Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 397 (2022) 115122
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Interpolatory tensorial reduced order models for parametric dynamical systems

Alexander V. Mamonov, Maxim A. Olshanskii*

Department of Mathematics, University of Houston, Houston, TX 77204, United States of America

Received 9 November 2021; received in revised form 10 May 2022; accepted 12 May 2022
Available online xxxx

Abstract

The paper introduces a reduced order model (ROM) for numerical integration of a dynamical system which depends on multiple parameters. The ROM is a projection of the dynamical system on a low dimensional space that is both problem-dependent and *parameter-specific*. The ROM exploits compressed tensor formats to find a low rank representation for a sample of high-fidelity snapshots of the system state. This tensorial representation provides ROM with an orthogonal basis in a universal space of all snapshots *and* encodes information about the state variation in parameter domain. During the online phase and for any incoming parameter, this information is used to find a reduced basis that spans a parameter-specific subspace in the universal space. The computational cost of the online phase then depends only on tensor compression ranks, but not on space or time resolution of high-fidelity computations. Moreover, certain compressed tensor formats enable to avoid the adverse effect of parameter space dimension on the online costs (known as the curse of dimension). The analysis of the approach includes an estimate for the representation power of the acquired ROM basis. We illustrate the performance and prediction properties of the ROM with several numerical experiments, where tensorial ROM's complexity and accuracy is compared to those of conventional POD-ROM.

Keywords: Model order reduction; Parametric PDEs; Low-rank tensors; Dynamical systems; Proper orthogonal decomposition

1. Introduction

In numerical optimal control, inverse modeling or uncertainty quantification, one commonly needs to integrate a parameter-dependent dynamical system for various values of the parameter vector. For example, inverse modeling may require repeated solutions of the forward problem represented by a dynamical system, along the search path in a high-dimensional parameter space. This may lead to extreme-scale computations that, if implemented straightforwardly, often result in overwhelming computational costs. Reduced order models (ROMs) offer a possibility to alleviate these costs by replacing a high-fidelity model with a low-dimensional surrogate model [1,2]. Thanks to this practical value and apparent success, ROMs for parametric dynamical systems have already attracted considerable attention; see, e.g., [3–8].

* Corresponding author.
E-mail addresses: avmamonov@uh.edu (A.V. Mamonov), maolshanskiy@uh.edu (M.A. Olshanskii).

https://doi.org/10.1016/j.cma.2022.115122
0045-7825/© 2022 Elsevier B.V. All rights reserved.

In this paper, we are interested in projection based ROMs that build the surrogate model by projecting a high-fidelity model onto a low-dimensional problem-dependent vector space [3]. Projection-based ROMs for dynamical systems include such well-known model order reduction techniques as proper orthogonal decomposition (POD) ROMs [9,10] (and its variants such as POD-DEIM [11] and balanced POD [12]) and PGD-ROMs [13,14]. In these approaches the basis for the projection space is computed by building on the information about the dynamical system provided through high-fidelity solutions sampled for certain time instances and/or parameter values, the so-called solution snapshots. However, building a general low-dimensional space for all times and parameters of interest might be challenging, if possible at all, for wide parameter ranges and long times. Several studies aimed to address this challenge: In [15–17] the authors considered partitioning strategies which introduce a subdivision of the parameter domain and assign an individual local reduced-order basis to each subdomain offline. Another idea [18,19] is to adapt precomputed equal-dimension reduced order spaces by interpolating them for out-of-sample parameters along geodesics on the Grassmann manifold. The present paper introduces a different approach that builds on recent developments in tensor decompositions and low-rank approximations to quickly compute parameter-specific reduced bases for projection based ROMs.

