific local approximation spaces in the sense of model reduction. The empirical modes and spectral modes exhibit a comparable convergence until $N = 17$, but for $N > 17$ the relative error in the spectral approximation is one order of magnitude smaller than that of the empirical port mode approximation. This can be explained by the fact that thanks to its conception the pairwise training algorithm is able to identify and include the most significant modes, but (in contrast to the spectral greedy algorithm) might have difficulties to detect subtle modes that affect the shape of the function at the interface $\Gamma_{m,m'}$ only slightly. Note that the temporary stagnation of the relative error for $N = 7, \ldots, 17$ for the spectral modes is due to the fact that the spectral greedy prepares the interface space for all possible boundary conditions and parameter configurations. Thus, for the boundary conditions considered here some spectral modes, as, say, a mode related to a twisting (torsion) of the beam, are not needed for the approximation.

6.5 A posteriori error estimation

6.5.1 Residual-based a posteriori error estimation

A global residual-based a posteriori error estimator for projection-based model reduction is readily defined as

$$\Delta(u_N(\boldsymbol{\mu})) := \frac{1}{\alpha(\boldsymbol{\mu})} \|R(u_N(\boldsymbol{\mu}); \boldsymbol{\mu})\|_{V_h'}, \tag{6.55}$$

where $R(u_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \in V_h'$ is the global residual given as $\langle R(u_N(\boldsymbol{\mu}); \boldsymbol{\mu}), \varphi_h \rangle = f_h(\varphi_h; \boldsymbol{\mu}) - a_h(u_N(\boldsymbol{\mu}), \varphi_h; \boldsymbol{\mu})$ for all $\varphi_h \in V_h$. This error estimator is known to be robust and efficient (cf. [50, Proposition 4.4]), i. e., we have

$$\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \Delta(u_N(\boldsymbol{\mu})) \leq \frac{\gamma(\boldsymbol{\mu})}{\alpha(\boldsymbol{\mu})} \|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V. \tag{6.56}$$

For localized model order reduction, however, we are merely interested in localized a posteriori error estimation. To this end, we first present abstract localized lower and upper bounds for the dual norm of a linear functional (see [20]).

Theorem 6.16 (Localized lower and upper bounds for functionals). *Let O_i, $1 \leq i \leq \tilde{M}$, be a collection of linear subspaces of V_h, and let $P_{O_i} : V_h \longrightarrow O_i \subseteq V_h$ be mappings which satisfy $\sum_{i=1}^{\tilde{M}} P_{O_i} = \mathrm{id}_{V_h}$. Moreover, assume that for $J \in \mathbb{N}$ there exists a partition $\dot{\bigcup}_{j=1}^{J} \Upsilon_j = \{1, \ldots, \tilde{M}\}$ such that for arbitrary $1 \leq j \leq J$ and $i_1 \neq i_2 \in \Upsilon_j$ we have $O_{i_1} \perp O_{i_2}$.*

Defining the stability constant of this partition modulo V_N as

$$c_N := \sup_{\varphi \in V_h \setminus \{0\}} \frac{(\sum_{i=1}^{\tilde{M}} \inf_{\tilde{\varphi} \in V_N \cap O_i} \|P_{O_i}(\varphi) - \tilde{\varphi}\|^2)^{\frac{1}{2}}}{\|\varphi\|}, \tag{6.57}$$

we have for any linear functional $f \in V_h'$ with $\langle f, \varphi \rangle = 0 \ \forall \varphi \in V_N$ the estimate

$$\frac{1}{\sqrt{J}} \left(\sum_{i=1}^{\tilde{M}} \|f\|_{O_i'}^2 \right)^{\frac{1}{2}} \leq \|f\|_{V_h'} \leq c_N \cdot \left(\sum_{i=1}^{\tilde{M}} \|f\|_{O_i'}^2 \right)^{\frac{1}{2}}. \tag{6.58}$$

Here, $\|f\|_{O_i'}$ denotes the norm of the restriction of f to O_i.

When grouping the spaces O_i so that in each group, all spaces are orthogonal to each other, J is the number of groups needed. Note that subtracting the projection onto V_N in (6.57) allows subtracting, say, the mean value of a function or the orthogonal projection onto the rigid body motions, if the respective functions are included in V_N. We may thus employ, say, Poincaré's inequality or Korn's inequality in subdomains that do not lie at Γ_D.

Applying both estimates to the residual $R(u_N(\boldsymbol{\mu}); \boldsymbol{\mu}) \in V_h'$, we obtain from (6.55) and Theorem 6.16 a robust and efficient, localized error estimate.

Corollary 6.17 (Localized residual-based a posteriori error estimate). *Let the assumptions on the subspace collection O_i and the mappings P_i from Theorem 6.16 be satisfied. Then, the error estimator $\Delta_{\mathrm{loc}}(u_N(\boldsymbol{\mu}))$ defined as*

$$\Delta_{\mathrm{loc}}(u_N(\boldsymbol{\mu})) := \frac{1}{\alpha(\boldsymbol{\mu})} c_N \left(\sum_{i=1}^{\tilde{M}} \|R(u_N(\boldsymbol{\mu}); \boldsymbol{\mu})\|_{O_i'}^2 \right)^{\frac{1}{2}} \tag{6.59}$$

is robust and efficient, i. e.,

$$\|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V \leq \Delta_{\mathrm{loc}}(u_N(\boldsymbol{\mu})) \leq \frac{\gamma(\boldsymbol{\mu}) \sqrt{J} c_N}{\alpha(\boldsymbol{\mu})} \|u_h(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})\|_V. \tag{6.60}$$

Online-offline decomposition of this error estimator can be done by applying the usual strategy for online-offline decomposition used with the standard reduced basis error estimator (see, e. g., [50, Section 4.2.5] or a numerically more stable approach [19, 25, 105]) to every dual norm in $\Delta_{\mathrm{loc}}(u_N(\boldsymbol{\mu}))$.

The a posteriori error estimator for the ArbiLoMod derived in [20] and the a posteriori error estimator for the scRBE method as suggested in [100] both fit into the framework above, as will be detailed below in Examples 6.18 and 6.19. In contrast, for instance the error estimators proposed in [61, 60] for the scRBE method exploit matrix perturbation analysis at the system level to bound the Euclidean norm of the error between the coefficients of the static condensation solution and the coefficients of the static condensation solution using a reduced basis approximation in the interior. To estimate the error caused by interface reduction in [37] a computationally tractable nonconforming approximation to the exact error is employed. To take into account the error due to the intra-element reduced basis approximations ideas from [61] are used. It can also be noted that the error estimators in [61, 60, 37] are only valid under certain assumptions on the accuracy of the reduced basis approximation. In [83] a localized a posteriori error estimator for interface reduction and intra-element reduced basis approximation is presented for the coupled Stokes–Darcy system. The a posteriori error estimator for the CMS method derived in [64] employs the dual norms of residuals and eigenvalues of the eigenproblems used for the construction of the (local) basis functions. The error estimator in [64] is however only partially local as it involves the residual for the port or interface space on the whole interface Γ. For localized a posteriori error estimation in the context of adaptive GMsFEM we refer to [31, 30, 28, 29].

Example 6.18 (Localized a posteriori error estimate for ArbiLoMod [20]). Let us assume $V_h \subset V = H_0^1(\Omega)$ and choose O_i as subspaces of $H^1(\Omega_i)$, where $\tilde{\mathcal{T}}_H := \{\tilde{\Omega}_1, \ldots, \tilde{\Omega}_{\tilde{M}}\}$ is an arbitrary *overlapping* decomposition of Ω, which may be chosen independently from $\mathcal{T}_H$. Assume that there is a partition of unity $p_i \in H^{1,\infty}(\tilde{\Omega}_i) \cap C(\tilde{\Omega}_i)$, $\sum_{i=1}^{\tilde{M}} p_i = 1$, such that $\|p_i\|_\infty \leq 1$ and $\|\nabla p_i\|_\infty \leq c_{\mathrm{pu}} \operatorname{diam}(\tilde{\Omega}_i)^{-1}$. The constant c_{pu} will depend on the size of the overlap of the subdomains $\tilde{\Omega}_i$ with their neighbors in relation to their diameters.

Moreover, we assume that there is a linear interpolation operator $\mathcal{I}$ onto V_h such that $\mathcal{I}$ is the identity on V_h with $\mathcal{I}(p_i V_h) \subseteq O_i$ and $\|\mathcal{I}(p_i v_h) - p_i v_h\|_V \leq c_I \|v_h\|_{\tilde{\Omega}_i, 1}$ for all $v_h \in V_h$. We then can define mappings

$$P_{O_i}(v_h) := \mathcal{I}(p_i \cdot v_h),$$

which satisfy the assumptions of Theorem 6.16. In case V_h comes from an FE discretization, a possible choice for $\mathcal{I}$ is Lagrange interpolation.

If we now ensure that the partition of unity p_i is included in V_N, we can choose $\tilde{\varphi}$ in the definition of c_N as $\tilde{\varphi} := p_i \cdot |\tilde{\Omega}_i|^{-1} \int_{\tilde{\Omega}_i} \varphi$, which allows us to prove [20, Proposition 5.7]

that c_N can be bounded by

$$c_N \leq \sqrt{4 + 2c_I^2 + 4(c_{\mathrm{pu}}c_{\mathrm{pc}})^2} \cdot \sqrt{c_{\mathrm{ovlp}}}.$$

In this estimate $c_{\mathrm{ovlp}} := \max_{x \in \Omega} \#\{i \in \Upsilon_E \mid x \in \Omega_i\}$ is the maximum number of estimator domains $\tilde{\Omega}_i$ overlapping in any point x of Ω, and c_{pc} is a Poincaré inequality constant associated with $\tilde{\mathcal{T}}_H$. In particular, this result shows that the efficiency of (6.59) is independent of the number of subdomains in $\mathcal{T}_H$, provided that the partition of unity p_i is included in V_N.

Example 6.19 (ScRBE method and interface reduction from [100]). We exemplify the a posteriori error estimator from Corollary 6.17 for the scRBE method, which is equally applicable when considering solely static condensation and no intra-element reduced basis approximations.[7] To simplify notations we define interface spaces $V_h^\gamma := \mathrm{span}\{\psi_1^\gamma, \ldots, \psi_{N_h^\gamma}^\gamma\}$, where $N_h^\gamma = \dim(V_h|_\gamma)$; for the definition of ψ_k^γ we refer to Section 6.3.1. Recall that we then have the following space decomposition of the (global) FE space V_h:

$$V_h = \bigoplus_{m=1}^{M} V_{h;0}^m \oplus \left(\bigoplus_{\gamma \in \mathcal{T}_H^\gamma} V_h^\gamma \right). \tag{6.61}$$

We may thus uniquely rewrite every $\varphi \in V_h$ as

$$\varphi = \sum_{m=1}^{M} \varphi^m + \sum_{\gamma \in \mathcal{T}_H^\gamma} \varphi^\gamma, \tag{6.62}$$

where $\varphi^m \in V_{h;0}^m$ and $\varphi^\gamma \in V_h^\gamma$, extending φ^m and φ^γ by zero. This allows us to define mappings $P_{V_{h;0}^m} : V_h \to V_{h;0}^m, \varphi \mapsto \varphi^m$ and $P_{V_h^\gamma} : V_h \to V_h^\gamma, \varphi \mapsto \varphi^\gamma$, as required in Theorem 6.16. Thanks to (6.21) we also obtain

$$V_{h;0}^m \perp V_{h;0}^{m'}, \quad m \neq m' \quad \text{and} \quad V_{h;0}^m \perp V_h^\gamma, \quad m = 1, \ldots, \Omega, \gamma \in \mathcal{T}_H^\gamma.$$

It thus remains to verify that we can bound the constant c_N with V_N as defined in (6.28). To that end, we first note that thanks to (6.21) we have the following stability result [100, Proposition 4.1]:

$$\|\varphi\|_V^2 = \sum_{m=1}^{M} \|\varphi^m\|_V^2 + \left\| \sum_{\gamma \in \mathcal{T}_H^\gamma} \varphi^\gamma \right\|_V^2. \tag{6.63}$$

---

7 The error estimator in [100] is derived for mutually disjoint interfaces. However, we conjecture that the estimator can be generalized to general decompositions of Ω.

We thus obtain

$$c_N \le \sup_{\varphi \in V_h \setminus \{0\}} \frac{(\sum_{m=1}^{M} \|\varphi_m\|_V^2 + \sum_{\gamma \in \mathcal{T}_H^{\gamma}} \inf_{\tilde{\varphi}_f \in V_N^{\gamma}} \|\varphi^{\gamma} - \tilde{\varphi}^{\gamma}\|_V^2)^{1/2}}{\|\varphi\|_V}$$

$$\overset{(6.63)}{\le} \sup_{\varphi \in V_h \setminus \{0\}} \frac{(\|\varphi\|_V^2 + \sum_{\gamma \in \mathcal{T}_H^{\gamma}} \inf_{\tilde{\varphi}^{\gamma} \in V_N^{\gamma}} \|\varphi^{\gamma} - \tilde{\varphi}^{\gamma}\|_V^2)^{1/2}}{\|\varphi\|_V},$$

where $V_N^{\gamma} := \mathrm{span}\{\psi_1^{\gamma}, \dots, \psi_{N^{\gamma}}^{\gamma}\}$. To show $\sum_{\gamma \in \mathcal{T}_H^{\gamma}} \inf_{\tilde{\varphi}^{\gamma} \in V_N^{\gamma}} \|\varphi^{\gamma} - \tilde{\varphi}^{\gamma}\|_V^2 \le c\|\varphi\|_V$ for a constant c we choose $\tilde{\varphi}^{\gamma}$ such that $(\varphi^{\gamma} - \tilde{\varphi}^{\gamma})|_{\gamma}$ equals the trace of φ minus the orthogonal projection on the kernel of the bilinear form; for further details see [21, 103]. Then, we can use [103, Lemma B.4] to conclude boundedness of c_N and thus (6.60), the latter corresponding to [100, Proposition 4.2 and Corollary 4.6].

Finally, we shortly discuss how to compute the dual norms of the residuals in (6.58). The dual norms of the residuals of the intra-element reduced basis approximations can be computed by employing Riesz representations (see for instance Chapters 1 and 4 of this volume of *Model order reduction*). The dual norms of the residuals in the interface space can be computed by means of conservative fluxes [57], which have been extended to interface reduction in [100]. In detail, we compute the conservative flux $H_N^m(\boldsymbol{\mu})$ such that

$$\sum_{\gamma \in \overline{\Omega}_m} (H_N^m(\boldsymbol{\mu}), \psi^{\gamma})_{\gamma} = f_m(\psi^{\gamma}; \boldsymbol{\mu}) - a_m(u_N(\boldsymbol{\mu}), \psi^{\gamma}; \boldsymbol{\mu}) \quad \forall \psi^{\gamma} \in \bigoplus_{\gamma \in \overline{\Omega}_m} V_h^{\gamma}, \tag{6.64}$$

where $(\cdot, \cdot)_{\gamma}$ denotes a suitable inner product on the interface γ. Note that thanks to our mutual disjoint interface assumption problem (6.64) decouples and we may compute the conservative flux separately for each interface γ. Moreover, by orthonormalizing the interface basis functions χ_k^{γ} defined in Section 6.3.1 with respect to the $(\cdot, \cdot)_{\gamma}$ inner product, the computation of $H_N^m(\boldsymbol{\mu})$ reduces to the assembling of the residual in (6.64). The computational costs thus scale linearly in $(N_h^{\gamma} - N^{\gamma})$ and N^{γ}. For further details we refer to [100].

6.5.2 Local flux reconstruction-based error estimation

Following [90], we discuss local flux reconstruction-based a posteriori error estimation of the full approximation error $u(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu})$ (that is, the discretization as well as the model reduction error) in the context of nonconforming approximations of elliptic multiscale problems such as Example 6.2. An extension to convection–diffusion–reaction problems based on [39] is straightforward. This estimate was introduced in the IP localized nonconforming setting of the LRBMS method (compare Section 6.3.2.2).

Recalling the broken Sobolev space and broken gradient operator from Section 6.3.2.2, the key idea of flux reconstruction-based error estimation is to observe

that not only the approximate solution $u_N(\boldsymbol{\mu})$ is nonconforming, but also the approximate diffusive flux $-\kappa(\boldsymbol{\mu})\nabla_h u_N(\boldsymbol{\mu})$, in the sense that it is not contained in $H_{\mathrm{div}}(\Omega)$ (i. e., the space of functions in $L^2(\Omega)^d$ whose divergence exists in a weak sense and lies in $L^2(\Omega)$).

We may then obtain computable estimates by comparing these quantities with conforming reconstructions, as detailed further below.[8] The respective reconstructed diffusive flux is locally conservative and is related to the conservative flux reconstruction to compute the dual norm of the residuals in the interface space in Example 6.19.

To begin with, we specify the parameter-dependent (semi-)energy norm induced by the bilinear form a for a parameter $\bar{\boldsymbol{\mu}} \in \mathcal{P}$, $\|\!|\cdot|\!\|_{\bar{\boldsymbol{\mu}}} : H^1(\tau_h) \to \mathbb{R}, v \mapsto \|\!|v|\!\|_{\bar{\boldsymbol{\mu}}} := a(v, v; \bar{\boldsymbol{\mu}})^{\frac{1}{2}}$ (by using the broken gradient in the definition of a) and note that we can compare these semi-norms for two parameters by means of the affine decomposition of a (compare (6.69)),

$$\underline{\Theta_a}(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}})^{1/2} \, \|\!|v|\!\|_{\bar{\boldsymbol{\mu}}} \le \|\!|v|\!\|_{\boldsymbol{\mu}} \le \overline{\Theta_a}(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}})^{\frac{1}{2}} \, \|\!|v|\!\|_{\bar{\boldsymbol{\mu}}},$$

with the equivalence constants given by $\underline{\Theta_a}(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}) := \min_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \, \Theta_a^q(\bar{\boldsymbol{\mu}})^{-1}$ and $\overline{\Theta_a}(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}}) := \max_{q=1}^{Q_a} \Theta_a^q(\boldsymbol{\mu}) \, \Theta_a^q(\bar{\boldsymbol{\mu}})^{-1}$, respectively. The first abstract result is the following discretization-agnostic lemma, which leaves the choice of the reconstructions, v and s, open. (We give estimates on the full V_h-norm at the end of this subsection.)

Lemma 6.20 (Abstract energy norm estimate (Lemma 4.1 in [90])). *For $\boldsymbol{\mu} \in \mathcal{P}$, let $u(\boldsymbol{\mu}) \in V$ denote the weak solution of* (6.1) *with the data functions κ and q as in Example* 6.2. *Then for arbitrary $v_N \in H^1(\tau_h)$ and $\bar{\boldsymbol{\mu}} \in \mathcal{P}$, we have*

$$\|\!|u(\boldsymbol{\mu}) - v_N|\!\|_{\bar{\boldsymbol{\mu}}} \le \underline{\Theta_a}(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}})^{-\frac{1}{2}} \left\{ \overline{\Theta_a}(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}})^{\frac{1}{2}} \inf_{v \in V} \|\!|u(\boldsymbol{\mu}) - v|\!\|_{\bar{\boldsymbol{\mu}}} \right.$$
$$+ \inf_{s \in H_{\mathrm{div}}} \left(\sup_{\substack{\varphi \in V \\ |\varphi|_{a;\boldsymbol{\mu}}=1}} \left\{ (q - \nabla \cdot s, \varphi)_{L^2(\Omega)} - (\kappa(\boldsymbol{\mu})\nabla_h v_N + s, \nabla\varphi)_{L^2(\Omega)} \right\} \right) \Bigg\}$$
$$\le \frac{\overline{\Theta_a}(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}})^{\frac{1}{2}}}{\underline{\Theta_a}(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}})} \, 2 \, \|\!|u(\boldsymbol{\mu}) - v_N|\!\|_{\bar{\boldsymbol{\mu}}}.$$

To obtain a fully computable localizable estimate we need to specify the conforming reconstruction of the solution (v in the above lemma) and of the diffusive flux (s in the above lemma). We define both reconstructions with respect to the global fine grid τ_h and note that their respective computations can be localized with respect to the domain decomposition to allow for offline-online decomposable localized estimates.

---

[8] Note that the entire analysis holds for the FOM solution $u_h(\boldsymbol{\mu})$ as well as the ROM solution $u_N(\boldsymbol{\mu})$ (compare [90]), but we restrict the exposition to the latter. In particular, the presented estimates can thus also be used to steer grid adaptation of the FOM solution.

We reconstruct the nonconforming solution $u_h(\boldsymbol{\mu}) \in V_h$ by means of its *Oswald interpolant* $I_{OS}[u_h(\boldsymbol{\mu})] \in V$. We define the corresponding Oswald interpolation operator $I_{OS} : V_h \to V_h \cap V$ by specifying its values on each Lagrange node v of τ_h: Given any $v_h \in V_h$, we set $I_{OS}[v_h](v) := v_h|_t(v)$ for any Lagrange node lying inside a grid element $t \in \tau_h$,

$$I_{OS}[v_h](v) := 0 \quad \text{for all boundary nodes and} \quad I_{OS}[v_h](v) := \frac{1}{|\tau_h^v|} \sum_{t \in \tau_h^v} v_h|_t(v)$$

for all nodes which are shared by multiple grid elements, which we collect in $\tau_h^v \subset \tau_h$.

The definition of the conforming reconstruction of the nonconforming diffusive flux $-\kappa(\boldsymbol{\mu}) \nabla_h u_h(\boldsymbol{\mu}) \in L^2(\Omega)^d$ is more involved. Given $l \geq 0$, we define the *l*-th-order *Raviart–Thomas–Nédélec* space of vector-valued functions by

$$\text{RTN}_h^l(\tau_h) := \{s \in H_{\text{div}}(\Omega) \mid s|_t \in [\mathbb{P}_l(t)]^d + x\mathbb{P}_l(t) \ \forall t \in \tau_h\}$$

and note that the degrees of freedom of any $s_h \in \text{RTN}_h^l(\tau_h)$ are uniquely defined by specifying the moments of order up to $l - 1$ of $s_h|_t$ on all elements $t \in \tau_h$ and the moments of order up to l of $s_h|_\sigma \cdot n_\sigma$ on all faces $\sigma \in \tau_h^\gamma$ (compare [18]). With these preliminaries we define the *diffusive flux reconstruction operator* $R_h^l : \mathcal{P} \to [V_h \to \text{RTN}_h^l(\tau_h)]$, given some $v_h \in V_h$ and some $\boldsymbol{\mu} \in \mathcal{P}$ by specifying the degrees of freedom of $R_h^l[v_h; \boldsymbol{\mu}] \in \text{RTN}_h^l(\tau_h)$, such that

$$(R_h^l[v_h; \boldsymbol{\mu}] \cdot n_\sigma, r)_{L^2(\sigma)} = a_\sigma^c(v_h, r; \boldsymbol{\mu}) + (v_h, r)_\sigma^p \quad \text{for all } r \in \mathbb{P}_l(\sigma) \tag{6.65}$$

on all $\sigma \in \tau_h^\gamma$ and

$$(R_h^l[v_h; \boldsymbol{\mu}], \nabla r)_{L^2(t)} = -a^{CG}(R_h^l[v_h; \boldsymbol{\mu}]|_t, r; \boldsymbol{\mu}) - \sum_{\sigma \in \tau_h^\gamma \cap t} a_\sigma^c(r, v_h; \boldsymbol{\mu}) \tag{6.66}$$

for all $\nabla r \in [\mathbb{P}_{l-1}(t)]^d$ with $r \in \mathbb{P}_l(t)$ on all $t \in \tau_h$. Given a FOM space V_h of polynomial order $k \geq 1$, we choose a $(k-1)$-th order reconstruction. With this definition, the reconstructed diffusive flux of a given reduced solution $u_N(\boldsymbol{\mu})$ fulfills the following *local conservation* property, given that the constant function 1 is present in the local reduced spaces V_N^m:

$$(\nabla \cdot R_h^{k-1}[u_N(\boldsymbol{\mu}); \boldsymbol{\mu}], 1)_{L^2(\Omega_m)} = (q, 1)_{L^2(\Omega_m)}, \quad \text{for all } \Omega_m \in \mathcal{T}_H.$$

When inserting this diffusive flux reconstruction for s in Lemma 6.20, this local conservation property is key to obtaining the following estimate.

Theorem 6.21 (Locally computable energy norm a posteriori estimate). *Let the domain decomposition $\mathcal{T}_H$ from Definition 6.5 be such that the Poincaré inequality holds on each subdomain $\Omega_m \in \mathcal{T}_H$ with a constant $C_P^m > 0$,*

$$\|\varphi - \Pi_0^m \varphi\|_{L^2(\Omega_m)}^2 \leq C_P^m h_m^2 \|\nabla \varphi\|_{L^2(\Omega_m)}^2 \quad \text{for all } \varphi \in H^1(\Omega_m),$$

where $h_m := \mathrm{diam}(\Omega_m)$ and where $\Pi_0^m \varphi$ denotes the mean value of φ over Ω_m. Let further $u(\boldsymbol{\mu}) \in V$ be the weak solution of (6.2) and let $u_N(\boldsymbol{\mu}) \in V_N$ be the IP localized ROM solution, with $1 \in V_N^m$ for $1 \le m \le M$. Then for arbitrary $\bar{\boldsymbol{\mu}}, \hat{\boldsymbol{\mu}} \in \mathcal{P}$, we have

$$|\!|\!| u(\boldsymbol{\mu}) - u_N(\boldsymbol{\mu}) |\!|\!|_{\bar{\boldsymbol{\mu}}} \le \eta(\boldsymbol{\mu}; \bar{\boldsymbol{\mu}}; \hat{\boldsymbol{\mu}})$$

with the a posteriori error estimator $\eta(\boldsymbol{\mu}; \bar{\boldsymbol{\mu}}; \hat{\boldsymbol{\mu}})$ given by

$$\eta(\boldsymbol{\mu}; \bar{\boldsymbol{\mu}}; \hat{\boldsymbol{\mu}}) := \underline{\Theta_a}(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}})^{-\frac{1}{2}} \Bigg[\overline{\Theta_a}(\boldsymbol{\mu}, \bar{\boldsymbol{\mu}})^{\frac{1}{2}} \bigg(\sum_{\Omega_m \in \mathcal{T}_H} \eta_{\mathrm{nc}}^{\Omega_m}(\boldsymbol{\mu}; \bar{\boldsymbol{\mu}})^2 \bigg)^{\frac{1}{2}}$$

$$+ \bigg(\sum_{\Omega_m \in \mathcal{T}_H} \big(\eta_{\mathrm{r}}^{\Omega_m}(\boldsymbol{\mu}) + \underline{\Theta_a}(\boldsymbol{\mu}, \hat{\boldsymbol{\mu}})^{-1} \, \eta_{\mathrm{df}}(\boldsymbol{\mu}; \hat{\boldsymbol{\mu}}) \big)^2 \bigg)^{\frac{1}{2}} \Bigg],$$

and the local nonconformity, residual, and diffusive flux indicators given by

$$\eta_{\mathrm{nc}}^{\Omega_m}(\boldsymbol{\mu}; \bar{\boldsymbol{\mu}}) := |(v_N(\boldsymbol{\mu}) - I_{\mathrm{OS}}[v_N(\boldsymbol{\mu})]|_{\Omega_m}|_{a; \bar{\boldsymbol{\mu}}},$$

$$\eta_{\mathrm{r}}^{\Omega_m}(\boldsymbol{\mu}) := \frac{{C_{\Omega_m}^P}^{\frac{1}{2}}}{\underline{\kappa}_{\Omega_m}} \, \big\| q - \nabla \cdot R_h^{k-1}[u_N(\boldsymbol{\mu}); \boldsymbol{\mu}] \big\|_{L^2(\Omega_m)}, \quad \text{and}$$

$$\eta_{\mathrm{df}}(\boldsymbol{\mu}; \hat{\boldsymbol{\mu}})) := \big\| \kappa(\hat{\boldsymbol{\mu}})^{-1} \big(\kappa(\boldsymbol{\mu}) \nabla_h u_N(\boldsymbol{\mu}) + R_h^{k-1}[u_N(\boldsymbol{\mu}); \boldsymbol{\mu}] \big) \big\|_{L^2(\Omega_m)}, \tag{6.67}$$

respectively, where $\underline{\kappa}_{\Omega_m}$ denotes the minimum eigenvalue of κ over Ω_m and $\mathcal{P}$.

We obtain an a posteriori error estimate with respect to the V_h-norm or a full energy norm, $|\!|\!|\cdot|\!|\!|_{\boldsymbol{\mu}} + (\sum_{\sigma \in \tau_h^\gamma} (\cdot, \cdot)_\sigma^p)^{\frac{1}{2}}$, by noting that $(u(\boldsymbol{\mu}), u(\boldsymbol{\mu}))_\sigma^p = 0$ for a weak solution $u(\boldsymbol{\mu})$ of sufficient regularity.

6.6 Basis enrichment and online adaptivity

Model order reduction is usually employed either (i) in the context of real-time decision making and embedded devices or (ii) in the context of outer loop applications, such as optimal control, inverse problems, or Monte Carlo methods. In (i), one is usually interested in reduced spaces V_N of very low dimension to obtain ROMs as small as possible, at the possible expense of very involved offline computations. Here, localized model order reduction may help to reduce the latter, but we can usually not expect the resulting reduced space to be smaller than the one generated using traditional global model order reduction methods. In (ii), however, one is interested in a black-box-like approximation scheme which is queried for a huge amount of parameters, with a somehow "optimal" computational cost (including offline as well as online cost). Here, one may keep high-dimensional data throughout

the computational process (offline as well as online), and it is in this context that localized model order reduction techniques may truly outperform other approaches. In the context of PDE-constrained optimization this has been investigated, e. g., in [91, 88, 116].

The localized a posteriori error estimation as discussed in Section 6.5 enables adaptive enrichment of the local reduced approximation spaces, whenever the quality of the reduced scheme is estimated to be insufficient – be it due to insufficient training due to lacking computational resources or due to limited knowledge about the range of possible parameters or due to other reasons altogether.

Let us thus assume that an initial (possibly empty) localized reduced approximation space V_N is given, compare Section 6.4. The goal of an adaptive enrichment is to enlarge the local solution spaces with additional modes that reflect nonlocal influences of the true solution such as channeling effects or singularities. Local adaptive basis enrichment can be employed both offline for the whole parameter range and/or online for a specific chosen parameter. Empirical training followed by offline enrichment is, e. g., used in a greedy manner for the basis construction in ArbiLoMod (cf. Example 6.18) in [20]. Adaptive enrichment for the GMsFEM is presented in [31, 30] and online adaptive enrichment in [28, 29]. For the exposition in this section, we restrict ourselves to online enrichment as introduced in [90], i. e., for local enrichment of the basis when a certain parameter is already chosen.

From a bird's-eye perspective, we can think of an online adaptive reduced scheme as a p-adaptive FE scheme with problem-adapted basis functions, where the local reduced bases are adapted during online enrichment.[9] Thus, we can think of online enrichment in the usual Solve → Estimate → Mark → Refine (SEMR) manner, well known in grid-adaptive discretization schemes. In the Estimate step we employ an a posteriori error estimate η that is localizable with respect to the domain decomposition, i. e., $\eta^2 \leq \sum_{m=1}^{M} \eta_m^2$, with appropriate local indicators η_m. Examples are given in Section 6.5. As such, most marking strategies from grid-adaptive schemes are applicable, and we give examples in Section 6.8.1. In this context, refinement is locally done by enriching the local reduced spaces, that is, by adding additional basis functions to the local reduced bases on selected subdomains. We thus presume we are given a parameter $\boldsymbol{\mu} \in \mathcal{P}$ and a reduced solution $u_N(\boldsymbol{\mu}) \in V_N$, the estimated error of which is above a given tolerance.

As an example, we detail the online enrichment procedure used in the context of the LRBMS (compare Section 6.3.2.2), using the a posteriori error estimation techniques from Section 6.5.2. Inspired by domain decomposition as well as numerical multiscale methods, we may then obtain a candidate for the next element of a local

---

9 We would also like to mention the h-adaptive model order reduction approach from [24], which is based on a k-means clustering of the degrees of freedom, but we restrict the exposition here to localization with respect to a domain decomposition.

reduced basis by solving local corrector problems on a collection $\widetilde{\mathcal{T}_H} \subseteq \mathcal{T}_H$ of marked subdomains with $u_N(\boldsymbol{\mu})$ as boundary values. For each marked subdomain $\Omega_m \in \widetilde{\mathcal{T}_H}$, we denote by $\tilde{\Omega}_m := \{\Omega_{m'} \in \mathcal{T}_H \mid \Omega_m \cap \Omega_{m'} \neq \emptyset\}$ an overlapping subdomain and by $V_h^{\tilde{\Omega}_m} := \{v|_{\tilde{\Omega}_m} \mid v \in V_h, v|_{\partial \tilde{\Omega}_m} = 0\}$ the associated restricted FOM space, encoding zero Dirichlet boundary values. We are then looking for a local correction $\varphi^{\tilde{\Omega}_m} \in V_h^{\tilde{\Omega}_m}$, such that

$$a_h(\varphi^{\tilde{\Omega}_m}, v_h; \boldsymbol{\mu}) = f_h(v_h; \boldsymbol{\mu}) - a_h(u_N(\boldsymbol{\mu})|_{\tilde{\Omega}_m}, v_h; \boldsymbol{\mu}) \quad \text{for all } v_h \in V_h^{\tilde{\Omega}_m}, \tag{6.68}$$

where we understand all quantities to be implicitly extended to Ω by zero, if required, and note that $\varphi^{\tilde{\Omega}_m}$ can be computed involving only quantities associated with $\tilde{\Omega}_m$. Using this local correction on the overlapping subdomain, we obtain the next element of the local reduced basis associated with Ω_m by an orthonormalization of $(\varphi^{\tilde{\Omega}_m} + u_N(\boldsymbol{\mu}))|_{\Omega_m}$ with respect to the existing basis on V_N^m.

Given a marking strategy and an orthonormalization procedure, we summarize the adaptive online enrichment used in the context of the LRBMS in Algorithm 6.3.

Algorithm 6.3: Adaptive online enrichment in the context of the LRBMS.

Input : a marking strategy MARK, an orthonormalization procedure ONB, a localizable offline-online decomposable a posteriori error estimate $\eta(\boldsymbol{\mu})^2 \leq \sum_{m=1}^{M} \eta_m(\boldsymbol{\mu})^2$, local reduced bases Φ^m for $1 \leq m \leq M$, $\boldsymbol{\mu} \in \mathcal{P}$, $u_N(\boldsymbol{\mu})$, $\Delta_{\text{online}} > 0$

Output: Updated reduced solution

1   $\Phi^{m(0)} \leftarrow \Phi^m, \forall 1 \leq m \leq M$

2   $n \leftarrow 0$

3   **while** $\eta(\boldsymbol{\mu}) > \Delta_{online}$ **do**

4   **forall** $1 \leq m \leq M$ **do**

5   compute local error indicator $\eta_m(\boldsymbol{\mu})$

6     $\widetilde{\mathcal{T}_H} \leftarrow \text{MARK}(\mathcal{T}_H, \{\eta_m(\boldsymbol{\mu})\}_{1 \leq m \leq M})$

7   **forall** $\Omega_m \in \widetilde{\mathcal{T}_H}$ **do**

8   Solve (6.68) for $\varphi^{\tilde{\Omega}_m}$ $\Phi^{m(n+1)} \leftarrow \text{ONB}(\{\Phi^{m(n)}, (\varphi^{\tilde{\Omega}_m} + u_N(\boldsymbol{\mu}))|_{\Omega_m}\})$

9   update all reduced quantities (system matrices, error estimates) with respect to the newly added basis elements

10   solve (6.13) for the reduced solution $u_N(\boldsymbol{\mu})$ using the updated quantities

11   **return** $u_N(\boldsymbol{\mu})$

6.7 Computational complexity

In this section we discuss the computational efficiency of localized model order reduction schemes in comparison to standard, nonlocalized techniques. Imposing a localization constraint on the reduced space naturally yields suboptimal spaces in the sense of Kolmogorov N-width. However, this is mitigated by the sparse structure of the resulting reduced system matrices. In particular, for problems with high-dimensional parameter domains with localized influence of each parameter component on the solution, we can expect localized ROMs to show comparable or even better online efficiency in comparison to a standard ROM. In addition, localized model order reduction provides more flexibility to balance computational and storage requirements between the offline and online phases and has thus the potential to be optimized with respect to the specific needs. This is particularly favorable for large-scale or multiscale problems, where global snapshot computations are extremely costly or even prohibitive.

In the offline (and enrichment) phase of the localized schemes, only relatively low-dimensional local problems are solved instead of the computation of global solution snapshots. In comparison to a global reduction approach with a parallel solver for snapshot generation (e. g., a domain decomposition scheme), the preparation of the local reduced spaces via training (Section 6.4) can be performed almost communication-free, allowing the application of these schemes on parallel compute architecture without fast interconnect such as cloud environments. Via adaptive enrichment of the approximation spaces – based on the solution of local correction problems (Section 6.6) – smaller and more efficient ROMs can be obtained. In comparison to domain-decomposition methods, where similar correction problems are solved, these correction problems are only solved in regions of the domain where the approximation space is insufficient. Thus, for problems with a localized effect of the parameterization, a significant reduction of the computational effort can be expected in the reduced basis generation process.

In the context of component-based localized model order reduction (e. g., CMS, scRBE, reduced basis hybrid method, RDF) large computational savings can be achieved by the preparation of local approximation spaces (components) with respect to arbitrary neighboring components (connected through so-called ports). In addition to parametric changes of the governing equations or computational domain, this allows the (nonparametric) recombination of components in arbitrary new configurations without requiring additional offline computations.

6.7.1 Online efficiency

In view of Definition 6.8, we can interpret the localized model order reduction methods introduced in Section 6.3 as standard projection-based model reduction methods – such as the reduced basis method – subject to the constraint that the reduced space

V_N admits a localizing decomposition of the form (6.12). As such, the usual offline-online decomposition methodology can be applied. To this end, let us assume that the bilinear form $a(\cdot,\cdot;\boldsymbol{\mu})$ and the source functional $f(\cdot;\boldsymbol{\mu})$ admit affine decompositions

$$a(v,w;\boldsymbol{\mu}) = \sum_{q=1}^{Q_a} \Theta_q^a(\boldsymbol{\mu})a^q(v;w), \quad f(w;\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_q^f(\boldsymbol{\mu})f^q(w), \tag{6.69}$$

for all $v,w \in V$, $\boldsymbol{\mu} \in \mathcal{P}$ with nonparametric bilinear forms $a^q : V \times V \to \mathbb{R}$, functionals $f^q \in V'$, and some parameter functionals $\Theta_q^a, \Theta_q^f : \mathcal{P} \to \mathbb{R}$. If the given problem is not of the form (6.69), we can employ empirical interpolation [13] to compute an approximate affine decomposition.

We begin by computing the reduced approximation space V_N using the methods outlined in Section 6.4. After that, a reduced model is assembled by computing matrix representations $\mathbb{A}^q \in \mathbb{R}^{N \times N}$ of a^q and vector representations $\mathbb{F}^q \in \mathbb{R}^N$ of f^q with respect to a given basis $\varphi_1, \ldots, \varphi_N$ of V_N, i.e.,

$$\mathbb{A}_{ij}^q := a^q(v_j, w_i), \quad \mathbb{F}_i^q := f^q(w_i). \tag{6.70}$$

After this computationally demanding offline phase, the coordinate representation $\mathbb{U}_N(\boldsymbol{\mu}) \in \mathbb{R}^N$ of the reduced solution $u_N(\boldsymbol{\mu})$ of (6.13) is quickly obtained for arbitrary new parameters $\boldsymbol{\mu}$ by solving

$$\sum_{q=1}^{Q_a} \Theta_q^a(\boldsymbol{\mu})\mathbb{A}^q \cdot \mathbb{U}_N(\boldsymbol{\mu}) = \sum_{q=1}^{Q_f} \Theta_q^f(\boldsymbol{\mu})\mathbb{F}^q \tag{6.71}$$

in the following online phase. The computational effort to determine $u_N(\boldsymbol{\mu})$ is of order

$$\mathcal{O}(Q_a N^2 + Q_f N) + \mathcal{O}(N^3) \tag{6.72}$$

for the assembly and solution of the dense equation system (6.71). In particular, we have obtained full offline-online splitting, i.e., the effort to obtain $\mathbb{U}_N(\boldsymbol{\mu})$ is independent of dim V_h. From $\mathbb{U}_N(\boldsymbol{\mu})$ we can then either reconstruct $u_N(\boldsymbol{\mu})$ by linear combination with the reduced basis or evaluate arbitrary linear functionals of $u_N(\boldsymbol{\mu})$ by additionally computing vector representations of these functionals in the offline phase.

Reduced basis methods aim at constructing reduced spaces V_N which are near-optimal approximation spaces for the discrete solution manifold $\{u_h(\boldsymbol{\mu}) \,|\, \boldsymbol{\mu} \in \mathcal{P}\}$ in the sense of Kolmogorov, i.e., we should have

$$\sup_{\boldsymbol{\mu} \in \mathcal{P}} \inf_{v \in V_N} \|u_h(\mu) - v\| \approx d_N := \inf_{\substack{W \subseteq V_h \\ \dim W = N}} \sup_{\boldsymbol{\mu} \in \mathcal{P}} \inf_{v \in W} \|u_h(\mu) - v\|, \tag{6.73}$$

where d_N is the Kolmogorov N-width of the solution manifold. Localized reduced basis methods aim at reducing the computational effort of the offline phase by replacing

the computation of solution snapshots $u_h(\boldsymbol{\mu})$ of the global discrete full-order model by solutions of smaller localized problems associated with the domain $\mathcal{T}_H$ (see below). This comes at the expense of replacing the set of all N-dimensional subspaces of V_h by the smaller set of all N-dimensional subspaces of V_h of the form (6.12), i. e., we aim at constructing V_N with

$$\sup_{\boldsymbol{\mu}\in\mathcal{P}} \inf_{v\in V_N} \|u_h(\mu) - v\| \approx d_N^{\mathrm{loc}} := \inf_{\substack{W\subseteq V_h \\ \dim W=N \\ W \text{ satisf. (6.12)}}} \sup_{\boldsymbol{\mu}\in\mathcal{P}} \inf_{v\in W} \|u_h(\mu) - v\|. \tag{6.74}$$

As $d_N^{\mathrm{loc}} > d_N$, localized reduced basis methods generally result in larger V_N to satisfy a given approximation error tolerance ε. Since we can represent any basis vector of a global reduced basis approximation of (6.11) with respect to the localizing space decomposition (6.10) as a sum of $M^{\mathrm{tot}} := M + \#\mathcal{T}_H^\gamma + \#\mathcal{T}_H^e + \#\mathcal{T}_H^\nu$ local vectors, we have the a priori bound $d_{M^{\mathrm{tot}}\cdot N}^{\mathrm{loc}} < d_N$. In other words, if we denote by N (N^{glob}) the number of reduced basis vectors required for a localized (global) reduced basis approximation for given ε and denoting by N^{loc} the maximum dimension of the local reduced basis spaces V_N^m, V_N^γ, V_N^e, V_N^ν, we have

$$N \le N^{\mathrm{loc}} M^{\mathrm{tot}} \le N^{\mathrm{glob}} M^{\mathrm{tot}} \le C_{\mathcal{T}_H} M N^{\mathrm{glob}}, \tag{6.75}$$

where the constant $C_{\mathcal{T}_H}$ only depends on the topology of the domain decomposition $\mathcal{T}_H$. Whether or not estimate (6.75) is sharp largely depends on the dependence of the solution $u(\boldsymbol{\mu})$ on the parameter $\boldsymbol{\mu}$. When a change in $\boldsymbol{\mu}$ equally affects the solution in all subdomains Ω_m, we expect that optimal local reduced basis spaces will be of similar dimension N^{loc} and that $N^{\mathrm{loc}} \approx N^{\mathrm{glob}}$. On the other hand, it may be the case that the influence of $\boldsymbol{\mu}$ on $u(\boldsymbol{\mu})$ is weak in many Ω_m, in which case $N \ll N^{\mathrm{loc}} M^{\mathrm{tot}}$, or that each of the p components of $\boldsymbol{\mu} \in \mathbb{R}^p$ affects $u(\boldsymbol{\mu})$ on different subdomains, in which case $N^{\mathrm{loc}} M^{\mathrm{tot}} \ll N^{\mathrm{glob}} M^{\mathrm{tot}}$. Thus, the actual loss in online efficiency due to localization will strongly depend on the type of problem to be solved.

More importantly though, note that the localization of V_N results in a change of the structure of the reduced system matrices $\mathbb{A}^q$. While these matrices are dense for global reduced basis approximations, localized reduced basis schemes yield $\mathbb{A}^q$ with a sparse block structure of $M^{\mathrm{tot}} \times M^{\mathrm{tot}}$ blocks of maximum dimension $N^{\mathrm{loc}} \times N^{\mathrm{loc}}$ and a maximum of C_{cup} blocks per row; C_{cup} depends on the specific localization method and on the topology of $\mathcal{T}_H$. For instance, for nonconforming methods, $C_{\mathrm{cup}} - 1$ is given by the maximum number of interfaces of a given subdomain Ω_m, whereas for the ArbiLoMod with a quadrilateral mesh $C_{\mathrm{cup}} = 25$.

Thus, estimate (6.75) has to be interpreted in relation to the fact that the computational complexity for solving (6.71) can be vastly reduced in comparison to (6.72) by exploiting the structure of $\mathbb{A}^q$. In particular, the costs for assembling (6.71) can be reduced to $\mathcal{O}(C_{\mathcal{T}_H} C_{\mathrm{cup}} (N^{\mathrm{loc}})^2 M)$. For the solution of (6.71) direct or block-preconditioned iterative solvers can be used. For the latter, the computational effort can be expected to

increase subquadratically in the number of subdomains M. In the scRBE method, the volume degrees of freedom associated with the spaces V_N^m are eliminated from (6.71) using static condensation to improve computational efficiency.

6.7.2 Offline costs and parallelization

While the local reduced basis spaces V_N^m, V_N^γ, V_N^e, V_N^v can be initialized by decomposing global solution snapshots $u_h(\boldsymbol{\mu})$ with respect to (6.10) [5], the core element of localized reduced basis methods is the construction of local reduced basis spaces from local problems associated with the subdomains Ω_m, as described in Section 6.4. This has various computational benefits:

First, we can expect a reduction of computational complexity as for most linear solvers we expect a superlinear increase in computational complexity for an increasing dimension of V_h, whereas the ratio of the dimensions of V_h and the local subspaces in (6.10) remains constant of order $1/M$ (for the volume spaces V_h^m and smaller for the spaces V_h^γ, V_h^e, V_h^v) as $h \to 0$. Thus, solving $\mathcal{O}(MN^{\mathrm{loc}})$ training problems of size $\dim V_h/M$ is expected to be faster than solving N global problems. At the same time, we expect N^{loc} to decrease for $H \to h$. Thus smaller subdomains Ω_m will generally lead to shorter offline times at the expense of less optimal spaces V_N. In particular, for the nonconforming schemes in Section 6.3.2.2 it is readily seen that for $H = h$ we have $V_N = V_h$ and that (6.13) and (6.11) are equivalent.

Even more important than a potential reduction of complexity is the possibility to choose H small enough such that each local training problem can be solved communication-free on a single compute node without the need for a high-performance interconnect. Also the problem setup and the computation of the reduced system (6.71) can be performed mostly communication-free: Instead of instantiating a global fine-scale compute mesh, each compute node can generate a local mesh from a geometry definition and solve training problems for a given local reduced basis space and all coupling spaces to obtain corresponding block-entries in $\mathbb{A}^q$, $\mathbb{F}^q$. Only the local geometry and the resulting reduced-order quantities are communicated (see [20, Section 8]). This makes localized reduced basis methods attractive for cloud-based environments, where large computational resources can be dynamically made available, but communication speed is limited.

Depending on the problem structure, the use of online enrichment (Section 6.6) can yield smaller, problem-adapted reduced spaces V_N. Similar to the training of V_N, online enrichment is based on the solution of small independent local problems that can easily be parallelized. As typically only some fraction of the subspaces of V_N undergo enrichment, fewer computational resources need to be allocated during an online enrichment phase. It has to be noted, however, that online enrichment leads to a propagation of snapshot data through the computational domain as the value of the current solution $u_N(\mu)$ at the boundary of the enrichment problem domain enters

the problem definition. Thus, to perform online enrichment, (boundary values of) reduced basis vectors have to be communicated between compute nodes and the entire reduced basis has to be kept available.

6.8 Applications and numerical experiments

6.8.1 Multiscale problems

We demonstrate the IP localized reduced basis methods from Section 6.3.2.2 in the context of parametric multiscale problems, such as Example 6.2, with a focus on online adaptivity as in Section 6.6 (using the a posteriori error estimate from Section 6.5.2), rather than offline training. These experiments were first published in the context of the online adaptive LRBMS in [90]. We consider a multiplicative splitting of the parameter dependency and the multiscale nature of the data functions, in the sense that $\kappa(\boldsymbol{\mu}) := \lambda(\boldsymbol{\mu})\kappa_\varepsilon$, with a parametric total mobility $\lambda : \mathcal{P} \to L^\infty(\Omega)$ and a highly heterogeneous permeability field $\kappa_\varepsilon \in L^\infty(\Omega)^{d \times d}$. To be more precise, we consider (6.2) on $\Omega = [0,5] \times [0,1]$ with $f(x,y) = 2 \cdot 10^3$ if $(x,y) \in [0.95, 1.10] \times [0.30, 0.45]$, $f(x,y) = -1 \cdot 10^3$ if $(x,y) \in [3.00, 3.15] \times [0.75, 0.90]$ or $(x,y) \in [4.25, 4.40] \times [0.25, 0.40]$, and 0 everywhere else, $\lambda(x,y;\boldsymbol{\mu}) = 1 + (1-\boldsymbol{\mu})\lambda_c(x,y)$, homogeneous Dirichlet boundary values, and a parameter space $\mathcal{P} = [0.1, 1]$. On each $t \in \tau_h$, $\kappa_\varepsilon|_t$ is the corresponding 0-th entry of the permeability tensor used in the first model of the 10th SPE Comparative Solution Project (which is given by 100×20 constant tensors, see [104]) and λ_c models a channel, as depicted in Figure 6.6, top left.

The right-hand side f models a strong source in the middle left of the domain and two sinks in the top and right middle of the domain, as is visible in the structure of the solutions (Figure 6.6, third row). The role of the parameter $\boldsymbol{\mu}$ is to toggle the existence of the channel λ_c. Thus $\lambda(1)\kappa_\varepsilon = \kappa_\varepsilon$ while $\boldsymbol{\mu} = 0.1$ models the removal of a large conductivity region near the center of the domain (see the second row in Figure 6.6). This missing channel has a visible impact on the structure of the pressure distribution as well as the reconstructed velocities, as we observe in the last two rows of Figure 6.6. With a contrast of 10^6 in the diffusion tensor and an ε of about $|\Omega|/2{,}000$, this setup is a challenging heterogeneous multiscale problem.

We used several software packages for this numerical experiment and refer to [90] for a full list and instructions on how to reproduce these results. We would like to mention that all grid-related structures (such as data functions, operators, functionals, products, norms) were implemented in a DUNE-based C++ discretization (which is by now contained in the DUNE extension modules[10] and the generic discretization

---

10 https://github.com/dune-community/dune-xt/
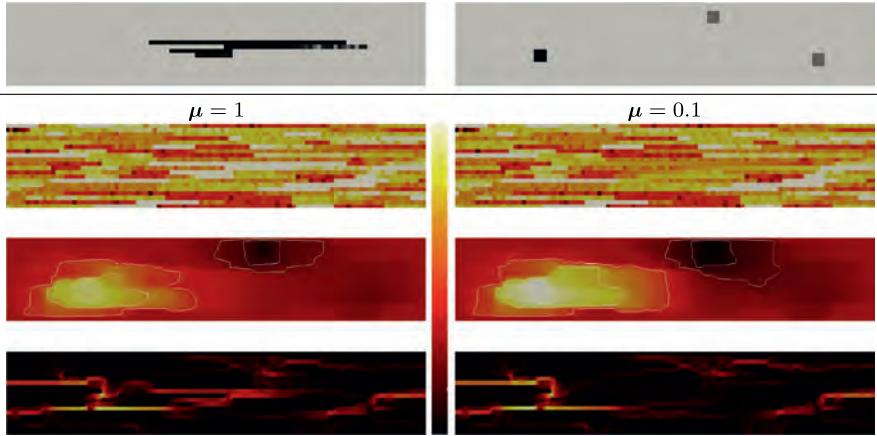
Figure 6.6: Data functions and sample solutions of the experiment in Section 6.8.1. First row: Location of the channel function λ_c (left) and plot of the force f (right) modeling one source (black: $2 \cdot 10^3$) and two sinks (dark gray: $-1 \cdot 10^3$, zero elsewhere). Second to fourth rows: Both plots in each row share the same color map (middle) with different ranges per row, for parameters $\boldsymbol{\mu} = 1$ (left column) and $\boldsymbol{\mu} = 0.1$ (right column). From top to bottom: Logarithmic plot of $\lambda(\boldsymbol{\mu})\kappa_\varepsilon$ (dark: $1.41 \cdot 10^{-3}$, light: $1.41 \cdot 10^3$), plot of the pressure $u_h(\boldsymbol{\mu})$ (IP localized FOM solution of (6.2), dark: $-3.92 \cdot 10^{-1}$, light: $7.61 \cdot 10{-1}$, isolines at 10%, 20%, 45%, 75%, and 95%), and plot of the magnitude of the reconstructed diffusive flux $R_h^0[u_h(\boldsymbol{\mu}); \boldsymbol{\mu}]$ (defined in (6.65) and (6.66), dark: $3.10 \cdot 10^{-6}$, light: $3.01 \cdot 10^2$). Note the presence of high-conductivity channels in the permeability (second row left, light regions) throughout large parts of the domain. The parameter dependency models a removal of one such channel in the middle right of the domain (second row right), well visible in the reconstructed Darcy velocity fields (bottom).

toolbox dune-gdt[11]), while we used pyMOR [84] for everything related to model reduction (such as Gram–Schmidt and greedy). We consider a domain decomposition of $|\mathcal{T}_H| = 25 \times 5$ squares, each refined such that the full global grid would consist of $|\tau_h| = 1{,}014{,}000$ elements. For the IP localized FOM, following Section 6.3.2.2, we choose on each subdomain $\Omega_m \in \mathcal{T}_H$ the discontinuous Galerkin space (first order), product, and bilinear form from Example 6.10. For error estimation, we employed the flux reconstruction ansatz from Section 6.5.2 using a zero-order diffusive flux reconstruction (compare Theorem 6.21).

The sole purpose of these experiments is to demonstrate the capabilities of localized reduced basis methods regarding online enrichment. We thus initialize the local reduced spaces V_N^m on each subdomain a priori by orthonormalized Lagrangian shape functions of order up to one, thus obtaining a reduced space with poor approximation properties (comparable to a standard discontinuous Galerkin space with respect to the domain decomposition). Since we employ the a posteriori error estimate η on the full approximation error (including the discretization as well as the model reduction

---

11 https://github.com/dune-community/dune-gdt/

error) from Theorem 6.21, and since we omit grid refinement in these experiments, the estimated discretization error over all parameters of 1.66 is a lower bound for the overall approximation error, and we thus choose a tolerance of $\Delta_{\text{online}} = 2$ for the online enrichment in Algorithm 6.3.

We compare two different strategies, corresponding to the two plots in Figure 6.7. In both cases, we simulate an outer loop application in the online part by randomly choosing ten parameters $\mathcal{P}_{\text{online}} \subset \mathcal{P}$ which are subsequently processed. For each parameter, the local reduced spaces are enriched according to Algorithm 6.3 and the respective marking strategy, until the estimated error is below the specified tolerance. Note that the evaluation of the localizable a posteriori error estimate can be fully offline-online decomposed and that after each enrichment only information from a subdomain and its neighbors are required to locally update the offline-online decomposed data.
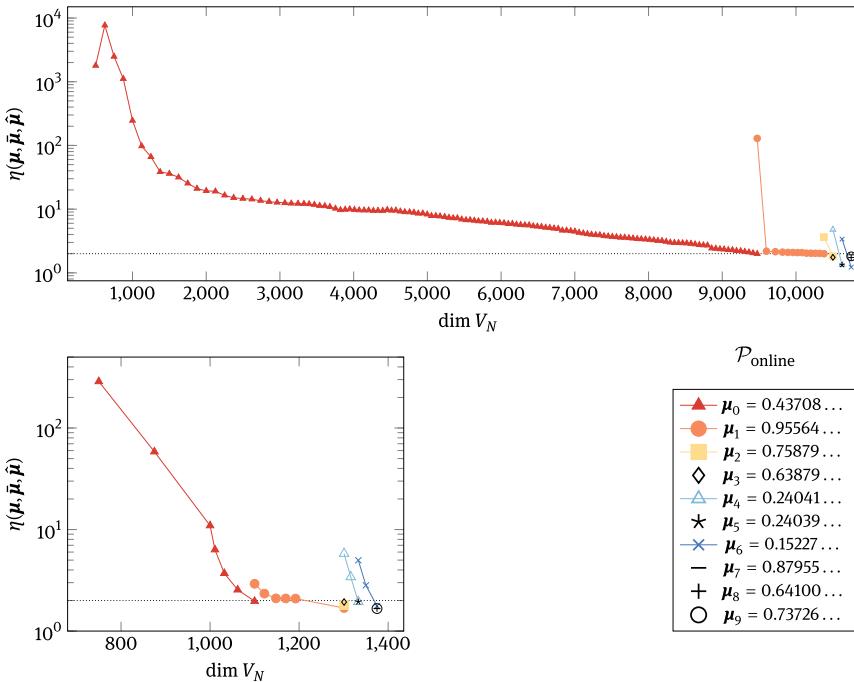
Figure 6.7: Estimated error evolution during the adaptive online phase for the experiment in Section 6.8.1 with $|\mathcal{T}_H| = 125$, $k_H = 1$, $\Delta_{\text{online}} = 2$ (dotted line), $\bar{\mu} = \hat{\mu} = 0.1$, for different online and offline strategies: no global snapshot (greedy search disabled, $N_{\text{greedy}} = 0$) during the offline phase, uniform marking during the online phase (top) and two global snapshots (greedy search on $\mathcal{P}_{\text{train}} = \{0.1, 1\}$, $N_{\text{greedy}} = 2$) and combined uniform marking while $\eta(\mu, \bar{\mu}, \hat{\mu}) > \theta_{\text{uni}}\Delta_{\text{online}}$ with $\theta_{\text{uni}} = 10$, Dörfler marking with $\theta_{\text{doerf}} = 0.85$, and age-based marking with $N_{\text{age}} = 4$ (bottom left); note the different scales. With each strategy the local reduced bases are enriched according to Algorithm 6.3 while subsequently processing the online parameters $\mu_0, \ldots, \mu_9$ (bottom right).

In the first experiment, we use a uniform marking strategy, which results in an unconditional enrichment on each subdomain (comparable to domain decomposition methods). As we observe in Figure 6.7 (top), however, it takes 129 enrichment steps to lower the estimated error below the desired tolerance for the first online parameter μ_0. After this extensive enrichment it takes 12 steps for μ_1 and none or one enrichment step to reach the desired tolerance for the other online parameters. The resulting coarse reduced space is of size 10,749 (with an average of 86 basis functions per subdomain), which is clearly not optimal. Although each subdomain was marked for enrichment, the sizes of the final local reduced bases differ since the local Gram–Schmidt basis extension may reject updates (if the added basis function is locally not linearly independent). As we observe in Figure 6.8 (left) this is indeed the case with local basis sizes ranging between 24 and 148. Obviously, a straightforward domain decomposition ansatz without suitable training is not feasible for this setup. This is not surprising since the data functions exhibit strong multiscale features and nonlocal high-conductivity channels connecting domain boundaries; see Figure 6.6.

Figure 6.8: Spatial distribution of the final sizes of the local reduced bases on each subdomain, after the adaptive online phase for the experiment in Section 6.8.1 with $\Omega = [0,5] \times [0,1]$, $|\mathcal{T}_H| = 25 \times 5$ for the two strategies shown in Figure 6.7: no global snapshot with uniform enrichment (left, light: 24, dark: 148) and two global snapshots with adaptive enrichment (right, light: 9, dark: 20). Note the pronounced structure (right) reflecting the spatial structure of the data functions (compare Figure 6.6).

To remedy the situation we allow for two global snapshots during the offline phase (for parameters $\mu \in \{0.1, 1\}$) and use an adaptive marking strategy which combines uniform marking, Dörfler marking, and age-based marking (see the caption of Figure 6.7) in the online phase. This strategy employs uniform marking until a saturation condition is reached, and afterwards uses a Dörfler marking combined with a marking based on counting how often a subdomain has not been marked. With two global solution snapshots incorporated in the basis the situation improves significantly, as we observe in Figure 6.7 (bottom left). In total we observe only two enrichment steps with uniform marking (see the first two steps for μ_0), which indicates that further offline training would be beneficial. The number of elements marked ranges between 11 and 110 (over all online parameters and all but the first two enrichment steps) with a mean of 29 and a median of 22. Of these marked elements only once have 87 out of 110 elements been marked due to their age (see the last step for μ_1). Overall we could reach a significantly lower overall basis size than in the previous setup (1,375 vs. 10,749) and the sizes of the final local bases range between only 9 and 20 (compared to 24 to 148 above). We also observe in Figure 6.8 (right) that the spatial distribution of the basis sizes follows

the spatial structure of the data functions (compare Figure 6.6), which nicely shows the localization qualities of our error estimator.

6.8.2 Fluid dynamics

Flow simulations in pipelined channels have a growing interest in many biological and industrial applications. The localized model order reduction approaches presented in this chapter are suitable for the study of internal flows in hierarchical parameterized geometries. In particular, the nonconforming approach introduced in Section 6.3.2 has applications in the analysis of blood flow in specific compartments of the circulatory system that can be represented as a combination of few deformed vessels from a reference one.

We want to solve the Stokes equation defined in (6.3), with $\delta = 0$, in a computational domain Ω composed by two stenosed blocks Ω_{μ_1} and Ω_{μ_2} (Figure 6.9), by imposing nonhomogeneous boundary conditions $\sigma_n^{in} = [0, 5]^T$ in the inlet surface ($x_1 = 10$), nonhomogeneous boundary conditions $\sigma_n^{in} = [0, -1]^T$ in the outlet surface ($x_1 = 0$) and homogeneous Dirichlet boundary conditions on the remaining boundaries of the domain. Here, the Taylor–Hood FEM has been used to compute the basis functions, $\mathbb{P}_2$ elements for velocity and supremizer (cf. Chapter 8 of Volume 3 of *Model order reduction* for a definition of supremizer functions), $\mathbb{P}_1$ for pressure, respectively, and consequently $\mathbb{P}_1(\Gamma_{m,m'})$ for the Lagrange multipliers space. Figure 6.11 shows the distribution of the parameter values selected by the greedy algorithm, by applying the offline stage of the reduced basis method to the single stenosis block. By taking into account that the range $[-5, 5]$ is not admitted, we can see that the higher concentration of values is in the intervals $[-10, -5]$ and $[5, 10]$ in correspondence to larger deformation of the pipe.

Figure 6.9: Computational domain ($\mu_1 = 7, \mu_2 = 10$).

The geometry of a single stenosis is obtained by the deformation of a reference pipe through a parameter that represents the contraction in the middle of the pipe. The deformed domain Ω_μ is mapped from the straight reference pipe $\hat{\Omega}$ of length $L = 5$ and radius $r = 1$ through the following coordinate transformation $T_\mu : \hat{\Omega} \to \Omega_\mu$ such as $\boldsymbol{x} = T_\mu(\hat{\boldsymbol{x}})$ and $x_1 = \hat{x}_1 + \frac{\hat{x}_1}{\mu}(\cos(\frac{2\pi\hat{x}_3}{L}) - 1)$, $x_2 = \hat{x}_2 + \frac{\hat{x}_2}{\mu}(\cos(\frac{2\pi\hat{x}_3}{L}) - 1)$, $x_3 = \hat{x}_3$. The range of the parameter μ is $[-20, -5] \cup [5, 20]$; Figure 6.10 shows the reference pipe and some representative deformations of the geometry. In order to compute the basis functions, we consider a parameterized Stokes problem for each subdomain. For the
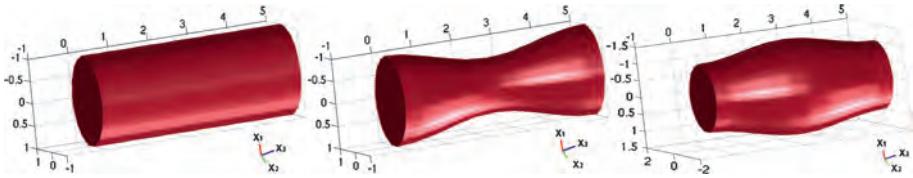
Figure 6.10: Reference pipe and two deformed pipes ($\mu = -5$, $\mu = 5$): stenosis and aneurysm configurations.
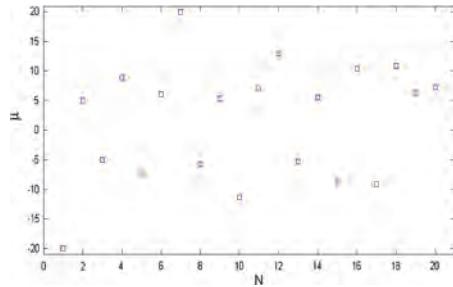
Figure 6.11: Distribution of the selected parameter values by the greedy algorithm used to generate the basis functions in a single block.

first subdomain, we compute the reduced basis imposing zero Dirichlet condition on the wall and Neumann boundary conditions given by imposing $\sigma_n = \sigma \cdot \mathbf{n} = \nu \frac{\partial \boldsymbol{u}}{\partial \mathbf{n}} - p\mathbf{n}$ to be $\sigma_n^{in} = [0,5]^T$ on Γ_{in} and $\sigma_n^{out} = \mathbf{0}$ on the internal interface. For the second subdomain, we compute the reduced basis imposing zero Dirichlet condition on the wall and Neumann boundary conditions imposing $\sigma_n^{in} = \mathbf{0}$ on the internal interface and $\sigma_n^{out} = [0,-1]^T$ on the outflow interface Γ_{out}.

Moreover, we enrich the local reduced basis spaces by a coarse FE solution of the problem computed in the global domain. This strategy ensures not only the continuity of the velocity, but also the one of the normal stress along the internal interface. For this reason this method is called reduced basis hybrid method. Coarse and fine grids have been chosen in order to deal with respectively 155 and 2,714 nodes in a single block domain. Figure 6.12 shows a representative flow solution in Ω, found with the
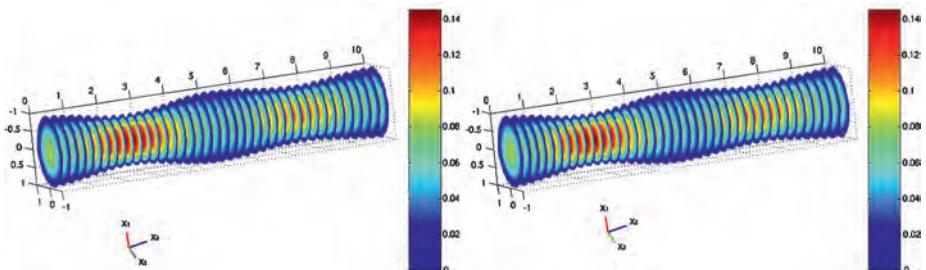
Figure 6.12: Representative solutions of velocity using the reduced basis hybrid method (with $N_1 = N_2 = 19$) (left) and using the FEM as a global solution (right), $\mu_1 = 7$, $\mu_2 = 10$.
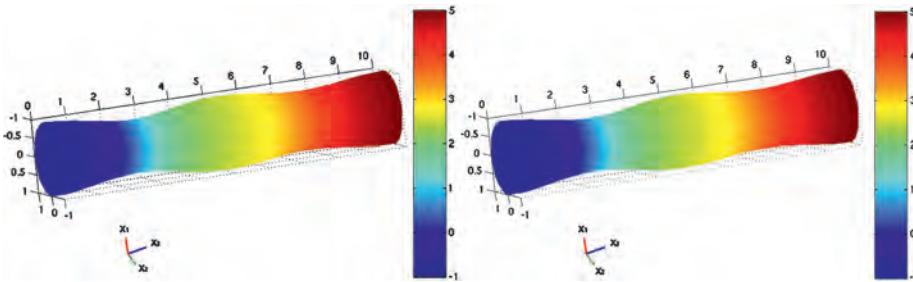
Figure 6.13: Representative solutions of pressure using the reduced basis hybrid method (with $N_1 = N_2 = 19$) (left) and using the FEM as a global solution (right), $\mu_1 = 7$, $\mu_2 = 10$.

reduced basis hybrid method, to be compared with the FE solution. The same comparison, regarding the pressure solutions, is shown in Figure 6.13.

6.9 Further perspectives

6.9.1 Parabolic problems

Most of the techniques presented in this chapter so far can be extended or even directly applied to parabolic problems. For instance, local approximation spaces that are optimal in the sense of Kolmogorov are proposed in [99] and the LRBMS method for parabolic problems is presented in [87, 86]. To facilitate an adaptive construction of the local reduced space or online adaptivity, a suitable, localized a posteriori error estimator is key. Therefore, we present in this subsection an abstract framework for a posteriori error estimation for approximations of scalar parabolic evolution equations, based on elliptic reconstruction techniques. For further reading and the application to localized model reduction we refer to [44, 87].

Definition 6.22 (Parameterized parabolic problem in variational form). Let a Gelfand triple of suitable Hilbert spaces $V \subset H = H' \subset V'$, an end time $T_{\text{end}} > 0$, initial data $u_0 \in V$, and a right-hand side $f \in H$ be given. For a parameter $\boldsymbol{\mu} \in \mathcal{P}$ find $u(\cdot; \boldsymbol{\mu}) \in L^2(0, T_{\text{end}}; V)$ with $\partial_t u(\cdot; \boldsymbol{\mu}) \in L^2(0, T_{\text{end}}; V')$, such that $u(0; \boldsymbol{\mu}) = u_0$ and

$$\langle \partial_t u(t; \boldsymbol{\mu}), q \rangle + a(u(t; \boldsymbol{\mu}), v; \boldsymbol{\mu}) = f(v; \boldsymbol{\mu}) \quad \text{for all } v \in V. \tag{6.76}$$

Depending on the error we want to quantify, the space V in (6.76) can be either an analytical function space as in (6.1) or an already discretized function space V_h. We drop the parameter dependency in this section to simplify the notation.

Definition 6.23 (Approximations of the parabolic problem). Let $\widetilde{V} \subseteq H$ be a finite-dimensional approximation space for V, not necessarily contained in V. Potential

candidates for $\widetilde{V}$ are conforming or nonconforming localized model reduction spaces V_N as discussed above, but also FE or FV spaces fit into this setting. Denote by $(\cdot, \cdot)$, $\|\cdot\|$ the H-inner product and the norm induced by it.

Let $f \in H$, and let $a_h : (V + \widetilde{V}) \times (V + \widetilde{V}) \to \mathbb{R}$ be a discrete bilinear form which coincides with a on $V \times V$ and is thus continuous and coercive on V. Let further $\|\|\cdot\|\|$ be a norm over $V + \widetilde{V}$, which coincides with the square root of the symmetric part of a_h over V.

Our goal is to bound the error $e(t) := u(t) - \tilde{u}(t)$ between the analytical (or discrete) solution $u \in L^2(0, T_{\text{end}}; V)$, $\partial_t u \in L^2(0, T_{\text{end}}; V')$ of (6.76), where the duality pairing $\langle \partial_t u(t), v \rangle$ is induced by the H-scalar product via the Gelfand triple and the $\widetilde{V}$-Galerkin approximation $\tilde{u} \in L^2(0, T_{\text{end}}, \widetilde{V})$, $\partial_t \tilde{u} \in L^2(0, T_{\text{end}}, \widetilde{V})$, solution of

$$(\partial_t \tilde{u}(t), \tilde{v}) + a_h(\tilde{u}(t), \tilde{v}) = (f, \tilde{v}) \quad \text{for all } \tilde{v} \in \widetilde{V}. \tag{6.77}$$

Definition 6.24 (Elliptic reconstruction). Denote by $\widetilde{\Pi}$ the H-orthogonal projection onto $\widetilde{V}$. For $\tilde{v} \in \widetilde{V}$, define the elliptic reconstruction $\mathcal{R}_{\text{ell}}(\tilde{v}) \in V$ of $\tilde{v}$ to be the unique solution of the variational problem

$$a_h(\mathcal{R}_{\text{ell}}(\tilde{v}), v) = (A_h(\tilde{v}) - \widetilde{\Pi}(f) + f, v) \quad \text{for all } v \in V, \tag{6.78}$$

where $A_h(\tilde{v}) \in \widetilde{V}$ is the H-inner product Riesz representative of the functional $a_h(\tilde{v}, \cdot)$, i. e., $(A_h(\tilde{v}), \tilde{v}') = a_h(\tilde{v}, \tilde{v}')$ for all $\tilde{v}' \in \widetilde{V}$. Note that $\mathcal{R}_{\text{ell}}(\tilde{v})$ is well-defined, due to the coercivity of a_h on V.

From the definition it is clear that $\tilde{v}$ is the $\widetilde{V}$-Galerkin approximation of the elliptic reconstruction $\mathcal{R}_{\text{ell}}(\tilde{v})$.

Let us assume that for each t we have a decomposition $\tilde{u}(t) =: \tilde{u}^c(t) + \tilde{u}^d(t)$ (not necessarily unique), where $\tilde{u}^c(t) \in V$, $\tilde{u}^d(t) \in \widetilde{V}$ are the conforming and nonconforming parts of $\tilde{u}(t)$. We consider the following error quantities:

$$\begin{aligned}
\rho(t) &:= u(t) - \mathcal{R}_{\text{ell}}(\tilde{u}(t)), & \varepsilon(t) &:= \mathcal{R}_{\text{ell}}(\tilde{u}(t)) - \tilde{u}(t), \\
e^c(t) &:= u(t) - \tilde{u}^c(t), & \varepsilon^c(t) &:= \mathcal{R}_{\text{ell}}(\tilde{u}(t)) - \tilde{u}^c(t).
\end{aligned}$$

Theorem 6.25 (Abstract semi-discrete error estimate). *Let $C := (2\gamma_h^2 + 1)^{1/2}$, where γ_h denotes the continuity constant of a_h on V with respect to $\|\|\cdot\|\|$. Then*

$$\begin{aligned}
\|e\|_{L^2(0, T_{end}; \|\|\cdot\|\|)} \leq &\|e^c(0)\| + \sqrt{3} \|\partial_t \tilde{u}^d\|_{L^2(0, T_{end}; \|\|\cdot\|\|_{V, -1})} \\
&+ (C + 1) \cdot \|\varepsilon\|_{L^2(0, T_{end}; \|\|\cdot\|\|)} + C \cdot \|\tilde{u}^d\|_{L^2(0, T_{end}; \|\|\cdot\|\|)}.
\end{aligned}$$

Note that $\varepsilon(t)$ denotes the approximation error of the coercive variational problem (6.78). Hence, this error contribution can be controlled by invoking any (localized) a posteriori error estimate for coercive variational problem as, e. g., presented in Section 6.5.

It is straightforward to modify the estimate in Theorem 6.25 for semi-discrete solutions $\tilde{u}(t)$ to take the time discretization error into account.

Corollary 6.26. *Let* $\tilde{u} \in L^2(0, T_{end}, \widetilde{V})$, $\partial_t \tilde{u} \in L^2(0, T_{end}, \widetilde{V})$ *be an arbitrary discrete approximation of* $u(t)$, *not necessarily satisfying* (6.77). *Let* $\mathcal{R}_T[\tilde{p}](t) \in \widetilde{V}$ *denote the* $\widetilde{V}$-*Riesz representative with respect to the* H-*inner product of the time-stepping residual of* $\tilde{u}(t)$, *i. e.,*

$$\left(\mathcal{R}_T[\tilde{u}](t), \tilde{v}\right) = \left(\partial_t \tilde{u}(t), \tilde{v}\right) + a_h\left(\tilde{u}(t), \tilde{v}\right) - (f, \tilde{q}) \quad \forall \tilde{v} \in \tilde{v}.$$

Then, with $C := (3\gamma_h^2 + 2)^{1/2}$, *the following error estimate holds:*

$$
\begin{aligned}
\|e\|_{L^2(0, T_{end}; \||\cdot\||)} &\leq \left\|e^c(0)\right\| + 2\left\|\partial_t \tilde{u}^d\right\|_{L^2(0, T_{end}; \||\cdot\||_{V, -1})} \\
&\quad + (C + 1) \cdot \|\varepsilon\|_{L^2(0, T_{end}; \||\cdot\||)} + C \cdot \left\|\tilde{u}^d\right\|_{L^2(0, T_{end}; \||\cdot\||)} \\
&\quad + 2C_{H, V}^b \cdot \left\|\mathcal{R}_T[\tilde{u}]\right\|_{L^2(0, T_{end}; H)}.
\end{aligned}
$$

6.9.2 Nonaffine parameter dependence and nonlinear problems

A key ingredient towards model order reduction for nonlinear problems is the empirical interpolation method introduced in [13] and further developed in [33], [26, 75]. For a general exposition we refer to Chapter 5 of this volume of *Model order reduction*.

In the context of localized model order reduction, empirical interpolation has been employed in, e. g., [22, 96, 86]. Based on the concept of empirical operator interpolation from [33], localization strategies can be employed as follows. To present the main ideas, let us assume the simple situation that

$$V_h = \bigoplus_{m=1}^{M} V_h^m$$

and that we have a localized decomposition as follows:

$$a_h(u_h(\boldsymbol{\mu}), v_h; \boldsymbol{\mu}) = \sum_{m=1}^{M} a_h^m(u_h^m(\boldsymbol{\mu}), v_h^m; \boldsymbol{\mu}),$$

with $a_h(u_h(\boldsymbol{\mu}), \cdot; \boldsymbol{\mu}) \in (V_h)'$. The strategy will then rely on an empirical operator interpolation of the local volume operators $a_h^m(u_h^m(\boldsymbol{\mu}), \cdot; \boldsymbol{\mu}) \in (V_h^m)'$ and will thus only involve localized computations in the construction of the interpolation operator. As an example, the interpolation of the local volume operator will be of the form

$$\mathcal{I}_L^m[a_h^m(u_h^m(\boldsymbol{\mu}), \cdot; \boldsymbol{\mu})] = \sum_{l=1}^{L} \mathscr{S}_l^m(a_h^m(u_h^m(\boldsymbol{\mu}), \cdot; \boldsymbol{\mu})) q_l^m$$

for a local collateral basis $\{q_l^m\}_{l=1}^L \subset (V_h^m)'$ and corresponding interpolation functionals $\{\mathscr{S}_l^m\}_{l=1}^L \subset \Sigma_h^{m''}$ from a suitable local dictionary $\Sigma_h^{m''} \subset (V_h^m)''$, the choice of which is crucial to ensure the accuracy as well as an online-efficient evaluation of the interpolant. Note that because the isomorphism between V_h^m and is bi-dual, the local dictionary of interpolation functionals $\Sigma_h^{m''}$ can be identified with a dictionary of functions $\Sigma_h^m \subset V_h^m$, such that $\mathscr{S}_l^m(a_h^m(u_h^m(\boldsymbol{\mu}), \cdot; \boldsymbol{\mu})) = a_h^m(u_h^m(\boldsymbol{\mu}), \sigma_l^m; \boldsymbol{\mu})$, where $\sigma_l^m \in \Sigma_h^m$

corresponds to $\mathscr{S}_l^m \in \Sigma_h^{m''}$. An online-efficient evaluation of the interpolated operator $\mathcal{I}_L^m[a_h^m(u_h^m(\boldsymbol{\mu}), \cdot; \boldsymbol{\mu})]$ can be ensured by choosing the local dictionary Σ_h^m such that the computational complexity of the evaluation $a_h^m(u_h^m(\boldsymbol{\mu}), \sigma^m; \boldsymbol{\mu})$ for $\sigma^m \in \Sigma_h^m$ does not depend on the dimension of V_h^m. The choice of Σ_h^m thus depends on the underlying discretization: Possible choices in the context of FE schemes include the FE basis of V_h^m. Other choices of Σ_h^m are conceivable and could improve the interpolation quality, which is subject to further investigation.

6.10 Conclusion

In this chapter we have given an introduction to projection-based localized model order reduction techniques for parameterized coercive problems. Starting from an abstract localization framework, we have presented conforming and nonconforming coupling schemes for the local reduced spaces within this unified framework. For the generation of the local reduced spaces we have discussed a priori approaches based on polynomial spaces at the domain interfaces, as well as empirical approaches based on randomized localized training as an approximation of the optimal local reduced spaces. Further, we have introduced rigorous localizable a posteriori error estimates and discussed their use to steer an adaptive online enrichment of the approximation spaces based on the solution of local correction problems. Finally, we have discussed the online efficiency and parallelizability of the presented methods. We have given application examples from the fields of multiscale problems, linear elasticity, and fluid dynamics. Extensions to time-dependent or nonlinear problems are under active research, and we have given a brief outlook towards localized model order reduction for these problem classes in the final section of this chapter.

Bibliography

[1]   J. Aarnes and T. Y. Hou, Multiscale domain decomposition methods for elliptic problems with high aspect ratios, *Acta Mathematicae Applicatae Sinica*, **18** (1) (2002), 63–76.

[2]   A. Abdulle, On a priori error analysis of fully discrete heterogeneous multiscale FEM, *Multiscale Modeling & Simulation*, **4** (2) (2005), 447–459.

[3]   A. Abdulle and Y. Bai, Reduced basis finite element heterogeneous multiscale method for high-order discretizations of elliptic homogenization problems, *Journal of Computational Physics*, **231** (21) (2012), 7014–7036.

[4]   P. Abdulle and A. Henning, A reduced basis localized orthogonal decomposition, *Journal of Computational Physics*, **295** (2015), 379–401.

[5]   F. Albrecht, B. Haasdonk, S. Kaulmann, and M. Ohlberger, The localized reduced basis multiscale method, in *Proceedings of Algoritmy 2012, Conference on Scientific Computing, Vysoke Tatry, Podbanske, September 9-14, 2012*, pp. 393–403, 2012.

[6]   P. F. Antonietti, P. Pacciarini, and A. Quarteroni, A discontinuous Galerkin reduced basis element method for elliptic problems, *ESAIM: Mathematical Modelling and Numerical Analysis*, **50** (2) (2016), 337–360.

[7]   I. Babuška, U. Banerjee, and J. Osborn, Generalized finite element methods — main ideas, results and perspective, *International Journal of Computational Methods*, **1** (1) (2004), 67–103.

[8]   I. Babuška, G. Caloz, and J. E. Osborn, Special finite element methods for a class of second order elliptic problems with rough coefficients, *SIAM Journal on Numerical Analysis*, **31** (4) (1994), 945–981.

[9]   I. Babuška, X. Huang, and R. Lipton, Machine computation using the exponentially convergent multiscale spectral generalized finite element method, *ESAIM: Mathematical Modelling and Numerical Analysis*, **48** (2) (2014), 493–515.

[10]   I. Babuška and R. Lipton, Optimal local approximation spaces for generalized finite element methods with application to multiscale problems, *Multiscale Modeling & Simulation*, **9** (1) (2011), 373–406.

[11]   I. Babuška and J. M. Melenk, The partition of unity method, *International Journal for Numerical Methods in Engineering*, **40** (4) (1997), 727–758.

[12]   M. Bampton and R. Craig, Coupling of substructures for dynamic analyses, *AIAA Journal*, **6** (7) (1968), 1313–1319.

[13]   M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations, *Comptes Rendus. Mathématique*, **339** (9) (2004), 667–672.

[14]   P. Benner, A. Cohen, M. Ohlberger, and K. Willcox (eds.), *Model Reduction and Approximation*. Computational Science & Engineering, vol. 15, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2017 Theory and algorithms.

[15]   P. Benner, S. Gugercin, and K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, *SIAM Review*, **57** (4) (2015), 483–531.

[16]   C. Bernardi, Y. Maday, and A. T. Patera, A new nonconforming approach to domain decomposition: the mortar element method, in *Nonlinear Partial Differential Equations and Their Applications. Collège de France Seminar, Vol. XI (Paris, 1989–1991)*. Pitman Res. Notes Math. Ser., vol. 299, pp. 13–51, Longman Sci. Tech., Harlow, 1994.

[17]   F. Bourquin, Component mode synthesis and eigenvalues of second order operators: discretization and algorithm, *RAIRO Modélisation Mathématique et Analyse Numérique*, **26** (3) (1992), 385–423.

[18]   F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag New York, Inc., 1991.

[19]   A. Buhr, C. Engwer, M. Ohlberger, and S. Rave, A numerically stable a posteriori error estimator for reduced basis approximations of elliptic equations, in X. Oliver, E. Onate, and A. Huerta (eds.), *Proceedings of the 11th World Congress on Computational Mechanics*, pp. 4094–4102, CIMNE, Barcelona, 2014.

[20]   A. Buhr, C. Engwer, M. Ohlberger, and S. Rave, ArbiLoMod, a simulation technique designed for arbitrary local modifications, *SIAM Journal on Scientific Computing*, **39** (4) (2017), A1435–A1465.

[21]   A. Buhr and K. Smetana, Randomized local model order reduction, *SIAM Journal on Scientific Computing*, **40** (4) (2018), A2120–A2151.

[22]   V. M. Calo, Y. Efendiev, J. Galvis, and M. Ghommem, Multiscale empirical interpolation for solving nonlinear PDEs, *Journal of Computational Physics*, **278** (2014), 204–220.

[23]   V. M. Calo, Y. Efendiev, J. Galvis, and G. Li, Randomized oversampling for generalized multiscale finite element methods, *Multiscale Modeling & Simulation*, **14** (1) (2016), 482–501.

[24] K. Carlberg, Adaptive *h*-refinement for reduced-order models, *International Journal for Numerical Methods in Engineering*, **102** (5) (2015), 1192–1210.

[25] F. Casenave, A. Ern, and T. Lelièvre, Accurate and online-efficient evaluation of the *a posteriori* error bound in the reduced basis method, *ESAIM: Mathematical Modelling and Numerical Analysis*, **48** (2014), 207–229.

[26] S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM Journal on Scientific Computing*, **32** (5) (2010), 2737–2764.

[27] Y. Chen, J. S. Hesthaven, and Y. Maday, A seamless reduced basis element method for 2D Maxwell's problem: an introduction, in *Spectral and High Order Methods for Partial Differential Equations*. Lect. Notes Comput. Sci. Eng., vol. 76, pp. 141–152, Springer, Heidelberg, 2011.

[28] E. T. Chung, Y. Efendiev, and W. T. Leung, Residual-driven online generalized multiscale finite element methods, *Journal of Computational Physics*, **302** (2015), 176–190.

[29] E. T. Chung, Y. Efendiev, and W. T. Leung, An online generalized multiscale discontinuous Galerkin method (GMsDGM) for flows in heterogeneous media, *Communications in Computational Physics*, **21** (2) (2017), 401–422.

[30] E. T. Chung, Y. Efendiev, and W. T. Leung, An adaptive generalized multiscale discontinuous Galerkin method for high-contrast flow problems, *Multiscale Modeling & Simulation*, **16** (3) (2018), 1227–1257.

[31] E. T. Chung, Y. Efendiev, and G. Li, An adaptive GMsFEM for high-contrast flow problems, *Journal of Computational Physics*, **273** (Sep 2014), 54–76.

[32] P. Drineas and M. W. Mahoney, RandNLA: randomized numerical linear algebra, *Communications of the ACM*, **59** (6) (2016), 80–90.

[33] M. Drohmann, B. Haasdonk, and M. Ohlberger, Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation, *SIAM Journal on Scientific Computing*, **34** (2) (2012), A937–A969.

[34] Y. Efendiev, J. Galvis, and T. Y. Hou, Generalized multiscale finite element methods (GMsFEM), *Journal of Computational Physics*, (January 2013).

[35] Y. Efendiev, T. Hou, and V. Ginting, Multiscale finite element methods for nonlinear problems and their applications, *Communications in Mathematical Sciences*, **2** (4) (2004), 553–589.

[36] Y. Efendiev and T. Y. Hou, *Multiscale Finite Element Methods*. Surveys and Tutorials in the Applied Mathematical Sciences, vol. 4, Springer, New York, 2009 Theory and applications.

[37] J. L. Eftang and A. T. Patera, Port reduction in parametrized component static condensation: approximation and a posteriori error estimation, *International Journal for Numerical Methods in Engineering*, **96** (5) (2013), 269–302.

[38] J. L. Eftang and A. T. Patera, A port-reduced static condensation reduced basis element method for large component-synthesized structures: approximation and *A posteriori* error estimation, *Advanced Modeling and Simulation in Engineering Sciences*, **1** (3) (2014).

[39] A. Ern, A. F. Stephansen, and M. Vohralík, Guaranteed and robust discontinuous Galerkin a posteriori error estimates for convection–diffusion–reaction problems, *Journal of Computational and Applied Mathematics*, **234** (1) (2010), 114–130.

[40] A. Ern, A. F. Stephansen, and P. Zunino, A discontinuous Galerkin method with weighted averages for advection–diffusion equations with locally small and anisotropic diffusivity, *IMA Journal of Numerical Analysis*, **29** (2) (2009), 235–256.

[41] A. Ferrero, A. Iollo, and F. Larocca, Global and local POD models for the prediction of compressible flows with DG methods, *International Journal for Numerical Methods in Engineering*, **116** (5) (2018), 332–357.

[42] J. Galvis and Y. Efendiev, Domain decomposition preconditioners for multiscale flows in high-contrast media, *Multiscale Modeling & Simulation*, **8** (4) (2010), 1461–1483.

[43] M. J. Gander and A. Loneland, SHEM: an optimal coarse space for RAS and its multiscale approximation, in *Domain Decomposition Methods in Science and Engineering XXIII*. Lect. Notes Comput. Sci. Eng., vol. 116, pp. 313–321, Springer, Cham, 2017.

[44] E. H. Georgoulis, O. Lakkis, and J. M. Virtanen, A posteriori error control for discontinuous Galerkin methods for parabolic problems, *SIAM Journal on Numerical Analysis*, **49** (2) (2011), 427–458.

[45] I. G. Graham, P. O. Lechner, and R. Scheichl, Domain decomposition for multiscale PDEs, *Numerische Mathematik*, **106** (4) (2007), 589–626.

[46] N. Halko, P. Martinsson, and J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, *SIAM Review*, **53** (2) (2011), 217–288.

[47] A. Heinlein, A. Klawonn, J. Knepper, and O. Rheinbach, Multiscale coarse spaces for overlapping Schwarz methods based on the ACMS space in 2D, *Electronic Transactions on Numerical Analysis*, **48** (2018), 156–182.

[48] P. Henning, A. Malqvist, and D. Peterseim, A localized orthogonal decomposition method for semi-linear elliptic problems, *ESAIM: Mathematical Modelling and Numerical Analysis*, **48** (5) (2014), 1331–1349.

[49] P. Henning, M. Ohlberger, and B. Schweizer, An adaptive multiscale finite element method, *Multiscale Modeling & Simulation*, **12** (3) (2014), 1078–1107.

[50] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, SpringerBriefs in Mathematics. Springer, Cham; BCAM Basque Center for Applied Mathematics, Bilbao, 2016. BCAM SpringerBriefs.

[51] U. Hetmaniuk and A. Klawonn, Error estimates for a two-dimensional special finite element method based on component mode synthesis, *Electronic Transactions on Numerical Analysis*, **41** (2014), 109–132.

[52] U. Hetmaniuk and R. B. Lehoucq, A special finite element method based on component mode synthesis, *ESAIM: Mathematical Modelling and Numerical Analysis*, **44** (3) (2010), 401–420.

[53] C. Himpe, T. Leibner, and S. Rave, Hierarchical approximate proper orthogonal decomposition, *SIAM Journal on Scientific Computing*, **40** (5) (2018), A3267–A3292.

[54] T. Y. Hou and X. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, *Journal of Computational Physics*, **134** (1) (1997), 169–189.

[55] A. Huerta, E. Nadal, and F. Chinesta, Proper generalized decomposition solutions within a domain decomposition strategy, *International Journal for Numerical Methods in Engineering*, **113** (13) (1972–1994), 2018.

[56] T. J. R. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, *Computer Methods in Applied Mechanics and Engineering*, **127** (1-4) (1995), 387–401.

[57] T. J. R. Hughes, G. Engel, L. Mazzei, and M. G. Larson, The continuous Galerkin method is locally conservative, *Journal of Computational Physics*, **163** (2) (2000), 467–488.

[58] T. J. R. Hughes, G. R. Feijóo, L. Mazzei, and J.-B. Quincy, The variational multiscale method – a paradigm for computational mechanics, *Computer Methods in Applied Mechanics and Engineering*, **166** (1-2) (1998), 3–24.

[59] W. C. Hurty, Dynamic analysis of structural systems using component modes, *AIAA Journal*, **3** (4) (1965), 678–685.

[60] D. B. P. Huynh, D. J. Knezevic, and A. T. Patera, A static condensation reduced basis element method: complex problems, *Computer Methods in Applied Mechanics and Engineering*, **259** (2013), 197–216.

[61] D. B. P. Huynh, D. J. Knezevic, and A. T. Patera, A static condensation reduced basis element method: approximation and *a posteriori* error estimation, *ESAIM: Mathematical Modelling and Numerical Analysis*, **47** (1) (2013), 213–251.

[62] L. Iapichino, A. Quarteroni, and G. Rozza, A reduced basis hybrid method for the coupling of parametrized domains represented by fluidic networks, *Computer Methods in Applied Mechanics and Engineering*, **221/222** (2012), 63–82.

[63] L. Iapichino, A. Quarteroni, and G. Rozza, Reduced basis method and domain decomposition for elliptic problems in networks and complex parametrized geometries, *Computers & Mathematics with Applications*, **71** (1) (2016), 408–430.

[64] H. Jakobsson, F. Bengzon, and M. G. Larson, Adaptive component mode synthesis in linear elasticity, *International Journal for Numerical Methods in Engineering*, **86** (7) (2011), 829–844.

[65] S. Kaulmann, B. Flemisch, B. Haasdonk, K.-A. Lie, and M. Ohlberger, The localized reduced basis multiscale method for two-phase flows in porous media, *International Journal for Numerical Methods in Engineering*, **102** (5) (2015), 1018–1040.

[66] S. Kaulmann, M. Ohlberger, and B. Haasdonk, A new local reduced basis discontinuous Galerkin approach for heterogeneous multiscale problems, *Comptes Rendus. Mathématique*, **349** (23-24) (2011), 1233–1238.

[67] A. Klawonn, P. Radtke, and O. Rheinbach, A comparison of adaptive coarse spaces for iterative substructuring in two dimensions, *Electronic Transactions on Numerical Analysis*, **45** (2016), 75–106.

[68] A. Kolmogoroff, Über die beste Annäherung von Funktionen einer gegebenen Funktionenklasse, *Annals of Mathematics*, **37** (1) (1936), 107–110.

[69] R. Kornhuber, D. Peterseim, and H. Yserentant, An analysis of a class of variational multiscale methods based on subspace decomposition, *Mathematics of Computation*, **87** (314) (2018), 2765–2774.

[70] R. Kornhuber, J. Podlesny, and H. Yserentant, Direct and iterative methods for numerical homogenization, in *Domain Decomposition Methods in Science and Engineering XXIII*. Lect. Notes Comput. Sci. Eng., vol. 116, pp. 217–225, Springer, Cham, 2017.

[71] R. Kornhuber and H. Yserentant, Numerical homogenization of elliptic multiscale problems by subspace decomposition, *Multiscale Modeling & Simulation*, **14** (3) (2016), 1017–1036.

[72] M. G. Larson and A. Malqvist, Adaptive variational multiscale methods based on a posteriori error estimation: duality techniques for elliptic problems, in *Multiscale Methods in Science and Engineering*. Lect. Notes Comput. Sci. Eng., vol. 44, pp. 181–193, Springer, Berlin, 2005.

[73] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide*. Software, Environments, and Tools, vol. 6, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998 Solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods.

[74] A. E. Løvgren, Y. Maday, and E. M. Rønquist, A reduced basis element method for the steady stokes problem, *ESAIM: Mathematical Modelling and Numerical Analysis*, **40** (3) (2006), 529–552.

[75] Y. Maday, O. Mula, A. T. Patera, and M. Yano, The generalized empirical interpolation method: stability theory on Hilbert spaces with an application to the Stokes equation, *Computer Methods in Applied Mechanics and Engineering*, **287** (2015), 310–334.

[76] Y. Maday and E. M. Rønquist, A reduced-basis element method, *Journal of Scientific Computing*, **17** (1-4) (2002), 447–459.

[77] Y. Maday and E. M. Rønquist, The reduced basis element method: application to a thermal fin problem, *SIAM Journal on Scientific Computing*, **26** (1) (2004), 240–258, electronic.

[78] M. W. Mahoney, Randomized algorithms for matrices and data, *Foundations and Trends in Machine Learning*, **3** (2) (February 2011), 123–224.

[79]   M. W. Mahoney and P. Drineas, CUR matrix decompositions for improved data analysis, *Proceedings of the National Academy of Sciences of the United States of America*, **106** (3) (2009), 697–702.

[80]   I. Maier and B. Haasdonk, A Dirichlet–Neumann reduced basis method for homogeneous domain decomposition problems, *Applied Numerical Mathematics*, **78** (2014), 31–48.

[81]   A. Malqvist and D. Peterseim, Localization of elliptic multiscale problems, *Mathematics of Computation*, **83** (290) (2014), 2583–2603.

[82]   J. Mandel and B. Sousedík, Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods, *Computer Methods in Applied Mechanics and Engineering*, **196** (8) (2007), 1389–1399.

[83]   I. Martini, G. Rozza, and B. Haasdonk, Reduced basis approximation and a-posteriori error estimation for the coupled Stokes-Darcy system, *Advances in Computational Mathematics*, **41** (5) (2015), 1131–1157.

[84]   R. Milk, S. Rave, and F. Schindler, pyMOR—generic algorithms and interfaces for model order reduction. *SIAM Journal on Scientific Computing*, **38** (5) (2016), S194–S216.

[85]   M. Ohlberger, A posteriori error estimates for the heterogeneous multiscale finite element method for elliptic homogenization problems, *Multiscale Modeling & Simulation*, **4** (1) (2005), 88–114.

[86]   M. Ohlberger and S. Rave, Localized reduced basis approximation of a nonlinear finite volume battery model with resolved electrode geometry, in *Model Reduction of Parametrized Systems*. MS&A. Model. Simul. Appl., vol. 17, pp. 201–212, Springer, 2017.

[87]   M. Ohlberger, S. Rave, and F. Schindler, True error control for the localized reduced basis method for parabolic problems, in *Model Reduction of Parametrized Systems*. MS&A. Model. Simul. Appl., vol. 17, pp. 169–182, Springer, Cham, 2017.

[88]   M. Ohlberger, M. Schaefer, and F. Schindler, *Localized Model Reduction in PDE Constrained Optimization*, pp. 143-163, Springer International Publishing, Cham, 2018.

[89]   M. Ohlberger and F. Schindler, A-posteriori error estimates for the localized reduced basis multi-scale method, in *Finite Volumes for Complex Applications VII. Methods and Theoretical Aspects*. Springer Proc. Math. Stat., vol. 77, pp. 421–429, Springer, Cham, 2014.

[90]   M. Ohlberger and F. Schindler, Error control for the localized reduced basis multiscale method with adaptive on-line enrichment, *SIAM Journal on Scientific Computing*, **37** (6) (2015), A2865–A2895.

[91]   M. Ohlberger and F. Schindler, Non-conforming localized model reduction with online enrichment: towards optimal complexity in PDE constrained optimization, in *FiNite Volumes for Complex Applications VIII—Hyperbolic, Elliptic and Parabolic Problems*. Springer Proc. Math. Stat., vol. 200, pp. 357–365, 2017.

[92]   M. Ohlberger and K. Smetana, A dimensional reduction approach based on the application of reduced basis methods in the framework of hierarchical model reduction, *SIAM Journal on Scientific Computing*, **36** (2) (2014), A714–A736.

[93]   P. Pacciarini, P. Gervasio, and A. Quarteroni, Spectral based discontinuous Galerkin reduced basis element method for parametrized Stokes problems, *Computers & Mathematics with Applications*, **72** (8) (1977–1987), 2016.

[94]   S. Perotto, A. Ern, and A. Veneziani, Hierarchical local model reduction for elliptic problems: a domain decomposition approach, *Multiscale Modeling & Simulation*, **8** (4) (2010), 1102–1127.

[95]   A. Pinkus, *n-Widths in Approximation Theory, volume 7*, Springer-Verlag, Berlin, 1985.

[96]   M. Presho and S. Ye, Reduced-order multiscale modeling of nonlinear *p*-Laplacian flows in high-contrast media, *Computational Geosciences*, **19** (4) (2015), 921–932.

[97]   A. Quarteroni, A. Manzoni, and F. Negri, *Reduced Basis Methods for Partial Differential Equations*. Unitext, vol. 92, Springer, Cham, 2016. An introduction, La Matematica per il 3+2.

[98]   A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*. Numerical Mathematics and Scientific Computation, The Clarendon Press, Oxford University Press, New York, reprint, 2005.

[99]   J. Schleuß, Optimal local approximation spaces for parabolic problems. Master's thesis, University of Münster, 2019.

[100] K. Smetana, A new certification framework for the port reduced static condensation reduced basis element method, *Computer Methods in Applied Mechanics and Engineering*, **283** (2015), 352–383.

[101] K. Smetana, Static condensation optimal port/interface reduction and error estimation for structural health monitoring, in B. Haasdonk and J. Fehr (ed.), *IUTAM Symposium on Model Order Reduction of Coupled Systems, Stuttgart, Germany, May 22–25, 2018: MORCOS 2018*, 2019.

[102] K. Smetana and M. Ohlberger, Hierarchical model reduction of nonlinear partial differential equations based on the adaptive empirical projection method and reduced basis techniques, *ESAIM: Mathematical Modelling and Numerical Analysis*, **51** (2) (2017), 641–677.

[103] K. Smetana and A. T. Patera, Optimal local approximation spaces for component-based static condensation procedures, *SIAM Journal on Scientific Computing*, **38** (5) (2016), A3318–A3356.

[104] Society of Petroleum Engineers, SPE Comparative Solution Project, http://www.spe.org/web/csp/index.html, 2001.

[105] A. Sommer, O. Farle, and R. Dyczij-Edlinger, A new method for accurate and efficient residual computation in adaptive model-order reduction, *IEEE Transactions on Magnetics*, **51** (3) (2015), 1–4.

[106] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, and R. Scheichl, Achieving robustness through coarse space enrichment in the two level Schwarz framework, in *Domain Decomposition Methods in Science and Engineering XXI*. Lect. Notes Comput. Sci. Eng., vol. 98, pp. 447–455, Springer, Cham, 2014.

[107] S. Stepanov, M. Vasilyeva, and V. I. Vasil'ev, Generalized multiscale discontinuous Galerkin method for solving the heat problem with phase change, *Journal of Computational and Applied Mathematics*, **340** (2018), 645–652.

[108] T. Taddei, *Model order reduction methods for data assimilation; state estimation and structural health monitoring*. PhD thesis, Massachusetts Insitute of Technology, 2016.

[109] T. Taddei and A. T. Patera, A localization strategy for data assimilation; application to state estimation and parameter estimation, *SIAM Journal on Scientific Computing*, **40** (2) (2018), B611–B636.

[110] A. Toselli and O. Widlund, *Domain Decomposition Methods—Algorithms and Theory*, Springer Series in Computational Mathematics, vol. 34, Springer-Verlag, Berlin, 2005.

[111] 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*, vol. 3847, 2003.

[112] M. Vogelius and I. Babuška, On a dimensional reduction method. I. The optimal selection of basis functions, *Mathematics of Computation*, **37** (155) (1981), 31–46.

[113] W. Wang and M. N. Vouvakis, Randomized omputations in domain decomposition methods, in *2015 IEEE International Symposium on Antennas and Propagation USNC/URSI National Radio Science Meeting*, pp. 177–178, July 2015.

[114] E. Weinan and B. Engquist, The heterogeneous multiscale methods, *Communications in Mathematical Sciences*, **1** (1) (2003), 87–132.

[115] E. Weinan and B. Engquist, The heterogeneous multi-scale method for homogenization problems, in *Multiscale Methods in Science and Engineering*. Lect. Notes Comput. Sci. Eng., vol. 44, pp. 89–110, Springer, Berlin, 2005.
[116] S. Wu Fung and L. Ruthotto, A multiscale method for model order reduction in PDE parameter estimation, *Journal of Computational and Applied Mathematics*, **350** (2019), 19–34.

Steven L. Brunton and J. Nathan Kutz

7 Data-driven methods for reduced-order modeling

Abstract: Data-driven mathematical methods are increasingly important for characterizing complex systems across the physical, engineering, and biological sciences. These methods aim to discover and exploit a relatively small subset of the full high-dimensional state space where low-dimensional models can be used to describe the evolution of the system. Emerging dimensionality reduction methods, such as the *dynamic mode decomposition* (DMD) and its Koopman generalization, have garnered attention due to the fact that they can (i) discover low-rank spatio-temporal patterns of activity, (ii) embed the dynamics in the subspace in an equation-free manner (i. e., the governing equations are unknown), unlike Galerkin projection onto proper orthogonal decomposition modes, and (iii) provide approximations in terms of linear dynamical systems, which are amenable to simple analysis techniques. The selection of observables (features) for the DMD/Koopman architecture can yield accurate low-dimensional embeddings for nonlinear partial differential equations (PDEs) while limiting computational costs. Indeed, a good choice of observables, including time delay embeddings, can often *linearize* the nonlinear manifold by making the spatio-temporal dynamics weakly nonlinear. In addition to DMD/Koopman decompositions, coarse-grained models for spatio-temporal systems can also be discovered using the *sparse identification of nonlinear dynamics* (SINDy) algorithm which allows one to construct reduced-order models in low-dimensional embeddings. These methods can be used in a nonintrusive, equation-free manner for improved computational performance on parametric PDE systems.

Keywords: reduced-order model, dynamic mode decomposition, Koopman theory, nonlinear system identification, sparse regression, system identification

PACS: 02.30.Hq, 02.30.Jr, 02.60.Cb, 02.30.Mv

7.1 Introduction

Data-driven modeling of complex systems is of increasing importance in modern scientific applications given the unprecedented rise of data collection on multi-scale, spatio-temporal systems. Enabled by emerging sensor technologies and high-performance computing platforms, the large-scale monitoring and collection of data

Steven L. Brunton, Department of Mechanical Engineering, University of Washington, Seattle, WA 98195, USA
J. Nathan Kutz, Department of Applied Mathematics, University of Washington, Seattle, WA 98195, USA

on such systems has shifted our modeling paradigm by exploiting data-driven, machine learning approaches. Specifically, instead of positing empirical or approximate spatio-temporal models, typically characterized by partial differential equations (PDEs), the low-dimensional features extracted from time snapshots of the data can be directly used to construct *reduced-order models* (ROMs) for a variety of important tasks such as state-space reconstruction and diagnostics, as well as future state prediction and forecasting [14]. In this chapter, we present a diverse set of data-driven methods that can be used to construct ROMs directly from data. The methods presented can be used with traditional ROM architectures where the governing PDEs are known, or they can be used to discover unknown spatio-temporal dynamics directly from the data. Most of the methods are nonintrusive, minimizing the need for prohibitively expensive high-performance simulations. This also allows for accurate, low-fidelity models, enabling inexpensive Monte Carlo simulations. We present four methods for enabling data-driven ROMs: The *dynamic mode decomposition* (DMD) and the associated Koopman decomposition [78], the *sparse identification of nonlinear dynamics* (SINDy) algorithm [25], and the *Hankel alternative view of Koopman* (HAVOK) algorithm [22]. Each method can be used to advantage in a variety of situations, including when the governing PDE equations are known, only partially known, or unknown.

Consider a governing system of nonlinear PDEs of a single spatial variable x, which can be modeled as [14]

$$\mathbf{u}_t = \mathbf{L}(x)\mathbf{u} + \mathbf{N}(\mathbf{u}, x, t), \tag{7.1}$$

where $\mathbf{L}(x)$ is a linear operator and $\mathbf{N}(\cdot)$ prescribes the nonlinear terms in the evolution dynamics. Both $\mathbf{L}(x)$ and $\mathbf{N}(\cdot)$ may be unknown, or only partially known. As an example, the Burgers equation, $u_t = u_{xx} + vuu_x$, has $L = \partial^2/\partial x^2$ and $N(u) = vuu_x$. Associated with (7.1) are a set of initial and boundary conditions on a domain $x \in \mathcal{D}$. Historically, a number of analytic solution techniques have been devised to study (7.1) provided the right-hand side is known. Typically the aim of such methods is to reduce the PDE (7.1) to a set of ordinary differential equations (ODEs). The standard PDE methods of *separation of variables* and *similarity solutions* are constructed for this express purpose. Once in the form of an ODE, a broader variety of analytic methods can be applied along with a *qualitative theory* in the case of nonlinear behavior [61]. This again highlights the role that *asymptotics* can play in characterizing behavior.

For the general form of (7.1) where the right-hand side is known, separation of variables can often be used to yield a computational algorithm capable of producing low-rank approximations. Since the spatial solutions are not known a priori, it is typical to assume a set of basis modes which can be used for the low-rank approximation. Indeed, such assumptions on basis modes underly the critical ideas of the method of *eigenfunction expansions*. This yields a separation of variables solution ansatz of the

form

$$\mathbf{u}(\mathbf{x}, t) = \mathbf{\Psi}(\mathbf{x})\mathbf{a}(t) = \sum_{k=1}^{r} \psi_k(\mathbf{x})a_k(t), \tag{7.2}$$

where $\mathbf{\Psi}(\mathbf{x}) \in \mathbb{C}^{n \times r}$ form a set of r orthonormal basis modes and $\mathbf{x} \in \mathbb{R}^{n \times 1}$ represents the spatial discretization of x in the governing PDE. The modal basis $\mathbf{\Psi}$ is often obtained via *proper orthogonal decomposition* (POD) [60, 14, 129]. This separation of variables solution approximates the true solution, provided r is large enough. A fundamental assumption of reduced-order modeling is that there exists a low-rank truncation, or subspace, that accurately characterizes the evolution of the spatio-temporal system. More broadly, such approximations are based upon modal methods for building ROMs as discussed in Chapters 1 and 4 of Volume 1 of *Model order reduction* [12].

The orthogonality properties of the basis functions $\psi_k(x)$, which are the columns of $\mathbf{\Psi}$, enable us to make use of (7.2). Inserting the expansion (7.2) into the governing equations gives [14]

$$\frac{d\mathbf{a}}{dt} = \mathbf{\Psi}^T \mathbf{L}\mathbf{\Psi}\mathbf{a} + \mathbf{\Psi}^T \mathbf{N}(\mathbf{\Psi}\mathbf{a}). \tag{7.3}$$

The given form of $\mathbf{N}(\cdot)$ determines the mode-coupling that occurs between the various r modes. Indeed, the hallmark feature of nonlinearity is the production of modal mixing from (7.3). Equation (7.3) is the canonical ROM identified as a Galerkin projection of the dynamics onto POD modes. It can be evaluated given full knowledge of the right-hand side of the governing PDE.

Equation (7.3) details how a low-rank subspace can be used to construct a Galerkin-POD-ROM model as a proxy, or surrogate, model for the high-fidelity model. In this reduction, the linear operator $\mathbf{\Psi}^T \mathbf{L}\mathbf{\Psi}$ can be computed once to produce an $r \times r$ matrix modeling the effects of the linear portion of the dynamics. What is more problematic is the evaluation of the nonlinear contribution $\mathbf{\Psi}^T \mathbf{N}(\mathbf{\Psi}\mathbf{a})$ in (7.3). Indeed, one of the primary challenges in producing the low-rank dynamical system is efficiently projecting the nonlinearity of the governing PDEs on the POD basis. This fact was recognized early on in the ROM community, and methods such as gappy POD [50, 142, 150] were proposed to more efficiently enable this hyperreduction task. More recently, the *empirical interpolation method* (EIM) [11], the *discrete EIM* (DEIM) [37], and the QR decomposition-based Q-DEIM [46], have provided a computationally efficient method for discretely (sparsely) sampling and evaluating the nonlinearity. These widely used hyperreduction methods ensure that the computational complexity of ROMs scale favorably with the rank of the approximation, even for complex nonlinearities.

Numerical schemes based on the Galerkin projection (7.3) are commonly used to perform simulations of the full governing system (7.1). Convergence to the true solution can be accomplished by judicious choice of both the modal basis elements $\mathbf{\Psi}$ and the total number of modes r. Interestingly, the separation of variables strategy, which is

rooted in *linear* PDEs, works for *nonlinear* and *nonconstant coefficient* PDEs, provided enough modal basis functions are chosen in order to capture the nonlinear mode mixing that occurs in (7.3). A good choice of modal basis elements allows for a smaller set of r modes to be chosen to achieve a desired accuracy. The POD method is designed to specifically address the data-driven selection of a set of basis modes that are tailored to the particular dynamics, geometry, and parameters.

Unfortunately, the Galerkin-POD projection of the dynamics (7.3) is often unstable [34], requiring modification to stabilize the time-stepping scheme [4]. Moreover, the evaluation in (7.3) of the nonlinear term $\mathbf{\Psi}^T \mathbf{N}(\mathbf{\Psi a})$ renders the ROM scheme intrusive, i. e., to compute the nonlinear contribution in the low-rank subspace requires an expensive sampling of the high-fidelity model in order to build the low-rank subspace. Instead, one can directly approximate the nonlinearity via DMD which directly computes this contribution via nonintrusive methods [2]. Thus there is no recourse to high-fidelity and expensive computations to construct an approximation to the nonlinear contribution. If latent variables are present, i. e., important portions of the state-space have not been measured, then the Hankel alternative view of Koopman (HAVOK) algorithm, which helps to discover a proxy for the latent variable space, can be used instead of DMD. Finally, if the right-hand side is unknown, then the SINDy algorithm can be used to discover a low-rank, nonlinear model characterizing the evolution. The diversity of mathematical techniques highlights the emerging use of regression and machine learning strategies that can help model complex, spatio-temporal systems.

7.2 Data-driven reductions

Numerical linear algebra plays a central role in scientific computing [135, 77, 24]. Specifically, methods that have historically improved the efficiency of solving $\mathbf{Ax} = \mathbf{b}$ have always been of critical importance for tractable computations, especially at scale, where *scale* is a relative term associated with the limits of computing in a given era. From QR decomposition to lower-upper factorization [135], matrix decompositions have been the primary methods to enable improved computational efficiency. But perhaps the most important factorization technique is the *singular value decomposition* (SVD) [77], which plays a key role in data analysis and computation, including applications in reduced-order modeling through POD and DMD.

7.2.1 Singular value decomposition

The success of the SVD algorithm is largely due to the fact that by construction, it extracts the dominant, correlated features from any given data matrix. This often allows one to approximate the matrix with a principled low-rank approximation which

is guaranteed to be the best approximation in an ℓ_2-sense. This mathematical architecture is so powerful and universal that it has been invented and used extensively in a wide range of fields [77]. Specifically, it is alternatively known as *principal component analysis* (PCA) in statistics (where to be precise, each column or row is scaled to have mean zero and unit variance), *POD* in the fluid dynamics community, *empirical mode decomposition* in atmospheric sciences, the *Hotelling transform*, *empirical eigenfunctions*, or *Karhunen-Loève decomposition*. Thus, from seemingly complex data from which a matrix is composed, a low-dimensional subspace can be computed on which the majority of the data resides.

The SVD of a matrix $\mathbf{X} \in \mathbb{C}^{n \times m}$ takes the form

$$\mathbf{X} = \mathbf{U \Sigma V}^*, \tag{7.4}$$

in terms of the following three matrices:

$$\mathbf{U} \in \mathbb{C}^{n \times n} \text{ is unitary,} \tag{7.5a}$$

$$\mathbf{V} \in \mathbb{C}^{m \times m} \text{ is unitary,} \tag{7.5b}$$

$$\mathbf{\Sigma} \in \mathbb{R}^{n \times m} \text{ is diagonal.} \tag{7.5c}$$

Additionally, it is assumed that the diagonal entries of $\mathbf{\Sigma}$ are nonnegative and ordered from largest to smallest so that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$, where $p = \min(m, n)$. The SVD of the matrix $\mathbf{X}$ thus shows that the matrix first applies a unitary transformation preserving the unit sphere via $\mathbf{V}^*$. This is followed by a stretching operation that creates an ellipse with principal semi-axes given by the matrix $\mathbf{\Sigma}$. Finally, the generated hyperellipse is rotated by the unitary transformation $\mathbf{U}$. Thus the image of a unit sphere under any $n \times m$ matrix is a hyperellipse. The following is the primary theorem concerning SVD [135].

Theorem. *Every matrix* $\mathbf{X} \in \mathbb{C}^{n \times m}$ *has an SVD* (7.4). *Furthermore, the singular values* $\{\sigma_j\}$ *are uniquely determined, and if* $\mathbf{X}$ *is square and* σ_j *is distinct, the singular vectors* $\{\mathbf{u}_j\}$ *and* $\{\mathbf{v}_j\}$ *are uniquely determined up to complex signs (complex scalar factors of absolute value* 1*).*

The above theorem guarantees the existence of the SVD, but in practice, it still remains to be computed. This is a fairly straightforward process if one considers the following matrix products:

$$\mathbf{X}^* \mathbf{X} = \left(\mathbf{U \Sigma V}^* \right)^* \left(\mathbf{U \Sigma V}^* \right) = \mathbf{V \Sigma}^2 \mathbf{V}^* \tag{7.6}$$

and

$$\mathbf{X X}^* = \left(\mathbf{U \Sigma V}^* \right) \left(\mathbf{U \Sigma V}^* \right)^* = \mathbf{U \Sigma}^2 \mathbf{U}^*. \tag{7.7}$$

Multiplying (7.6) and (7.7) on the right by **V** and **U**, respectively, gives the two self-consistent eigenvalue problems

$$\mathbf{X}^*\mathbf{X}\mathbf{V} = \mathbf{V}\mathbf{\Sigma}^2, \tag{7.8a}$$

$$\mathbf{X}\mathbf{X}^*\mathbf{U} = \mathbf{U}\mathbf{\Sigma}^2. \tag{7.8b}$$

Thus if the normalized eigenvectors are found for these two equations, then the orthonormal basis vectors are produced for **U** and **V**. Likewise, the square root of the eigenvalues of these equations produces the singular values σ_j.

Theorem (Schmidt–Eckart–Young–Mirsky theorem [118, 47, 96]). *For any N so that $0 \le N \le p = \min\{m, n\}$, we can define the partial sum*

$$\mathbf{X}_N = \sum_{j=1}^{N} \sigma_j \mathbf{u}_j \mathbf{v}_j^* . \tag{7.9}$$

And if $N = \min\{m, n\}$, we define $\sigma_{N+1} = 0$. Then

$$\|\mathbf{X} - \mathbf{X}_N\|_2 = \sigma_{N+1} . \tag{7.10}$$

Likewise, if using the Frobenius norm, then

$$\|\mathbf{X} - \mathbf{X}_N\|_F = \sqrt{\sigma_{N+1}^2 + \sigma_{N+2}^2 + \cdots + \sigma_p^2} . \tag{7.11}$$

The interpretation of this theorem is critical as it gives a geometrical perspective for understanding the SVD. Geometrically, the SVD gives *the best approximation of a hyperellipsoid by a line segment*, i. e., simply take the line segment to be the longest axis, i. e., that associated with the singular value σ_1. Continuing this idea, what is the best approximation by a two-dimensional ellipse? Take the longest and second longest axes, i. e., those associated with the singular values σ_1 and σ_2. After r steps, the total energy in **X** is completely captured. Thus the SVD gives an algorithm for a least-squares fit allowing us to project the matrix onto low-dimensional representations in a formal, algorithmic way.

The SVD provides a principled way to find a low-rank subspace on which to project the evolution dynamics of the PDE in (7.1). Specifically, the first r modes of a low-rank projection form the POD basis in (7.2) desired for model reduction

$$\mathbf{\Psi} = \mathbf{U}_r. \tag{7.12}$$

These basis modes are used to project the dynamics onto the dominant, low-rank subspace of activity as shown in (7.3). Of course, to use these POD modes, the dynamics of the governing PDE must be known in advance. Moreover, the Galerkin projection of the dynamics onto POD modes in (7.3) may be, depending on the underlying problem, unstable, requiring modification to stabilize the time-stepping scheme. Such stability

issues have been considered extensively by Carlberg and co-workers [34], Amsallem and Farhat [4], and Kalashnikova et al. [67]. Regardless, POD reductions arising from the SVD computation of the basis $\mathbf{\Psi}$ form the underpinnings of many ROMs [60, 14]. A significant advantage of the maturity of POD-based reductions is the ability to produce rigorous error bounds. Indeed, there is a rich literature on how to use the error properties of POD/SVD to derive rigorous error bounds for simulation as well as optimal control across a diverse set of applications [75, 76, 143, 110, 59, 136]. Such rigorous bounds provide trust regions and certifiable models for many critical application areas. Details of POD-based ROMs can be found in Chapter 2 of the current volume on *Model order reduction* [13].

Due to tremendous advances and innovations, modern large-scale simulations and/or the data collection process can quickly produce volumes of data that traditional methods could not easily analyze and diagnose in real-time. This emerging *big data* era requires diagnostic tools that can scale to meet the rapidly growing information acquired from new monitoring technologies which are producing increasingly fine-scale spatial and temporal measurements. In such cases, one can exploit new techniques that are capable of extracting the dominant global features of the high-dimensional dynamics. Specifically, emerging *randomized linear algebra* algorithms [55, 85, 48] are critically enabling for scalable *big data* applications, supplementing the *method of snapshots* [122] for efficient computation of the SVD. Randomized algorithms exploit the fact that the data themselves have low-rank features. Indeed, the method scales with the intrinsic rank of the dynamics rather than the dimension of the measurements/sensor space. This is in contrast to standard SVD/PCA/POD reductions which do not scale well with the data size. One can think of the scalable methods as being critically enabling for producing real-time analysis of emerging, streaming *big data* sets. Moreover, the dominant features of the data can be used for an efficient compression of the data for storage or reduced-order modeling applications [3]. Figure 7.1 outlines the basic algorithmic architecture which can be used for producing scalable SVD decompositions.

7.2.2 Dynamic mode decomposition

An alternative to POD is the DMD reduction [117]. Unlike POD, the DMD algorithm not only correlates spatial activity, but also enforces that various low-rank spatial modes be correlated in time, essentially merging the favorable aspects of POD/PCA in space and the Fourier transform in time. DMD is a matrix factorization method based upon the SVD algorithm. However, in addition to performing a low-rank SVD approximation, it further performs an eigendecomposition on a best-fit linear operator that advances measurements forward in time in the computed subspaces in order to extract critical temporal features. Thus the DMD method provides a spatio-temporal decomposition of data into a set of dynamic modes that are derived from snapshots or mea-
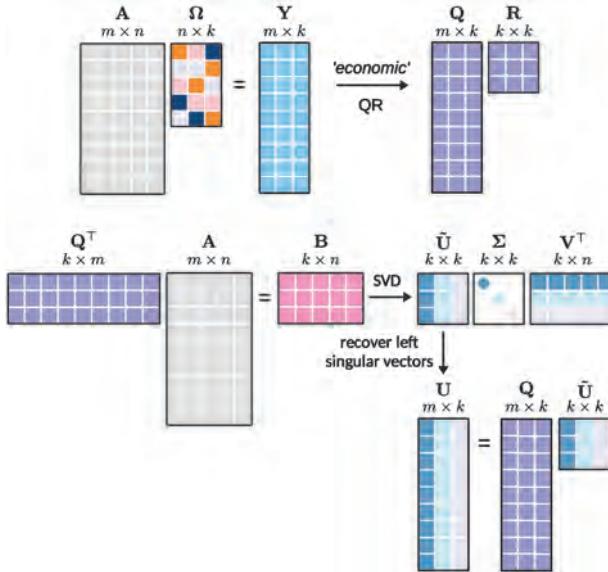
Figure 7.1: Illustration of the randomized matrix decomposition technique for scalable decompositions. The random sampling matrix **Ω** is used to produce a new matrix **Y** which can be decomposed using a QR decomposition. This leads to the construction of the matrix **B** which is used for approximating the left and right singular vector. *From Erichson et al.* [48].

surements of a given system in time, arranged as column state-vectors. The mathematics underlying the extraction of dynamic information from time-resolved snapshots is closely related to the idea of the Arnoldi algorithm, one of the workhorses of fast computational solvers. The DMD algorithm was originally designed to collect data at regularly spaced intervals of time. However, new innovations allow for both sparse spatial [27, 54] and temporal [139] collection of data as well as irregularly spaced collection times [6].

Like SVD, the DMD algorithm is based upon a regression. Thus there are a variety of algorithms that have been proposed in the literature for computing the DMD. A highly intuitive understanding of the DMD architecture was proposed by Tu et al. [138], which provides the *exact DMD* method.

Definition: Exact dynamic mode decomposition (Tu et al. 2014 [138]). Suppose we have a dynamical system (7.1) and two sets of measurement data

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_{m-1} \\ | & | & & | \end{bmatrix}, \tag{7.13a}$$

$$\mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{u}'_1 & \mathbf{u}'_2 & \cdots & \mathbf{u}'_{m-1} \\ | & | & & | \end{bmatrix} \tag{7.13b}$$

so that $\mathbf{u}'_k = \mathbf{F}(\mathbf{u}_k)$, where $\mathbf{F}$ is the map corresponding to the evolution of (7.1) for time Δt. Exact DMD computes the leading eigendecomposition of the best-fit linear operator $\mathbf{A}$ relating the data $\mathbf{u}' \approx \mathbf{A}\mathbf{u}$:

$$\mathbf{A} = \mathbf{X}'\mathbf{X}^{\dagger}. \tag{7.14}$$

The DMD modes, also called dynamic modes, are the eigenvectors of $\mathbf{A}$, and each DMD mode corresponds to a particular eigenvalue of $\mathbf{A}$.

The DMD framework takes an equation-free perspective where the original, nonlinear dynamics may be unknown. Thus measurements of the system alone are used to approximate the dynamics and predict the future state. However, DMD can also be used when governing equations are known [2]. This DMD-Galerkin procedure represents a potential hybrid between the POD-Galerkin and DMD methods. The integration of DMD and POD can also be used for model reduction numerical schemes [146]. The DMD procedure constructs a proxy, locally linear dynamical system approximation to (7.1):

$$\mathbf{u}_{k+1} \approx \mathbf{A}\mathbf{u}_k, \tag{7.15}$$

whose well-known solution is

$$\mathbf{u}_k = \sum_{j=1}^{n} \boldsymbol{\phi}_j \lambda_j^k b_j = \boldsymbol{\Phi}\boldsymbol{\Lambda}^k \mathbf{b}, \tag{7.16}$$

where $\boldsymbol{\phi}_j$ and λ_j are the eigenvectors and eigenvalues of the matrix $\mathbf{A}$, and the coefficients b_j are the coordinates of the initial condition $\mathbf{u}_0$ in the eigenvector basis. The eigenvalues λ of $\mathbf{A}$ determine the temporal dynamics of the system, at least in an asymptotic sense and for normal operators, i. e., transient dynamics are not well captured. It is often convenient to convert these eigenvalues to continuous time, $\omega = \log(\lambda)/\Delta t$, so the real parts of the eigenvalues ω determine growth and decay of the solution, and the imaginary parts determine oscillatory behaviors and their corresponding frequencies. The eigenvalues and eigenvectors are critically enabling for producing interpretable diagnostic features of the dynamics. It is important to note that the choice of the time step Δt is critical in the DMD algorithm. The time step must be small enough to resolve the fastest time scales of relevance. A consequence of the linear model produced by the DMD algorithm is its inability to model transient phenomena over the snapshots sampled, aside from transient growth potentially produced by nonnormal modes where eigenvalues are identical or nearly so.

The DMD algorithm produces a low-rank eigendecomposition of the matrix $\mathbf{A}$ that optimally fits the measured trajectory $\mathbf{u}_k$ for $k = 1, 2, \ldots, m$ snapshots in a least-squares sense so that $\|\mathbf{u}_{k+1} - \mathbf{A}\mathbf{u}_k\|_2$ is minimized across all points for $k = 1, 2, \ldots, m-1$. The optimality of the approximation holds only over the sampling window where $\mathbf{A}$ is constructed, and the approximate solution can be used to not only make future

state predictions, but also to derive dynamic modes critical for diagnostics. Indeed, in much of the literature where DMD is applied, it is primarily used as a diagnostic tool. This is much like POD analysis, where the POD modes are also primarily used for diagnostic purposes. Thus the DMD algorithm can be thought of as a modification of the SVD architecture which attempts to account for dynamic activity of the data. The eigendecomposition of the low rank space found from SVD enforces a Fourier mode time expansion which allows one to then make spatio-temporal correlations with the sampled data. Recently, DMD has also been rigorously connected to the spectral POD method [133].

Early variants of the DMD-computed eigenvalues that were biased by the presence of sensor noise [58, 44]. This was a direct result of the fact that the standard algorithms treated the data in a pairwise sense and favored the forward direction in time. Dawson et al. [44] and Hemati et al. [58] developed several methods for debiasing within the standard DMD framework. These methods have the advantage that they can be computed with essentially the same set of robust and fast tools as the standard DMD. As an alternative, the *optimized DMD* advocated by [38] treats all of the snapshots of the data at once. This avoids much of the bias of the original DMD but requires the solution of a potentially large nonlinear optimization problem. Askham and Kutz [6] recently showed that the optimized DMD algorithm could be rendered numerically tractable by leveraging the classical variable projection method [53]. Moreover, the optimized DMD method can be used to enforce all eigenvalues to have a real part less than or equal to zero. This ensures stability of solutions for future times as there are no growing modes. For input-output systems, DMD has also been modified through a postprocessing algorithm to generate a stable input-output model [15]. These methods show that DMD architectures can be imbued with advantageous stability properties for ROMs.

The variable projection algorithm is based upon the observation that the desired solutions of DMD are exponentials (7.16). Thus DMD is reformulated as an exponential data fitting (specifically, for inverse differential equations), an area of research that has been extensively developed and has many applications [52, 104]. The variable projection method leverages the special structure of the exponential data fitting problem, so that many of the unknowns may be eliminated from the optimization. An additional benefit of these tools is that the snapshots of data no longer need to be taken at regular intervals, i. e., the sample times do not need to be equispaced. The goal is then to rewrite the data matrix of snapshots as

$$\mathbf{X}^{\mathsf{T}} \approx \mathbf{\Phi}(\boldsymbol{\alpha})\mathbf{B} \, , \tag{7.17}$$

where $\mathbf{\Phi}(\boldsymbol{\alpha}) \in \mathbb{C}^{m \times r}$ with entries defined by $\Phi(\boldsymbol{\alpha})_{i,j} = \exp(\alpha_j t_i)$.

The preceding leads us to the following definition of the optimized DMD in terms of an exponential fitting problem. Suppose that $\hat{\boldsymbol{\alpha}}$ and $\hat{\mathbf{B}}$ solve

$$\text{minimize} \|\mathbf{X}^{\mathsf{T}} - \mathbf{\Phi}(\boldsymbol{\alpha})\mathbf{B}\|_F \quad \text{over } \boldsymbol{\alpha} \in \mathbb{C}^k, \mathbf{B} \in \mathbb{C}^{l \times n} \, . \tag{7.18}$$

The optimized DMD eigenvalues are then defined by $\lambda_i = \hat{\alpha}_i$ and the eigenmodes are defined by

$$\boldsymbol{\varphi}_i = \frac{1}{\|\hat{\mathbf{B}}^\mathsf{T}(:,i)\|_2} \hat{\mathbf{B}}^\mathsf{T}(:,i) \, , \tag{7.19}$$

where $\hat{\mathbf{B}}^\mathsf{T}(:,i)$ is the i-th column of $\hat{\mathbf{B}}^\mathsf{T}$. Details of the algorithm and code for computing the optimized DMD can be found in Askham and Kutz [6]. The improved and debiased decomposition (7.16) of this optimal DMD strategy are readily apparent in numerous examples. Moreover, a comparison of DMD variants shows how each method handles noise and takes on bias. Optimized DMD thus far outperforms all other variants at the cost of a nonlinear optimization.

A remarkable feature of the DMD algorithm is its modularity for mathematical enhancements. Specifically, the DMD algorithm can be engineered to exploit sparse sampling [27, 54], it can be modified to handle inputs and actuation [106], it can be used to more accurately approximate the Koopman operator when using judiciously chosen functions of the state-space [80], and it can easily decompose data into multiscale temporal features in order to produce a multiresolution DMD [79]. Few mathematical architectures are capable of seamlessly integrating such diverse modifications of the dynamical system. But since the DMD provides an approximation of a linear system, such modifications are easily constructed. Moreover, the DMD algorithm, unlike many other machine learning algorithms, is not data-intensive in comparison to most deep neural network architectures which require large labeled data sets. Thus a DMD approximation can always be achieved, especially as the first step in the algorithm is the SVD which is guaranteed to exist for any data matrix. However, for very large data sets, DMD can leverage randomized methods [55, 85, 48] to produce a scalable *randomized DMD* [49, 18].

DMD is closely related to the field of system identification, which identifies models from data, often for use with model-based controllers. Tu et al. [138] and Proctor et al. [106] established connections between DMD and several classical system identification approaches, including the eigensystem realization algorithm [64] and singular spectrum analysis (SSA) [20] in climate time-series analysis. Nearly all methods of system identification involve some form of regression of data onto dynamics, and the main distinction between the various techniques is the degree to which this regression is constrained. For example, DMD generates best-fit linear models.

7.2.3 Koopman theory and observable selection

Much of the challenge associated with predicting, estimating, controlling, and reducing complex systems arises from the inherent nonlinearity in the governing equations. Indeed, mathematical physics has a rich history in deriving coordinate transformations that simplify the dynamics and alleviate the challenge of nonlinearity. In 1931,

Koopman developed an alternative perspective to classical dynamical systems theory, showing that there is a linear, infinite-dimensional operator that acts on the Hilbert space of possible measurement functions of the system, advancing these measurements along the flow of the dynamics [71, 72]. Koopman's operator-theoretic perspective trades nonlinear dynamics for linear but infinite-dimensional dynamics, and was critical in Birkhoff's proof of the ergodic theorem [17, 97].

Recently, Koopman operator theory has seen a resurgence of interest [93, 29, 94], in large part because of the increasing availability of measurement data and improving computational capabilities. In 2005, Mezic showed that Koopman theory may be used to provide a modal decomposition of complex systems, providing direct relevance to engineering systems [93]. Since then, it has been shown that the DMD algorithm from fluid dynamics [117] actually approximates the Koopman operator [109], restricted to a set of linear measurements of the system; a more detailed treatment for fluid systems is given by Taira et al. [129].

The ability of Koopman analysis to transform nonlinear systems into a linear framework has tremendous promise to make complex systems amenable to optimal prediction, estimation, and control with simple techniques from linear systems theory. In a short time, Koopman theory has been extended to nonlinear estimation [125, 126] and control [106, 107], for example via model predictive control [73, 66], control in eigenfunction coordinates [65], and switching control [103]. However, Koopman theory appears to follow the principle of conservation of difficulty, in that finding the right nonlinear measurements that enable a tractable linear representation may be as challenging as solving the original problem. In a sense, obtaining Koopman embeddings may be seen as an expensive offline computation that enables fast and efficient online prediction, estimation, and control. In addition, the Koopman operator is one of two main candidates for analyzing a dynamical system using operator-based approaches, the other being the Perron–Frobenius operator. The Perron–Frobenius operator evolves probability density functions along the flow of the dynamics, while the Koopman operator evolves observable functions of the state. These two operators are adjoint to each other in appropriately defined function spaces and it should therefore theoretically not matter which one is used to study the system's behavior [70].

Before introducing the mathematical formulation of Koopman operator theory, we first consider the flow map $\mathbf{F}_{\Delta t}$ obtained by integrating the PDE in (7.1) for a short-time Δt, given by

$$\mathbf{u}_{k+1} = \mathbf{F}_{\Delta t}(\mathbf{u}_k). \tag{7.20}$$

The Koopman operator $\mathcal{K}$ is defined so that

$$\mathcal{K}_t g = g \circ \mathbf{F}_t, \tag{7.21}$$

where ∘ is the composition operator. For a discrete-time system with time step Δt, this becomes

$$\mathcal{K}_{\Delta t} g(\mathbf{u}_k) = g(\mathbf{F}_{\Delta t}(\mathbf{u}_k)) = g(\mathbf{u}_{k+1}). \tag{7.22}$$

In other words, the Koopman operator defines an infinite-dimensional linear dynamical system that advances the observation of the state $g_k = g(\mathbf{u}_k)$ to the next time step:

$$g(\mathbf{u}_{k+1}) = \mathcal{K}_{\Delta t} g(\mathbf{u}_k). \tag{7.23}$$

Note that this is true for *any* observable function g and for any state $\mathbf{u}_k$.

Much of the challenge of modern Koopman theory is obtaining a finite-dimensional representation $\mathbf{K}$ of the infinite-dimensional operator $\mathcal{K}$. In practice, this amounts to discovering eigenfunctions of the Koopman operator, which are measurement functions that behave linearly when evolved forward in time. A discrete-time Koopman eigenfunction $\varphi(\mathbf{u})$ corresponding to eigenvalue λ satisfies

$$\varphi(\mathbf{u}_{k+1}) = \mathcal{K}_{\Delta t} \varphi(\mathbf{u}_k) = \lambda \varphi(\mathbf{u}_k). \tag{7.24}$$

In continuous-time, a Koopman eigenfunction $\varphi(\mathbf{u})$ satisfies

$$\frac{d}{dt} \varphi(\mathbf{u}) = \mathcal{K} \varphi(\mathbf{u}) = \lambda \varphi(\mathbf{u}). \tag{7.25}$$

Obtaining Koopman eigenfunctions from data or from analytic expressions is a central applied challenge in modern dynamical systems. Discovering these eigenfunctions enables globally linear representations of strongly nonlinear systems. Applying the chain rule to the time derivative of the Koopman eigenfunction $\varphi(\mathbf{u})$ yields

$$\frac{d}{dt} \varphi(\mathbf{u}) = \nabla \varphi(\mathbf{u}) \cdot \dot{\mathbf{u}} = \nabla \varphi(\mathbf{u}) \cdot \mathbf{f}(\mathbf{u}). \tag{7.26}$$

Combined with (7.25), this results in a PDE for the eigenfunction $\varphi(\mathbf{u})$:

$$\nabla \varphi(\mathbf{u}) \cdot \mathbf{f}(\mathbf{u}) = \lambda \varphi(\mathbf{u}). \tag{7.27}$$

With this nonlinear PDE, it is possible to approximate the eigenfunctions, either by solving for the Laurent series or with data via regression, both of which are explored below. This formulation assumes that the dynamics are both continuous and differentiable. The discrete-time dynamics in (7.20) are more general, although in many examples the continuous-time dynamics have a simpler representation than the discrete-time map for long times. Koopman analysis has recently been extended to the continuous PDE formulation, rather than just the high-dimensional discretized ODE context, for example, showing that the Cole–Hopf transform is a Koopman embedding for Burgers' equation [80].

There are many approaches to obtain finite-dimensional approximations to the Koopman operator. DMD is a representation based on linear observables [109], which has been extended to nonlinear observables in the *extended* DMD (eDMD) [144] and the variational approach of conformation dynamics [99, 100]. In all of these cases, it is important that the measurements are chosen to form a Koopman-invariant subspace [23]; otherwise, the projection of the Koopman operator onto this subspace will result in spurious eigenvalues and eigenfunctions.

In eDMD, an augmented state is constructed:

$$\mathbf{y} = \boldsymbol{\Theta}^T(\mathbf{u}) = \begin{bmatrix} \theta_1(\mathbf{u}) \\ \theta_2(\mathbf{u}) \\ \vdots \\ \theta_p(\mathbf{u}) \end{bmatrix}. \tag{7.28}$$

The projection $\boldsymbol{\Theta}$ may contain the original state $\mathbf{u}$ as well as nonlinear measurements, so often $p \gg n$. Next, two data matrices are constructed, as in DMD:

$$\mathbf{Y} = \begin{bmatrix} | & | & & | \\ \mathbf{y}_1 & \mathbf{y}_2 & \cdots & \mathbf{y}_m \\ | & | & & | \end{bmatrix}, \quad \mathbf{Y}' = \begin{bmatrix} | & | & & | \\ \mathbf{y}_2 & \mathbf{y}_3 & \cdots & \mathbf{y}_{m+1} \\ | & | & & | \end{bmatrix}. \tag{7.29a}$$

Finally, a best-fit linear operator $\mathbf{A_Y}$ is constructed that maps $\mathbf{Y}$ into $\mathbf{Y}'$:

$$\mathbf{A_Y} = \mathrm{argmin}_{\mathbf{A_Y}} \|\mathbf{Y}' - \mathbf{A_Y}\mathbf{Y}\|_2 = \mathbf{Y}'\mathbf{Y}^\dagger. \tag{7.30}$$

This regression may be written in terms of the data matrices $\boldsymbol{\Theta}(\mathbf{X})$ and $\boldsymbol{\Theta}(\mathbf{X}')$:

$$\mathbf{A_Y} = \mathrm{argmin}_{\mathbf{A_Y}} \|\boldsymbol{\Theta}^T(\mathbf{X}') - \mathbf{A_Y}\boldsymbol{\Theta}^T(\mathbf{X})\|_2 = \boldsymbol{\Theta}^T(\mathbf{X}')(\boldsymbol{\Theta}^T(\mathbf{X}))^\dagger. \tag{7.31}$$

The resulting nonlinear model for u_k is given by the proxy eDMD variable $\mathbf{y}_{k+1} = \mathbf{A_Y}\mathbf{y}_k$. Because the augmented vector $\mathbf{y}$ may be significantly larger than the state $\mathbf{u}$, kernel methods are often employed to compute this regression [145]. In principle, the enriched library $\boldsymbol{\Theta}$ provides a larger basis in which to approximate the Koopman operator. It has been shown recently that in the limit of infinite snapshots, the eDMD operator converges to the Koopman operator projected onto the subspace spanned by $\boldsymbol{\Theta}$ [144, 70, 74]. However, if $\boldsymbol{\Theta}$ does not span a Koopman-invariant subspace, then the projected operator may not have any resemblance to the original Koopman operator, as all of the eigenvalues and eigenvectors may be different. In fact, it was shown that the eDMD operator will have spurious eigenvalues and eigenvectors unless it is represented in terms of a Koopman-invariant subspace [23]. Therefore, it is essential to use validation and cross-validation techniques to ensure that eDMD models are not overfit, as discussed below. For example, it was shown that eDMD cannot contain the original state $\mathbf{u}$ as a measurement and represent a system that has multiple fixed points,

periodic orbits, or other attractors, because these systems cannot be topologically conjugate to a finite-dimensional linear system [23]. Recently, researchers have been leveraging the representational power of deep neural networks to identify Koopman eigenfunctions and approximate Koopman operators [130, 149, 92, 141, 101, 83]. In the next section, we will discuss an alternative approach to obtain a Koopman-invariant subspace based on time delay coordinates [22].

7.2.4 Time-delay embeddings for Koopman embeddings

Instead of advancing instantaneous linear or nonlinear measurements of the state of a system directly, as in DMD, it may be possible to obtain intrinsic measurement coordinates for Koopman based on time-delayed measurements of the system [127, 22, 5, 43, 68]. This perspective is data-driven, relying on the wealth of information from previous measurements to inform the future. Unlike a linear or weakly nonlinear system, where trajectories may get trapped at fixed points or on periodic orbits, chaotic dynamics are particularly well suited to this analysis: Trajectories evolve to densely fill an attractor, so more data provide more information. The use of delay coordinates may be especially important for systems with long-term memory effects, where the Koopman approach has recently been shown to provide a successful analysis tool [128]. Interestingly, a connection between the Koopman operator and the Takens embedding was explored as early as in 2004 [95], where a stochastic Koopman operator is defined and a statistical Takens theorem is proven. One version of time-delay embeddings, the HAVOK, has been used successfully to diagnose a diverse set of dynamical systems [22]. More broadly, there are a number of analysis tools that can be applied to the Hankel matrix for analysis of dynamics [68].

The time-delay measurement scheme is shown schematically in Figure 7.2, as illustrated on the Lorenz system for a single time-series measurement of the first variable, $x(t)$. If the conditions of the Takens embedding theorem are satisfied [131], it is possible to obtain a diffeomorphism between a delay-embedded attractor and the attractor in the original coordinates. We then obtain eigentime-delay coordinates from a time series of a single measurement $x(t)$ by taking the SVD of the Hankel matrix $\mathbf{H}$:

$$\mathbf{H} = \begin{bmatrix} x(t_1) & x(t_2) & \cdots & x(t_{m_c}) \\ x(t_2) & x(t_3) & \cdots & x(t_{m_c+1}) \\ \vdots & \vdots & \ddots & \vdots \\ x(t_{m_o}) & x(t_{m_o+1}) & \cdots & x(t_m) \end{bmatrix} = \mathbf{\Psi}_{TD} \mathbf{\Sigma} \mathbf{V}^*, \tag{7.32}$$

where m_c is the number of snapshots and m_o is the total number of delays. The columns of $\mathbf{\Psi}_{TD}$ and $\mathbf{V}$ from the SVD are arranged hierarchically by their ability to model the columns and rows of $\mathbf{H}$, respectively. Often, $\mathbf{H}$ may admit a low-rank approximation by the first r columns of $\mathbf{\Psi}_{TD}$ and $\mathbf{V}$. Note that the Hankel matrix in (7.32) is
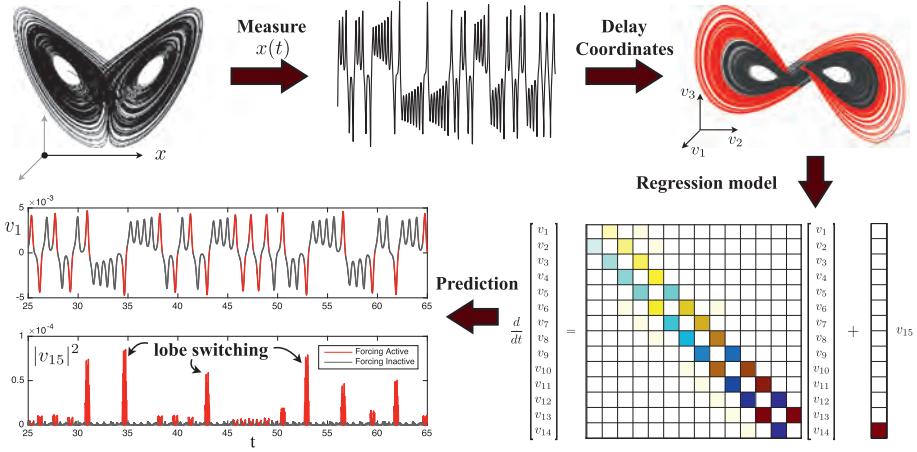
Figure 7.2: Schematic of the Hankel alternative view of Koopman (HAVOK) algorithm [22], as illustrated on the Lorenz 63 system. A time series $x(t)$ is stacked into a Hankel matrix **H**. The SVD of **H** yields a hierarchy of *eigen*time series that produce a delay-embedded attractor. A best-fit linear regression model is obtained on the delay coordinates **v**; the linear fit for the first $r - 1$ variables is excellent, but the last coordinate v_r is not well modeled as linear. Instead, v_r is an input that forces the first $r - 1$ variables. Rare forcing events correspond to lobe switching in the chaotic dynamics. *From Brunton and Kutz* [24]*, modified from* [22].

the basis of the eigensystem realization algorithm [64] in linear system identification and SSA [20] in climate time-series analysis.

The low-rank approximation to (7.32) provides a *data-driven* measurement system that is approximately invariant to the Koopman operator for states on the attractor. By definition, the dynamics map the attractor into itself, making it *invariant* to the flow. In other words, the columns of **U** form a Koopman-invariant subspace. We may rewrite (7.32) with the Koopman operator $\mathcal{K} \triangleq \mathcal{K}_{\Delta t}$:

$$
\mathbf{H} = \begin{bmatrix} x(t_1) & \mathcal{K}x(t_1) & \cdots & \mathcal{K}^{m_c-1}x(t_1) \\ \mathcal{K}x(t_1) & \mathcal{K}^2 x(t_1) & \cdots & \mathcal{K}^{m_c}x(t_1) \\ \vdots & \vdots & \ddots & \vdots \\ \mathcal{K}^{m_o-1}x(t_1) & \mathcal{K}^{m_o}x(t_1) & \cdots & \mathcal{K}^{m-1}x(t_1) \end{bmatrix}.
\tag{7.33}
$$

The columns of (7.32) are well approximated by the first r columns of $\mathbf{\Psi}_{\text{TD}}$. The first r columns of **V** provide a time series of the magnitude of each of the columns of $\mathbf{\Psi}_{\text{TD}}\mathbf{\Sigma}$ in the data. By plotting the first three columns of **V**, we obtain an embedded attractor for the Lorenz system (Figure 7.2).

The connection between eigentime-delay coordinates from (7.32) and the Koopman operator motivates a linear regression model on the variables in **V**. Even with an approximately Koopman-invariant measurement system, there remain challenges

to identifying a linear model for a chaotic system. A linear model, however detailed, cannot capture multiple fixed points or the unpredictable behavior characteristic of chaos with a positive Lyapunov exponent [23]. Instead of constructing a closed linear model for the first r variables in $\mathbf{V}$, we build a linear model on the first $r - 1$ variables and recast the last variable, v_r, as a forcing term:

$$\frac{d}{dt}\mathbf{v}(t) = \mathbf{A}\mathbf{v}(t) + \mathbf{B}v_r(t), \tag{7.34}$$

where $\mathbf{v} = \begin{bmatrix} v_1 & v_2 & \cdots & v_{r-1} \end{bmatrix}^T$ is a vector of the first $r - 1$ eigentime-delay coordinates. Other work has investigated the splitting of dynamics into deterministic linear and chaotic stochastic dynamics [93].

In all of the examples explored in [22], the linear model on the first $r - 1$ terms is accurate, while no linear model represents v_r. Instead, v_r is an input forcing to the linear dynamics in (7.34), which approximates the nonlinear dynamics. The statistics of $v_r(t)$ are non-Gaussian, with long tails corresponding to rare-event forcing that drives lobe switching in the Lorenz system; this is related to rare-event forcing distributions observed and modeled by others [86, 113, 87].

7.3 Data-driven model discovery

For many modern complex systems of interest, such as in materials science, neuroscience, epidemiology, climate science, and finance, there is a basic lack of physical laws and governing equations. Even when governing equations are available, for example in fluid turbulence, protein folding, and combustion, the equations are so complex that they are not readily amenable to analysis. With increasingly complex systems, and the emergence of powerful computing architectures and big data, the paradigm is shifting to data-driven methods to discover governing equations [19, 119, 25, 111].

7.3.1 SINDy: sparse identification of nonlinear dynamics

Discovering ROMs from data is a central challenge in modern computational physics. Typically, the form of a candidate model is either constrained via prior knowledge of the governing equations, as in Galerkin projection [98, 9, 34], or a handful of heuristic models are tested and parameters are optimized to fit data. Alternatively, best-fit linear models may be obtained using DMD. Simultaneously identifying the nonlinear structure and parameters of a model from data is considerably more challenging, as there are combinatorially many possible model structures.

The SINDy algorithm [25] bypasses the intractable combinatorial search through all possible model structures, leveraging the fact that many dynamical systems

$$\frac{d}{dt}\mathbf{a} = \mathbf{f}(\mathbf{a}) \tag{7.35}$$

have dynamics $\mathbf{f}$ with only a few active terms in the space of possible right-hand side functions; for example, the Lorenz equations (Figure 7.3) only have a few linear and quadratic interaction terms per equation. Here, $\mathbf{a} \in \mathbb{R}^r$ is a low-dimensional state, for example obtained via POD/SVD [25, 81], or constructed from physically realizable measurements, such as lift, drag, and the derivative of lift for aerodynamic systems [82].
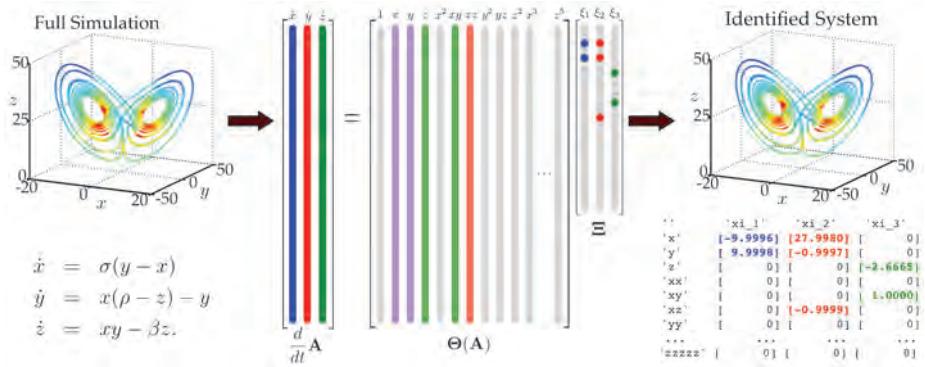
Figure 7.3: Schematic of the sparse identification of nonlinear dynamics (SINDy) algorithm [25], as illustrated on the Lorenz 63 system. *From Brunton and Kutz* [24]*, modified from* [25].

We then seek to approximate $\mathbf{f}$ by a generalized linear model in a set of candidate basis functions $\theta_k(\mathbf{a})$

$$\mathbf{f}(\mathbf{a}) \approx \sum_{k=1}^{p} \theta_k(\mathbf{a})\xi_k = \mathbf{\Theta}(\mathbf{a})\mathbf{\Xi}, \tag{7.36}$$

with the fewest nonzero terms in $\mathbf{\Xi}$. It is possible to solve for the relevant terms that are active in the dynamics using sparse regression [132, 155, 57, 63] that penalizes the number of terms in the dynamics and scales well to large problems.

First, time-series data are collected from (7.35) and formed into a data matrix:

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}(t_1) & \mathbf{a}(t_2) & \cdots \mathbf{a}(t_m) \end{bmatrix}^T. \tag{7.37}$$

A similar matrix of derivatives is formed:

$$\dot{\mathbf{A}} = \begin{bmatrix} \dot{\mathbf{a}}(t_1) & \dot{\mathbf{a}}(t_2) & \cdots \dot{\mathbf{a}}(t_m) \end{bmatrix}^T. \tag{7.38}$$

In practice, this may be computed directly from the data in **A** using a numerical differencing scheme, for instance. However, for noisy data, the total-variation regularized derivative tends to provide numerically robust derivatives [36]. Alternatively, it is possible to formulate the SINDy algorithm for discrete-time systems $\mathbf{a}_{k+1} = \mathbf{F}(\mathbf{a}_k)$, as in the DMD algorithm, and avoid derivatives entirely.

A library of candidate nonlinear functions $\boldsymbol{\Theta}(\mathbf{A})$ may be constructed from the data in **A**:

$$\boldsymbol{\Theta}(\mathbf{A}) = \begin{bmatrix} \mathbf{1} & \mathbf{A} & \mathbf{A}^2 & \cdots & \mathbf{A}^d & \cdots & \sin(\mathbf{A}) & \cdots \end{bmatrix}. \tag{7.39}$$

Here, the matrix $\mathbf{A}^d$ denotes a matrix with column vectors given by all possible time series of d-th-degree polynomials in the state **a**. In general, this library of candidate functions is only limited by one's imagination, but polynomials, trigonometric functions, and other well-known functions are a good starting point. The matrix $\boldsymbol{\Theta}$ tends to be ill-conditioned as the library elements, such as polynomials, are often not orthogonal. Indeed, they can often be nearly aligned over the time course where the library is evaluated. Despite the high condition number, the sparse selection advocated below is able to identify the correct dynamics provided the noise level is sufficiently small.

The dynamical system in (7.35) may now be represented in terms of the data matrices in (7.38) and (7.39) as

$$\dot{\mathbf{A}} = \boldsymbol{\Theta}(\mathbf{A})\boldsymbol{\Xi}. \tag{7.40}$$

Each column $\boldsymbol{\xi}_k$ in $\boldsymbol{\Xi}$ is a vector of coefficients determining the active terms in the k-th row in (7.35). A parsimonious model will provide an accurate model fit in (7.40) with as few terms as possible in $\boldsymbol{\Xi}$. Such a model may be identified using a convex ℓ_1-regularized sparse regression:

$$\boldsymbol{\xi}_k = \operatorname{argmin}_{\boldsymbol{\xi}_k'} \|\dot{\mathbf{A}}_k - \boldsymbol{\Theta}(\mathbf{A})\boldsymbol{\xi}_k'\|_2 + \lambda \|\boldsymbol{\xi}_k'\|_1. \tag{7.41}$$

Here, $\dot{\mathbf{A}}_k$ is the k-th column of $\dot{\mathbf{A}}$ and λ is a sparsity-promoting regularization weight, typically chosen by simple hyperparameter tuning. Sparse regression, such as the LASSO [132] or the sequential thresholded least-squares (STLS) algorithm used in SINDy [25], improves the numerical robustness of this identification for noisy overdetermined problems, in contrast to earlier methods [140] that used compressed sensing [45, 30, 32, 31, 33, 10, 137]. We advocate STLS to select active terms; there are recent guarantees on the convergence of this algorithm [152], and it has also been formalized in a general sparse regression framework called SR3 [154].

The sparse vectors $\boldsymbol{\xi}_k$ may be synthesized into a dynamical system:

$$\dot{a}_k = \boldsymbol{\Theta}(\mathbf{a})\boldsymbol{\xi}_k. \tag{7.42}$$

Note that x_k is the k-th element of **a** and $\boldsymbol{\Theta}(\mathbf{a})$ is a row vector of symbolic functions of **a**, as opposed to the data matrix $\boldsymbol{\Theta}(\mathbf{A})$.

The result of the SINDy regression is a parsimonious model that includes only the most important terms required to explain the observed behavior. The sparse regression procedure used to identify the most parsimonious nonlinear model is a convex procedure. The alternative approach, which involves regression onto every possible sparse nonlinear structure, constitutes an intractable brute-force search through the combinatorially many-candidate model forms. SINDy bypasses this combinatorial search with modern convex optimization and machine learning. It is interesting to note that for discrete-time dynamics, if $\Theta(\mathbf{A})$ consists only of linear terms, and if we remove the sparsity promoting term by setting $\lambda = 0$, then this algorithm reduces to DMD [117, 109, 138, 78]. If a least-squares regression is used, as in DMD, then even a small amount of measurement error or numerical round-off will lead to every term in the library being active in the dynamics, which is nonphysical. A major benefit of the SINDy architecture is the ability to identify parsimonious models that contain only the required nonlinear terms, resulting in interpretable models that avoid overfitting.

7.3.1.1 Extensions and applications

Because SINDy is formulated in terms of linear regression in a nonlinear library, it is highly extensible. The SINDy framework has been recently generalized by Loiseau and Brunton [81] to incorporate known physical constraints and symmetries in the equations by implementing a constrained sequentially thresholded least-squares optimization. In particular, energy-preserving constraints on the quadratic nonlinearities in the Navier–Stokes equations were imposed to identify fluid systems [81], where it is known that these constraints promote stability [86, 9, 34]. This work also showed that polynomial libraries are particularly useful for building models of fluid flows in terms of POD coefficients, yielding interpretable models that are related to classical Galerkin projection [25, 81]. Loiseau et al. [82] also demonstrated the ability of SINDy to identify dynamical systems models of high-dimensional systems, such as fluid flows, from a few physical sensor measurements, such as lift and drag measurements on the cylinder. SINDy has also been applied to identify models in nonlinear optics [123] and plasma physics [40]. For actuated systems, SINDy has been generalized to include inputs and control [26], and these models are highly effective for model predictive control [66]. It is also possible to extend the SINDy algorithm to identify dynamics with rational function nonlinearities [88], with integral terms [116], and based on highly corrupt and incomplete data [134]. SINDy was also recently extended to incorporate information criteria for objective model selection [89], and to identify models with hidden variables using delay coordinates [22]. Finally, the SINDy framework was generalized to include partial derivatives, enabling the identification of PDE models [111, 115], which will be explored below.

More generally, the use of sparsity-promoting methods in dynamics is quite recent [140, 114, 102, 84, 28, 105, 8, 7, 21, 90, 91]. Other techniques for dynami-

cal system discovery include methods to discover equations from time series [39], equation-free modeling [69], empirical dynamic modeling [124, 148], modeling emergent behavior [108], the nonlinear autoregressive model with exogenous inputs (NARMAX) [51, 153, 16, 121], and automated inference of dynamics [120, 41, 42]. Broadly speaking, these techniques may be classified as system identification, where methods from statistics and machine learning are used to identify dynamical systems from data. Nearly all methods of system identification involve some form of regression of data onto dynamics, and the main distinction between the various techniques is the degree to which this regression is constrained. For example, DMD generates best-fit linear models. Recent nonlinear regression techniques have produced nonlinear dynamic models that preserve physical constraints, such as conservation of energy. Yao and Bollt previously formulated the dynamical system identification problem as a similar linear inverse problem [147], although sparsity-promoting regression was not used, so the resulting models included all terms in $\mathbf{\Theta}$. In addition, SINDy is closely related to NARMAX [16], which identifies the structure of models from time-series data through an orthogonal least-squares procedure.

7.3.2 Model discovery for PDEs

A major extension of the SINDy modeling framework generalized the library to include partial derivatives, enabling the identification of PDEs [111, 115]. The resulting algorithm, called the PDE functional identification of nonlinear dynamics (PDE-FIND), shown in Figure 7.4, has been demonstrated to successfully identify several canonical PDEs from classical physics, purely from noisy data. These PDEs include Navier–Stokes, Kuramoto–Sivashinsky, Schrödinger, reaction diffusion, Burgers, Korteweg–de Vries (KdV), and the diffusion equation for Brownian motion [111].

PDE-FIND is similar to SINDy, in that it is based on sparse regression in a library constructed from measurement data. PDE-FIND is outlined below for PDEs in a single variable, although the theory is readily generalized to higher dimensional PDEs. The spatial time-series data are arranged into a single column vector $\mathbf{\Upsilon} \in \mathbb{C}^{mn}$, representing data collected over m time points and n spatial locations. Additional inputs, such as a known potential for the Schrödinger equation, or the magnitude of complex data, is arranged into a column vector $\mathbf{Q} \in \mathbb{C}^{mn}$. Next, a library $\mathbf{\Theta}(\mathbf{\Upsilon}, \mathbf{Q}) \in \mathbb{C}^{mn \times D}$ of D candidate linear and nonlinear terms and partial derivatives for the PDE is constructed. Derivatives are taken either using finite differences for clean data, or when noise is added, with polynomial interpolation. The candidate linear and nonlinear terms and partial derivatives are then combined into a matrix $\mathbf{\Theta}(\mathbf{\Upsilon}, \mathbf{Q})$ which takes the form

$$\mathbf{\Theta}(\mathbf{\Upsilon}, \mathbf{Q}) = \begin{bmatrix} \mathbf{1} & \mathbf{\Upsilon} & \mathbf{\Upsilon}^2 & \dots & \mathbf{Q} & \dots & \mathbf{\Upsilon}_x & \mathbf{\Upsilon}\mathbf{\Upsilon}_x & \dots \end{bmatrix}. \tag{7.43}$$

Each column of $\mathbf{\Theta}$ contains all of the values of a particular candidate function across all of the mn space-time grid points on which data are collected. The time derivative
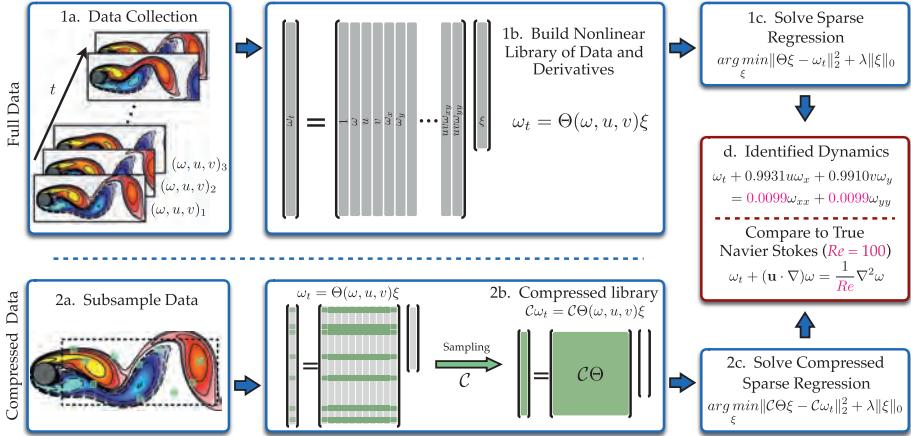
Figure 7.4: Schematic of PDE-FIND [111], as illustrated on the fluid flow past a circular cylinder. *From Rudy et al.* [111].

Υ_t is also computed and reshaped into a column vector. As an example, a column of $\Theta(\Upsilon, \mathbf{Q})$ may be qu_x^2.

The PDE evolution can be expressed in this library as follows:

$$\Upsilon_t = \Theta(\Upsilon, \mathbf{Q})\xi. \tag{7.44}$$

Each entry in ξ is a coefficient corresponding to a term in the PDE, and for canonical PDEs, the vector ξ is *sparse*, meaning that only a few terms are active.

If the library Θ has a sufficiently rich column space that the dynamics are in its span, then the PDE should be well represented by (7.44) with a sparse vector of coefficients ξ. To identify the few active terms in the dynamics, a sparsity-promoting regression is employed, as in SINDy. Importantly, the regression problem in (7.44) may be poorly conditioned. Errors in computing the derivatives will be magnified by numerical errors when inverting Θ. Thus a least-squares regression radically changes the qualitative nature of the inferred dynamics.

In general, we seek the sparsest vector ξ that satisfies (7.44) with a small residual. Instead of an intractable combinatorial search through all possible sparse vector structures, a common technique is to relax the problem to a convex ℓ_1-regularized least-squares [132]; however, this tends to perform poorly with highly correlated data. Instead, we use ridge regression with hard thresholding, which we call sequential threshold ridge regression. For a given tolerance and threshold λ, this gives a sparse approximation to ξ.

We iteratively refine the tolerance of Algorithm 1 to find the best predictor based on the selection criteria,

$$\hat{\xi} = \operatorname{argmin}_{\xi} \|\Theta(\Upsilon, \mathbf{Q})\xi - \Upsilon_t\|_2^2 + \epsilon \kappa(\Theta(\Upsilon, \mathbf{Q}))\|\xi\|_0, \tag{7.45}$$

where $\kappa(\boldsymbol{\Theta})$ is the condition number of the matrix $\boldsymbol{\Theta}$, providing stronger regularization for ill-posed problems. Penalizing $\|\boldsymbol{\xi}\|_0$ discourages overfitting by selecting from the optimal position in a Pareto front. While in general this problem is *NP*-hard we are restricting it to solutions generated via the STRidge algorithm, which promotes hard thresholding. Such hard thresholding has been recently shown to be a proxy for the ℓ_0-norm [154].

As in the SINDy algorithm, it is important to provide sufficiently rich training data to disambiguate between several different models. For example, if only a single traveling wave from the KdV equation is analyzed, the method incorrectly identifies the standard linear advection equation, as this is the simplest equation that describes a single traveling wave. However, if two traveling waves of different amplitudes are analyzed, the KdV equation is correctly identified, as it describes the different amplitude-dependent wave speeds [111].

The PDE-FIND algorithm can also be used to identify PDEs based on Lagrangian measurements that follow the path of individual particles. For example, it is possible to identify the diffusion equation describing Brownian motion of a particle based on a single long time-series measurement of the particle position. In this example, the time series is broken up into several short sequences, and the evolution of the distribution of these positions is used to identify the diffusion equation [111].

7.4 Data-driven ROMs

The methods detailed in the previous sections can be integrated with traditional model reduction architectures. In what follows, we highlight how such methods can be used in a data-driven way to construct ROM models in a nonintrusive, efficient manner.

7.4.1 Application of DMD and Koopman to ROM models

DMD provides an alternative approach to computing the projection of the nonlinearity onto the rank-r POD subspace in (7.3). Specifically, instead of using POD modes and gappy sampling for approximation of the nonlinear, low-rank contribution to the dynamics, DMD is used to directly compute a time evolution of the nonlinearity $\boldsymbol{\Psi}^T \mathbf{N}(\boldsymbol{\Psi}\mathbf{a})$ from snapshot data. Like the DEIM interpolation procedure [37], the DMD algorithm will proceed by constructing a snapshot matrix of the nonlinearity:

$$\mathbf{X}_{\mathrm{NL}} = \begin{bmatrix} | & | & & | \\ \mathbf{N}_1 & \mathbf{N}_2 & \cdots & \mathbf{N}_m \\ | & | & & | \end{bmatrix}, \tag{7.46}$$

where the columns $\mathbf{N}_k = \mathbf{N}(\mathbf{u}(t_k), \mathbf{x}, t_k) \in \mathbb{C}^n$ are evaluations of the nonlinearity at time t_k.

Following (7.16), a DMD of the matrix $\mathbf{X}_{\mathrm{NL}}$ gives a low-rank approximation of the form

$$\mathbf{N}(\mathbf{u}(t), \mathbf{x}, t) = \mathbf{\Phi}_{\mathrm{NL}} \exp(\mathbf{\Omega}_{\mathrm{NL}} t) \mathbf{b}_{\mathrm{NL}} . \tag{7.47}$$

This low-rank approximation is achieved directly with further recourse to gappy interpolation for projecting back the DMD modes. The approximation can be used to modify (7.3) so as to achieve the following low-rank model:

$$\frac{d\mathbf{a}}{dt} = \mathbf{\Psi}^T \mathbf{L} \mathbf{\Psi} \mathbf{a} + \mathbf{\Psi}^T \mathbf{\Phi}_{\mathrm{NL}} \exp(\mathbf{\Omega}_{\mathrm{NL}} t) \mathbf{b}_{\mathrm{NL}}. \tag{7.48}$$

This integration of POD and DMD methods has been shown to provide performance increases in comparison to POD alone [2]. Moreover, the technique can be integrated with randomized linear algebra decomposition methods to achieve further enhancements in computational speed and scalability. Alla and Kutz further show that the POD-DMD integration competes well with POD with DEIM in terms of accuracy, while significantly outperforming it in terms of computation time. The DMD algorithm itself is faster than POD with DEIM and POD-DMD, but suffers from poor accuracy. One can also envision using a DMD-DMD reduction whereby a projection-based reduction with DMD-Galerkin is performed along with a hyperreduction with DMD. DMD-based ROM models have also recently been successfully demonstrated in a number of technical applications [1, 56]. A more detailed analysis of interpolation methods can be found in Chapter 7 of Volume 1 of *Model order reduction* [12].

7.4.2 Application of SINDy for ROMs

The SINDy algorithm can also be used to construct ROM architectures (7.2) from data alone, i. e., no governing equations are known a priori. As an example, the flow past a cylinder (Figure 7.5) provides a model with a rich history in fluid mechanics and dynamical systems [98]. It has long been theorized that turbulence is the result of a series of Hopf bifurcations that occur as the flow velocity increases [112], giving rise to more rich and intricate structures in the fluid. After 15 years, the first Hopf bifurcation was discovered in a fluid system, in the transition from a steady laminar wake to laminar periodic vortex shedding at Reynolds number 47 [62, 151]. This discovery led to a long-standing debate about how a Hopf bifurcation, with cubic nonlinearity, can be exhibited in a Navier–Stokes fluid with quadratic nonlinearities. After 15 more years, this was resolved using a separation of time scales and a mean-field model by Noack et al. [98]. It was shown that coupling between oscillatory modes and the base flow gives rise to a slow manifold, resulting in algebraic terms that approximate cubic nonlinearities on slow time scales.
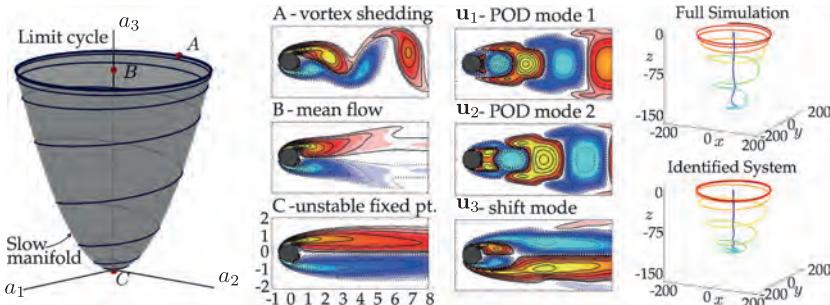
Figure 7.5: The vortex shedding past a cylinder is a prototypical example in fluid dynamics, with relevance to many applications, including drag reduction behind vehicles. Vortex shedding is the result of a Hopf bifurcation. However, because the Navier–Stokes equations have quadratic nonlinearity, it is necessary to employ a mean-field model with a separation of time scales, where a fast mean-field deformation is slave to the slow vortex shedding dynamics. The parabolic slow manifold is shown (left), with the unstable fixed point (C), mean flow (B), and vortex shedding (A). A POD basis and shift mode are used to reduce the dimension of the problem (middle right). The identified dynamics closely match the true trajectory in POD coordinates, and they capture the quadratic nonlinearity and time scales associated with the mean-field model. *From Brunton, Proctor and Kutz* [25].

This example provides a compelling test case for the proposed ROM-SINDy algorithm, since the underlying form of the dynamics took nearly three decades for experts in the community to uncover. Because the state dimension is large, it is advantageous to reduce the dimension of the system. POD provides a low-rank basis resulting in a hierarchy of orthonormal modes that, when truncated, capture the most energy of the original system for the given rank truncation. The first two most energetic POD modes capture a significant portion of the energy, and steady-state vortex shedding is a limit cycle in these coordinates. An additional mode, called the shift mode, is included to capture the transient dynamics connecting the unstable steady state with the mean of the limit cycle [98].

In the dominant POD coordinate system (rank $r = 3$), the mean-field model $\dot{\mathbf{a}} = \mathbf{f}(\mathbf{a})$ for the cylinder dynamics is discovered by SINDy to be [25]:

$$\dot{a}_1 = \mu a_1 - \omega a_2 + A a_1 a_3, \tag{7.49a}$$

$$\dot{a}_2 = \omega a_1 + \mu a_2 + A a_2 a_3, \tag{7.49b}$$

$$\dot{a}_3 = -\lambda(a_3 - a_1^2 - a_2^2). \tag{7.49c}$$

Note that the governing equations for $\mathbf{a}(t)$ in (7.49) are closely related to the slow-manifold formulation of Noack et al. [98] formulated using the standard Galerkin-POD projection. Specifically, it discovers the correct model $\dot{\mathbf{a}} = \mathbf{f}(\mathbf{a})$ with quadratic nonlinearities and reproduces a parabolic slow manifold. The a_3 variable corresponds to the shift-mode of Noack et al. [98], and if λ is large, so that the a_3-dynamics are fast, then the mean flow rapidly corrects to be on the slow manifold $a_3 = a_1^2 + a_2^2$ given

by the amplitude of vortex shedding. When substituting this algebraic relationship into equations (7.49a) and (7.49b), we recover the Hopf normal form on the slow manifold. Note that derivative measurements are not available, but are computed from the state variables. When the training data do not include trajectories that originate from the slow manifold, the algorithm incorrectly identifies cubic nonlinearities, and fails to identify the slow manifold. This model was subsequently improved by Loiseau and Brunton [81] to incorporate energy-conserving constraints and to include higher-order terms to model the effect of truncated POD modes.

7.4.3 Application of time-delay embeddings for ROMs

Time-delay embedding for building ROMs can be used in a completely data-driven architecture where the governing equations are unknown, or for building a Koopman operator for a known governing evolution equation [22]. Indeed, one can use time-delay embedding with the SINDy architecture when short time-delay embeddings are used, or for producing a direct Koopman approximation when long time-delay embeddings are used. Champion et al. [35] highlight the various architectures possible. The short-time and long-time embedding possibilities are detailed here.

7.4.3.1 Short time-delay embedding

For a short time-delay embedding, the time-shifted data can provide a more accurate assessment of the true rank of the underlying system. Such time-delay embedding was used by Tu et al. [138] in order to ensure that the data were not rank-deficient. Indeed, without time-shifting the data, the DMD approximation does not capture the correct complex eigenvalue pairs associated with the periodic (Fourier) time dynamics.

Figure 7.6 shows the effects of the time-delay embedding as illustrated on the simple Van der Pol oscillator. In the top left panel of this figure, the singular values of **H** given by (7.32) for a short time-delay embedding is shown. Specifically, the data were delayed by five time steps. For this delay, the rank of the matrix **H** is dominated by two modes. The time dynamics of the first three modes are shown in the middle panel of the figure, illustrating the strongly nonlinear Van der Pol oscillations. A reduced model can then be constructed from the first two modes so that $\mathbf{\Psi}_{TD}$ spans a rank-2 subspace. Importantly, the dominant nonlinear time-series data can then be used with the SINDy architecture to discover the governing equations and build a dynamical ROM model.
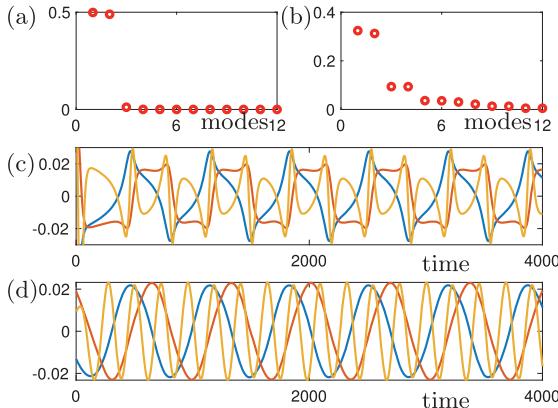
Figure 7.6: Time-delay embedding of the Van der Pol oscillator with time steps of 0.01. (a) With a short time-delay embedding of five time steps, the SVD produces a dominant low-rank (two-mode) truncation whose time-dynamic modes are illustrated in (c). (b) With a long time-delay embedding of several hundred time steps, the SVD produces a low-rank truncation of approximately a dozen modes whose time dynamic modes are illustrated in (d). Note that the short time-delay modes are strongly nonlinear oscillatory modes while the long time delay produces nearly perfect sinusoidal modes. Details can be found in Champion, Brunton, and Kutz [35].

7.4.3.2 Long time-delay embedding

For long time-delay embeddings, the nonlinear dynamics can be made to be approximately linear, thus providing an approximation to the Koopman operator and a linear ROM. The long time-delay embedding is especially useful in a data-driven architecture where the governing equations are unknown. Moreover, the time-delay embedding can significantly improve upon the DMD algorithm for producing an approximate dynamical system for forecasting.

Figure 7.6 shows the effects of the time-delay embedding as illustrated on the simple Van der Pol oscillator. In the top right panel of this figure, the singular values of **H** for a long-time delay embedding are shown. Specifically, the data were delayed by several hundred time steps which spanned more than a period of the nonlinear oscillations. Unlike the short time-delay embedding, the rank increases from two to about a dozen. The time dynamics of the first three of these dozen modes (i. e., the first three columns of the **V** matrix of (7.32)) are shown in the bottom panel. Note that the time modes with the long delay are now approximately sinusoidal, thus being ideal for a DMD/Koopman approximation. In this case, the SINDy architecture is unnecessary.

7.5 Conclusion and outlook

ROMs continue to play a critically enabling role in emulation and simulation strategies. Indeed, ROMs are making many intractable computations tractable by providing a surrogate model that can be computed at a fraction of the cost and with improved memory constraints. For emerging models in multiscale dynamical systems, such as in biology, atmospheric dynamics, and molecular dynamics simulations, ROMs provide a scalable mathematical framework, where it is possible to obtain accurate statistical estimates of the properties of the high-fidelity model from low-fidelity models.

Data-driven approaches to ROMs are also playing an increasingly important role in developing scalable and nonintrusive emulators. Thus the governing equations, which may be unknown or only partially known, can be approximated by a suite of emerging mathematical methods. Table 7.1 highlights the various methods that are available for producing data-driven ROMs. They are compared to the standard Galerkin-POD architecture. Importantly, for each ROM architecture, two things must be prescribed in the underlying separation of variable strategy (7.2): (i) the subspace on which the ROM is to be constructed, and (ii) the manner of extracting the dynamical evolution in this subspace. Of course, such reductions do not guarantee the construction of a stable ROM model, as recently highlighted by Carlberg et al. [34]. Thus for each ROM model strategy, care must be taken in order to produce a stable, low-rank emulator. Indeed, both POD-Galerkin and POD-DMD algorithms, for instance, must be modified in order to promote a stable time-stepping ROM.

If the governing evolution equations (7.1) are known, then a Galerkin-POD (or Petrov–Galerkin-POD) provides a simple projective method for producing a ROM. One can also use the DMD algorithm in this architecture (POD-DMD) for more rapid evaluation of the nonlinear terms. For unknown governing equations where the full state

Table 7.1: Model reduction algorithms and their subspaces. Included is one example reference highlighting the method.

Data-driven ROM algorithms	
ROM model $\mathbf{u}(\mathbf{x}, t) = \boldsymbol{\Psi}(\mathbf{x})\mathbf{a}(t)$	
Galerkin-POD [14]	$\dot{\mathbf{a}} = \boldsymbol{\Psi}^T \mathbf{L}\boldsymbol{\Psi}\mathbf{a} + \boldsymbol{\Psi}^T \mathbf{N}(\boldsymbol{\Psi}\mathbf{a})$
DMD [78]	$\mathbf{u} = \boldsymbol{\Phi}\exp(\boldsymbol{\Omega}t)\mathbf{b}$
POD-DMD [2]	$\dot{\mathbf{a}} = \boldsymbol{\Psi}^T \mathbf{L}\boldsymbol{\Psi}\mathbf{a} + \boldsymbol{\Psi}^T \boldsymbol{\Phi}_{\mathrm{NL}}\exp(\boldsymbol{\Omega}_{\mathrm{NL}}t)\mathbf{b}_{\mathrm{NL}}$
POD-SINDy [25]	$\dot{\mathbf{a}} = \mathbf{f}(\mathbf{a})$ dynamics on subspace $\boldsymbol{\Psi}$
HAVOK-SINDy [35]	$\dot{\mathbf{a}} = \mathbf{f}(\mathbf{a})$ dynamics on subspace $\boldsymbol{\Psi}_{\mathrm{TD}}$ (short delay)
HAVOK-Koopman [22]	$\dot{\mathbf{a}} = \mathbf{K}\mathbf{a}$ dynamics on subspace $\boldsymbol{\Psi}_{\mathrm{TD}}$ (long delay)
Basis elements (rank r)	
POD modes	$\mathbf{X} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_m] = \boldsymbol{\Psi}\boldsymbol{\Sigma}\mathbf{V}^*$
DMD modes	$\mathbf{X} = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_m] = \boldsymbol{\Phi}\exp(\boldsymbol{\Omega}t)\mathbf{b}$
nonlinear DMD modes	$\mathbf{N} = [\mathbf{N}_1 \ \mathbf{N}_2 \ \dots \ \mathbf{N}_m] = \boldsymbol{\Phi}_{\mathrm{NL}}\exp(\boldsymbol{\Omega}_{\mathrm{NL}}t)\mathbf{b}_{\mathrm{NL}}$
Time-delay Koopman modes	$\mathbf{H} = \boldsymbol{\Psi}_{\mathrm{TD}}\boldsymbol{\Sigma}\mathbf{V}^*$

space is sampled, DMD can be used to produce a low-rank, best-fit linear model for the dynamics. An alternative to DMD is the POD-SINDy algorithm, which discovers a low-rank, nonlinear dynamical system approximating the dynamics of the system. Time-delay embeddings allow for some flexibility in building a ROM depending upon the scenario. Time-delay embeddings also allow one to handle latent variables when the full state measurements are unknown or unavailable. For a long time-delay embedding with known or unknown governing equations, one can augment the DMD algorithm by producing a time-delay coordinate system which helps make the dynamics linearly dominant (HAVOK-Koopman). A short time delay can be used to determine the rank of the underlying dynamics and potentially build a SINDy model (HAVOK-SINDy). Alternatively, a long time-delay embedding can discover the intrinsic rank and linearize the dynamics in the time-delay coordinates. For more details on DMD, its variants, and its broad applications, please see [78]. For a broader overview of data-driven methods and machine learning applied to dynamics, please see [24].

The diversity of strategies is important in modern complex systems simulations where often the equations are only partially known, but where rich measurement data may be available. Thus data-driven strategies can bridge the gap between measurement space and model space. Table 7.1 gives a summary of the various current techniques. It is envisioned that refinement and innovations using the various strategies will greatly aid in modeling the challenge problems in many fields where high-dimensional, multiscale physics are prevalent. Figure 7.7 gives a summary of the decision space necessary when considering an appropriate ROM. One can either em-
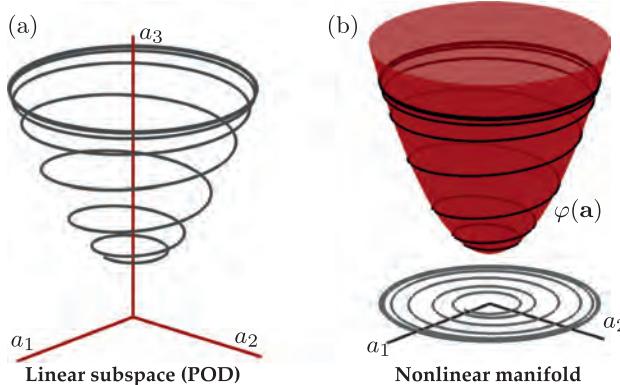
Figure 7.7: Low-order modeling of fluid flows begins with an appropriate coordinate system that captures the few dominant flow mechanisms that are dynamically relevant. It is most common to embed high-dimensional fluid data in a linear subspace, for example using POD (a). However, for the flow past a cylinder, it is clear that the data live on a low-dimensional manifold in the embedding space (b). Both approaches have been explored extensively, for example by Noack et al. [98] and Loiseau et al. [82]. After an appropriate coordinate system is obtained, there are several choices for model construction.

bed in a linear space or in a nonlinear space (manifold), and then determine the appropriate nonlinear dynamics. This can be done in a variety of ways depending on wether the underlying governing equations are known, or if only measurement data are available.

Bibliography

[1] I. Abraham, G. De La Torre, and T. D. Murphey, Model-based control using Koopman operators. arXiv preprint arXiv:1709.01568, 2017.

[2] A. Alla and J. N. Kutz, Nonlinear model order reduction via dynamic mode decomposition, *SIAM Journal on Scientific Computing*, **39** (5) (2017), B778–B796.

[3] A. Alla and J. N. Kutz, Randomized model order reduction, *Advances in Computational Mathematics*, **45** (3) (2019), 1251–1271.

[4] D. Amsallem and C. Farhat, Stabilization of projection-based reduced-order models, *International Journal for Numerical Methods in Engineering*, **91** (4) (2012), 358–377.

[5] H. Arbabi and I. Mezić, Ergodic theory, dynamic mode decomposition and computation of spectral properties of the Koopman operator, *SIAM Journal on Applied Dynamical Systems*, **16** (4) (2017), 2096–2126.

[6] T. Askham and J. N. Kutz, Variable projection methods for an optimized dynamic mode decomposition, *SIAM Journal on Applied Dynamical Systems*, **17** (1) (2018), 380–416.

[7] Z. Bai, S. L. Brunton, B. W. Brunton, J. N. Kutz, E. Kaiser, A. Spohn, and B. R. Noack, Data-driven methods in fluid dynamics: Sparse classification from experimental data, in *Invited chapter for Whither Turbulence and Big Data in the 21st Century*, 2015.

[8] Z. Bai, Z. Berger T. Wimalajeewa, G. Wang, M. Glauser, and P. K. Varshney, Low-dimensional approach for reconstruction of airfoil data via compressive sensing, *AIAA Journal*, **53** (4) (2014), 920–933.

[9] M. J. Balajewicz, E. H. Dowell, and B. R. Noack, Low-dimensional modelling of high-reynolds-number shear flows incorporating constraints from the navier–stokes equation, *Journal of Fluid Mechanics*, **729** (2013), 285–308.

[10] R. G. Baraniuk, Compressive sensing, *IEEE Signal Processing Magazine*, **24** (4) (2007), 118–120.

[11] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations, *Comptes Rendus. Mathématique*, **339** (9) (2004), 667–672.

[12] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Silveira (eds.), *Model Order Reduction. Volume 1: System- and Data-Driven Methods and Algorithms*, De Gruyter, Berlin, 2020.

[13] P. Benner, S. Grivet-Talocia, A. Quarteroni, G. Rozza, W. H. A. Schilders, and L. M. Silveira (eds.), *Model Order Reduction. Volume 2: Snapshot-Based Methods and Algorithms*, De Gruyter, Berlin, 2020.

[14] P. Benner, S. Gugercin, and K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, *SIAM Review*, **57** (4) (2015), 483–531.

[15] P. Benner, C. Himpe, and T. Mitchell, On reduced input-output dynamic mode decomposition, *Advances in Computational Mathematics*, **44** (6) (1751–1768), 2018.

[16] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*, John Wiley & Sons, 2013.

[17] G. D. Birkhoff and B. O. Koopman, Recent contributions to the ergodic theory, *Proceedings of the National Academy of Sciences*, **18** (3) (1932), 279–282.

[18] D. A. Bistrian and I. M. Navon, Randomized dynamic mode decomposition for nonintrusive reduced order modelling, *International Journal for Numerical Methods in Engineering*, **112** (1) (2017), 3–25.

[19] J. Bongard and H. Lipson, Automated reverse engineering of nonlinear dynamical systems, *Proceedings of the National Academy of Sciences*, **104** (24) (2007), 9943–9948.

[20] D. S. Broomhead and R. Jones, Time-series analysis, *Proceedings of the Royal Society of London A*, **423** (1864), 103–121.

[21] B. W. Brunton, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Sparse sensor placement optimization for classification, *SIAM Journal on Applied Mathematics*, **76** (5) (2016), 2099–2122.

[22] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, Chaos as an intermittently forced linear system, *Nature Communications*, **8** (19) (2017), 1–9.

[23] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control, *PLoS ONE*, **11** (2) (2016), e0150171.

[24] S. L. Brunton and J. N. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, Cambridge University Press, 2018.

[25] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, *Proceedings of the National Academy of Sciences*, **113** (15) (2016), 3932–3937.

[26] S. L. Brunton, J. L. Proctor, and J. N. Kutz, Sparse identification of nonlinear dynamics with control (SINDYc), *IFAC-PapersOnLine*, **49** (18) (2016), 710–715.

[27] S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz, Compressed sensing and dynamic mode decomposition, *Journal of Computational Dynamics*, **2** (2) (2015), 165–191.

[28] S. L. Brunton, J. H. Tu, I. Bright, and J. N. Kutz, Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems, *SIAM Journal on Applied Dynamical Systems*, **13** (4) (2014), 1716–1732.

[29] M. Budišić, R. Mohr, and I. Mezić, Applied Koopmanism a), *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **22** (4): 047510, 2012.

[30] E. J. Candès, Compressive sensing, in *Proceedings of the International Congress of Mathematics*, 2006.

[31] E. J. Candès, J. Romberg, and T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, *IEEE Transactions on Information Theory*, **52** (2) (2006), 489–509.

[32] E. J. Candès, J. Romberg, and T. Tao, Stable signal recovery from incomplete and inaccurate measurements. *Communications in Pure and Applied Mathematics*, **8**(1207–1223), 59.

[33] E. J. Candès and T. Tao, Near optimal signal recovery from random projections: Universal encoding strategies?, *IEEE Transactions on Information Theory*, **52** (12) (2006), 5406–5425.

[34] K. Carlberg, M. Barone, and H. Antil, Galerkin v. least-squares petrov–galerkin projection in nonlinear model reduction, *Journal of Computational Physics*, **330** (2017), 693–734.

[35] K. P. Champion, S. L. Brunton, and J. N. Kutz, Discovery of nonlinear multiscale systems: Sampling strategies and embeddings, *SIAM Journal on Applied Dynamical Systems*, **18** (1) (2019), 312–333.

[36] R. Chartrand, Numerical differentiation of noisy, nonsmooth data, *ISRN Applied Mathematics*, **2011**, 2011.

[37] S. Chaturantabut and D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, *SIAM Journal on Scientific Computing*, **32** (5) (2010), 2737–2764.

[38] K. K. Chen, J. H. Tu, and C. W. Rowley, Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses, *Journal of Nonlinear Science*, **22** (6) (2012), 887–915.

[39] J. P. Crutchfield and B. S. McNamara, Equations of motion from a data series, *Complex Systems*, **1** (1987), 417–452.

[40] M. Dam, M. Brøns, J. J. Rasmussen, V. Naulin, and J. S. Hesthaven, Sparse identification of a predator-prey system from simulation data of a convection model, *Physics of Plasmas*, **24** (2) (2017), 022310.

[41] B. C. Daniels and I. Nemenman, Automated adaptive inference of phenomenological dynamical models, *Nature Communications*, **6**, 2015.

[42] B. C. Daniels and I. Nemenman, Efficient inference of parsimonious phenomenological models of cellular dynamics using s-systems and alternating regression, *PLoS ONE*, **10** (3) (2015), e0119821.

[43] S. Das and D. Giannakis, Delay-coordinate maps and the spectra of Koopman operators, *Journal of Statistical Physics*, (2019), 1–39.

[44] S. T. M. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley, Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition, *Experiments in Fluids*, **57** (3) (2016), 1–19.

[45] D. L. Donoho, Compressed sensing, *IEEE Transactions on Information Theory*, **52** (4) (2006), 1289–1306.

[46] Z. Drmac and S. Gugercin, A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions, *SIAM Journal on Scientific Computing*, **38** (2) (2016), A631–A648.

[47] C. Eckart and G. Young, The approximation of one matrix by another of lower rank, *Psychometrika*, **1** (3) (1936), 211–218.

[48] N. B. Erichson and C. Donovan, Randomized low-rank dynamic mode decomposition for motion detection, *Computer Vision and Image Understanding*, **146** (2016), 40–50.

[49] N. B. Erichson, L. Mathelin, J. N. Kutz, and S. L. Brunton, Randomized dynamic mode decomposition, *SIAM Journal on Applied Dynamical Systems*, **18** (4) (2019), 1867–1891.

[50] R. Everson and L. Sirovich, Karhunen–Loeve procedure for gappy data, *Journal of the Optical Society of America. A, Online*, **12** (8) (1995), 1657–1664.

[51] B. Glaz, L. Liu, and P. P. Friedmann, Reduced-order nonlinear unsteady aerodynamic modeling using a surrogate-based recurrence framework, *AIAA Journal*, **48** (10) (2010), 2418–2429.

[52] G. H. Golub and R. J. LeVeque, Extensions and uses of the variable projection algorithm for solving nonlinear least squares problems, in *Proceedings of the 1979 Army Numerical Analsysis and Computers Conference*, 1979.

[53] G. H. Golub and V. Pereyra, The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate, *SIAM Journal on Numerical Analysis*, **10** (2) (1973), 413–432.

[54] F. Gueniat, L. Mathelin, and L. Pastur, A dynamic mode decomposition approach for large and arbitrarily sampled systems, *Physics of Fluids*, **27** (2) (2015), 025113.

[55] N. Halko, P.-G. Martinsson, and J. A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, *SIAM Review*, **53** (2) (2011), 217–288.

[56] S. Hanke, S. Peitz, O. Wallscheid, S. Klus, J. Böcker, and M. Dellnitz, Koopman operator based finite-set model predictive control for electrical drives. arXiv preprint arXiv:1804.00854, 2018.

[57] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, vol. 2, Springer, 2009.

[58]   M. S. Hemati, C. W. Rowley, E. A. Deem, and L. N. Cattafesta, De-biasing the dynamic mode decomposition for applied Koopman spectral analysis, *Theoretical and Computational Fluid Dynamics*, **31** (4) (2017), 349–368.

[59]   M. Hinze and S. Volkwein, Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control, in *Dimension Reduction of Large-Scale Systems*, pp. 261–306, Springer, 2005.

[60]   P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, 2nd paperback edition, Cambridge University Press, Cambridge, 2012.

[61]   P. Holmes and J. Guckenheimer, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, Applied Mathematical Sciences, vol. 42, Springer-Verlag, Berlin, Heidelberg, 1983.

[62]   C. P. Jackson, A finite-element study of the onset of vortex shedding in flow past variously shaped bodies, *Journal of Fluid Mechanics*, **182** (1987), 23–45.

[63]   G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, Springer, 2013.

[64]   J. N. Juang and R. S. Pappa, An eigensystem realization algorithm for modal parameter identification and model reduction, *Journal of Guidance, Control, and Dynamics*, **8** (5) (1985), 620–627.

[65]   E. Kaiser, J. N. Kutz, and S. L. Brunton, Data-driven discovery of Koopman eigenfunctions for control. arXiv preprint arXiv:1707.01146, 2017.

[66]   E. Kaiser, J. N. Kutz, and S. L. Brunton, Sparse identification of nonlinear dynamics for model predictive control in the low-data limit, *Proceedings of the Royal Society A*, **474** (2219) (2018), 20180335.

[67]   I. Kalashnikova, M. F. Barone, S. Arunajatesan, and B. G. van Bloemen Waanders, Construction of energy-stable projection-based reduced order models, *Applied Mathematics and Computation*, **249** (2014), 569–596.

[68]   M. Kamb, E. Kaiser, S. L. Brunton, and J. N. Kutz, Time-delay observables for Koopman: Theory and applications, *SIAM Journal on Applied Dynamical Systems*, **19** (2) (2020), 886–917.

[69]   I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, and C. Theodoropoulos, Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis, *Communications in Mathematical Sciences*, **1** (4) (2003), 715–762.

[70]   S. Klus, P. Koltai, and C. Schütte, On the numerical approximation of the Perron-Frobenius and Koopman operator. arXiv preprint arXiv:1512.05997, 2015.

[71]   B. O. Koopman, Hamiltonian systems and transformation in Hilbert space, *Proceedings of the National Academy of Sciences*, **17** (5) (1931), 315–318.

[72]   B. O. Koopman and J. v. Neumann, Dynamical systems of continuous spectra, *Proceedings of the National Academy of Sciences of the United States of America*, **18** (3) (1932), 255.

[73]   M. Korda and I. Mezić, Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control, *Automatica*, **93** (2018), 149–160.

[74]   M. Korda and I. Mezić, On convergence of extended dynamic mode decomposition to the Koopman operator, *Journal of Nonlinear Science*, **28** (2) (2018), 687–710.

[75]   K. Kunisch and S. Volkwein, Galerkin proper orthogonal decomposition methods for parabolic problems, *Numerische Mathematik*, **90** (1) (2001), 117–148.

[76]   K. Kunisch and S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, *SIAM Journal on Numerical Analysis*, **40** (2) (2002), 492–515.

[77]   J. N. Kutz, *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data*, Oxford University Press, 2013.

[78]   J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor, *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, SIAM, 2016.

[79]   J. N. Kutz, X. Fu, and S. L. Brunton, Multiresolution dynamic mode decomposition, *SIAM Journal on Applied Dynamical Systems*, **15** (2) (2016), 713–735.

[80]   J. N. Kutz, J. L. Proctor, and S. L. Brunton, Applied Koopman theory for partial differential equations and data-driven modeling of spatio-temporal systems, *Complexity*, (2018), 2018.

[81]   J.-C. Loiseau and S. L. Brunton, Constrained sparse Galerkin regression, *Journal of Fluid Mechanics*, **838** (2018), 42–67.

[82]   J.-C. Loiseau, B. R. Noack, and S. L. Brunton, Sparse reduced-order modeling: sensor-based dynamics to full-state estimation, *Journal of Fluid Mechanics*, **844** (2018), 459–490.

[83]   B. Lusch, J. N. Kutz, and S. L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics, *Nature Communications*, **9** (1) (2018), 4950.

[84]   A. Mackey, H. Schaeffer, and S. Osher, On the compressive spectral method, *Multiscale Modeling & Simulation*, **12** (4) (2014), 1800–1827.

[85]   M. W. Mahoney et al., Randomized algorithms for matrices and data, *Foundations and Trends in Machine Learning*, **3** (2) (2011), 123–224.

[86]   A. J. Majda and J. Harlim, Physics constrained nonlinear regression models for time series, *Nonlinearity*, **26** (1) (2012), 201.

[87]   A. J. Majda and Y. Lee, Conceptual dynamical models for turbulence, *Proceedings of the National Academy of Sciences*, **111** (18) (2014), 6548–6553.

[88]   N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Inferring biological networks by sparse identification of nonlinear dynamics, *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, **2** (1) (2016), 52–63.

[89]   N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, Model selection for dynamical systems via sparse regression and information criteria, *Proceedings of the Royal Society A*, **473** (2204) (2017), 1–16.

[90]   K. Manohar, S. L. Brunton, and J. N. Kutz, Environmental identification in flight using sparse approximation of wing strain, *Journal of Fluids and Structures*, **70** (2017), 162–180.

[91]   K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, Data-driven sparse sensor placement, *IEEE Control Systems Magazine*, **38** (3) (2018), 63–86.

[92]   A. Mardt, L. Pasquali, H. Wu, and F. Noé, VAMPnets: Deep learning of molecular kinetics, *Nature Communications*, **9** (5) (2018).

[93]   I. Mezić, Spectral properties of dynamical systems, model reduction and decompositions, *Nonlinear Dynamics*, **41** (1-3) (2005), 309–325.

[94]   I. Mezic, Analysis of fluid flows via spectral properties of the Koopman operator, *Annual Review of Fluid Mechanics*, **45** (2013), 357–378.

[95]   I. Mezić and A. Banaszuk, Comparison of systems with complex behavior, *Physica D: Nonlinear Phenomena*, **197** (1) (2004), 101–133.

[96]   L. Mirsky, Symmetric gauge functions and unitarily invariant norms, *Quarterly Journal of Mathematics*, **11** (1) (1960), 50–59.

[97]   C. C. Moore, Ergodic theorem, ergodic theory, and statistical mechanics, *Proceedings of the National Academy of Sciences*, **112** (7) (2015), 1907–1911.

[98]   B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele, A hierarchy of low-dimensional models for the transient and post-transient cylinder wake, *Journal of Fluid Mechanics*, **497** (2003), 335–363.

[99]   F. Noé and F. Nuske, A variational approach to modeling slow processes in stochastic dynamical systems, *Multiscale Modeling & Simulation*, **11** (2) (2013), 635–655.

[100] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. J. S. Mey, and F. Noé, Variational approach to molecular kinetics, *Journal of Chemical Theory and Computation*, **10** (4) (2014), 1739–1752.

[101] S. E. Otto and C. W. Rowley, Linearly recurrent autoencoder networks for learning dynamics, *SIAM Journal on Applied Dynamical Systems*, **18** (1) (2019), 558–593.

[102] V. Ozoliņš, R. Lai, R. Caflisch, and S. Osher, Compressed modes for variational problems in mathematics and physics, *Proceedings of the National Academy of Sciences*, **110** (46) (2013), 18368–18373.

[103] S. Peitz and S. Klus, Koopman operator-based model reduction for switched-system control of pdes, *Automatica*, **106** (2019), 184–191.

[104] V. Pereyra and G. Scherer, *Exponential Data Fitting and Its Applications*, Bentham Science Publishers, 2010.

[105] J. L. Proctor, S. L. Brunton, B. W. Brunton, and J. N. Kutz, Exploiting sparsity and equation-free architectures in complex systems (invited review), *The European Physical Journal Special Topics*, **223** (13) (2014), 2665–2684.

[106] J. L. Proctor, S. L. Brunton, and J. N. Kutz, Dynamic mode decomposition with control, *SIAM Journal on Applied Dynamical Systems*, **15** (1) (2016), 142–161.

[107] J. L. Proctor, S. L. Brunton, and J. N. Kutz, Generalizing Koopman theory to allow for inputs and control, *SIAM Journal on Applied Dynamical Systems*, **17** (1) (2018), 909–930.

[108] A. J. Roberts, *Model Emergent Dynamics in Complex Systems*, SIAM, 2014.

[109] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, Spectral analysis of nonlinear flows, *Journal of Fluid Mechanics*, **645** (2009), 115–127.

[110] C. W. Rowley, Model reduction for fluids, using balanced proper orthogonal decomposition, *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, **15** (03) (2005), 997–1013.

[111] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, Data-driven discovery of partial differential equations, *Science Advances*, **3** (2017), e1602614.

[112] D. Ruelle and F. Takens, On the nature of turbulence, *Communications in Mathematical Physics*, **20** (1971), 167–192.

[113] T. P. Sapsis and A. J. Majda, Statistically accurate low-order models for uncertainty quantification in turbulent dynamical systems, *Proceedings of the National Academy of Sciences*, **110** (34) (2013), 13705–13710.

[114] H. Schaeffer, R. Caflisch, C. D. Hauck, and S. Osher, Sparse dynamics for partial differential equations, *Proceedings of the National Academy of Sciences of the United States of America*, **110** (17) (2013), 6634–6639.

[115] H. Schaeffer, Learning partial differential equations via data discovery and sparse optimization, *Proceedings of the Royal Society A. Mathematical, Physical and Engineering Sciences*, **473** (2197) (2017), 20160446.

[116] H. Schaeffer and S. G. McCalla, Sparse model selection via integral terms, *Physical Review E*, **96** (2) (2017), 023302.

[117] P. J. Schmid, Dynamic mode decomposition of numerical and experimental data, *Journal of Fluid Mechanics*, **656** (August 2010), 5–28.

[118] E. Schmidt, Zur theorie der linearen und nichtlinearen integralgleichungen. 1. teil: Entwicklung willkürlicher funktionen nach systemen vorgeschriebener, *Mathematische Annalen*, **63** (1907), 433–476.

[119] M. Schmidt and H. Lipson, Distilling free-form natural laws from experimental data, *Science*, **324** (5923) (2009), 81–85.

[120] M. D. Schmidt, R. R. Vallabhajosyula, J. W. Jenkins, J. E. Hood, A. S. Soni, J. P. Wikswo, and H. Lipson, Automated refinement and inference of analytical models for metabolic networks, *Physical Biology*, **8** (5) (2011), 055011.

[121] O. Semeraro, F. Lusseyran, L. Pastur, and P. Jordan, Qualitative dynamics of wave packets in turbulent jets, *Physical Review Fluids*, **2** (9) (2017), 094605.

[122] L. Sirovich, Method of snapshots, *Quarterly of Applied Mathematics*, **45** (3) (1987), 561–571.

[123] M. Sorokina, S. Sygletos, and S. Turitsyn, Sparse identification for nonlinear optical communication systems: SINO method, *Optics Express*, **24** (26) (2016), 30433–30443.

[124] G. Sugihara, R. May, H. Ye, C.-h. Hsieh, E. Deyle, M. Fogarty, and S. Munch, Detecting causality in complex ecosystems, *Science*, **338** (6106) (2012), 496–500.

[125] A. Surana, Koopman operator based observer synthesis for control-affine nonlinear systems, in *55th IEEE Conference on Decision and Control (CDC)*, pp. 6492–6499, 2016.

[126] A. Surana and A. Banaszuk, Linear observer synthesis for nonlinear systems using Koopman operator framework, *IFAC-PapersOnLine*, **49** (18) (2016), 716–723.

[127] Y. Susuki and I. Mezić, A prony approximation of Koopman mode decomposition, in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. pp. 7022–7027, IEEE, 2015.

[128] A. Svenkeson, B. Glaz, S. Stanton, and B. J. West, Spectral decomposition of nonlinear systems with memory, *Physical Review E*, **93** (Feb 2016), 022211.

[129] K. Taira, S. L. Brunton, S. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley, Modal analysis of fluid flows: An overview, *AIAA Journal*, **55** (12) (2017), 4013–4041.

[130] N. Takeishi, Y. Kawahara, and T. Yairi, Learning Koopman invariant subspaces for dynamic mode decomposition, in *Advances in Neural Information Processing Systems*, pp. 1130–1140, 2017.

[131] F. Takens, Detecting strange attractors in turbulence, *Lecture Notes in Mathematics*, **898** (1981), 366–381.

[132] R. Tibshirani, Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society, Series B, Methodological*, (1996), 267–288.

[133] A. Towne, O. T. Schmidt, and T. Colonius, Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis, *Journal of Fluid Mechanics*, **847** (2018), 821–867.

[134] G. Tran and R. Ward, Exact recovery of chaotic systems from highly corrupted data, *Multiscale Modeling & Simulation*, **15** (3) (2017), 1108–1129.

[135] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, vol. 50, Siam, (1997).

[136] F. Tröltzsch and S. Volkwein, Pod a-posteriori error estimates for linear-quadratic optimal control problems, *Computational Optimization and Applications*, **44** (1) (2009), 83.

[137] J. A. Tropp and A. C. Gilbert, Signal recovery from random measurements via orthogonal matching pursuit, *IEEE Transactions on Information Theory*, **53** (12) (2007), 4655–4666.

[138] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz, On dynamic mode decomposition: theory and applications, *Journal of Computational Dynamics*, **1** (2) (2014), 391–421.

[139] J. H. Tu, C. W. Rowley, J. N. Kutz, and J. K. Shang, Spectral analysis of fluid flows using sub-nyquist-rate piv data, *Experiments in Fluids*, **55** (9) (2014), 1805.

[140] W. X. Wang, R. Yang, Y. C. Lai, V. Kovanis, and C. Grebogi, Predicting catastrophes in nonlinear dynamical systems by compressive sensing, *Physical Review Letters*, **106** (2011), 154101–1–154101–4.

[141] C. Wehmeyer and F. Noé, Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics, *Journal of Chemical Physics*, **148** (241703) (2018), 1–9.

[142] K. Willcox, Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition, *Computers & Fluids*, **35** (2) (2006), 208–226.

[143] K. Willcox and J. Peraire, Balanced model reduction via the proper orthogonal decomposition, *AIAA Journal*, **40** (11) (2002), 2323–2330.

[144] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, A data-driven approximation of the Koopman operator: extending dynamic mode decomposition, *Journal of Nonlinear Science*, **6** (2015), 1307–1346.

[145] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, A kernel approach to data-driven Koopman spectral analysis, *Journal of Computational Dynamics*, **2** (2) (2015), 247–265.

[146] M. O. Williams, P. J. Schmid, and J. N. Kutz, Hybrid reduced-order integration with proper orthogonal decomposition and dynamic mode decomposition, *Multiscale Modeling & Simulation*, **11** (2) (2013), 522–544.

[147] C. Yao and E. M. Bollt, Modeling and nonlinear parameter estimation with Kronecker product representation for coupled oscillators and spatiotemporal systems, *Physica D: Nonlinear Phenomena*, **227** (1) (2007), 78–99.

[148] H. Ye, R. J. Beamish, S. M. Glaser, S. C. H. Grant, C.-h. Hsieh, L. J. Richards, J. T. Schnute, and G. Sugihara, Equation-free mechanistic ecosystem forecasting using empirical dynamic modeling, *Proceedings of the National Academy of Sciences*, **112** (13) (2015), E1569–E1576.

[149] E. Yeung, S. Kundu, and N. Hodas, Learning deep neural network representations for Koopman operators of nonlinear dynamical systems, in *2019 American Control Conference (ACC)*. pp. 4832–4839, IEEE, 2019.

[150] B. Yildirim, C. Chryssostomidis, and G. E. Karniadakis, Efficient sensor placement for ocean measurements using low-dimensional concepts, *Ocean Modelling*, **27** (3) (2009), 160–173.

[151] Z. Zebib, Stability of viscous flow past a circular cylinder, *Journal of Engineering Mathematics*, **21** (1987), 155–165.

[152] L. Zhang and H. Schaeffer, On the convergence of the SINDy algorithm, *Multiscale Modeling & Simulation*, **17** (3) (2019), 948–972.

[153] W. Zhang, B. Wang, Z. Ye, and J. Quan, Efficient method for limit cycle flutter analysis based on nonlinear aerodynamic reduced-order models, *AIAA Journal*, **50** (5) (2012), 1019–1028.

[154] P. Zheng, T. Askham, S. L. Brunton, J. N. Kutz, and A. Y. Aravkin, A unified framework for sparse relaxed regularized regression: Sr3, *IEEE Access*, **7** (2018), 1404–1423.

[155] H. Zou and T. Hastie, Regularization and variable selection via the elastic net, *Journal of the Royal Statistical Society, Series B, Statistical Methodology*, **67** (2) (2005), 301–320.

Index