For parameterized systems of time-dependent differential equations, the generated data (the input of ROM) naturally takes a form of a multi-dimensional tensor of solution snapshots, with dimensionality $D+2$, where D is the dimension of the parameter space and 2 accounts for the spatial and time-wise distributions. Modern ROMs often proceed by unfolding such tensors into a matrix to perform standard POD based on truncated SVD. This leads to the loss of information about the dependency of solutions on parameters. We propose to overcome these issues by working directly with tensor data, and by exploiting low-rank tensor approximations based on the canonical polyadic (CP), high order SVD (HOSVD), and Tensor Train (TT) decompositions. The approach consists of two stages. First, at the offline stage, the compressed snapshot tensor is computed using one of the three tensor decompositions, thus preserving the essential information about variation of the solution with respect to parameters. Up to the compression accuracy, each of these decompositions provides a (global) basis for the universal space spanned by all observed snapshots. The so-called core of the compressed representation is then transmitted to the second stage, referred to as the online stage. At the online stage the transmitted part of compressed tensor allows for a fast computation of a *parameter-specific* reduced basis for any incoming out-of-sample parameter vector through an interpolation and fast linear algebra routines. The reduced order basis is then given in terms of its coordinates in the global basis that can be stored offline. For CP and TT formats, the cost of these computations is free of exponential growth with respect to the parameter space dimension. On analysis side of this work, we prove an estimate for prediction power of the parameter-specific reduced order basis. The estimate explicitly depends on the approximation accuracy of the original tensor by the compressed one, parameter interpolation error, and singular values of a small-size parameter-specific matrix.

Despite an outstanding recent progress in numerical multi-linear algebra and, in particular, in understanding tensor decompositions (see, e.g., review articles [20–22]), the application of tensor methods in reduced order modeling of dynamical systems is still rather scarce. We mention two reports by Nouy [23,24], who reviewed tensor compressed formats and discussed their possible use for sparse function representation and reduced order modeling, as well as a series of publications on the treatment in compressed tensor formats of algebraic systems resulting from the stochastic and parametric Galerkin finite element method, see e.g. [25–29]. A POD-ROM was combined with a low-rank tensor representation of a mapping from a parameter space onto an output domain in [30]. The authors of survey [3] observe that "The combination of tensor calculus ... and parametric model reduction techniques for time dependent problems is still in its infancy, but offers a promising research direction". We believe the statement holds true, and the present study contributes to this largely open research field.

The remainder of the paper is organized as follows. In Section 2 we set up a parameter-dependent Cauchy problem and recall the basics of POD-ROM approach that is needed for reference purpose later in the text. Section 3 introduces a general idea of the interpolatory tensorial ROM and considers its realization using three popular tensor compression formats. Details are worked out for a Cartesian grid-based sampling of the parameter domain, and then the approach is extended to a more general parameter sampling scheme. A separate subsection discusses online–offline complexity and storage requirements of the method. An estimate on the prediction power of the reduced order basis is proved in Section 4. Numerical examples in Section 5 illustrate the analysis and performance of the method. In particular, we compare the delivered accuracy with standard POD-ROM that employs a global low-dimensional basis.

2. Parameterized Cauchy problem and the conventional POD-ROM

To fix ideas, consider the following multi-parameter initial value problem. For a vector of parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_D)$ from the parameter domain $\mathcal{A} \subset \mathbb{R}^D$ find the trajectory $\mathbf{u} = \mathbf{u}(t, \boldsymbol{\alpha}) : [0, T) \to \mathbb{R}^M$ solving

$$\mathbf{u}_t = F(t, \mathbf{u}, \boldsymbol{\alpha}), \quad t \in (0, T), \quad \text{and } \mathbf{u}|_{t=0} = \mathbf{u}_0, \tag{2.1}$$

with a given continuous flow field $F : (0, T) \times \mathbb{R}^M \times \mathcal{A} \to \mathbb{R}^M$. Hereafter we denote all vector quantities by bold lowercase letters. We assume that the unique solution exists on $(0, T)$ for all $\boldsymbol{\alpha} \in \mathcal{A}$. Examples considered in this paper include parameter-dependent parabolic equations, in which case one can think of (2.1) as a system of ODEs for nodal values of the finite volume or finite element solution to the PDE problem, where material coefficients, boundary conditions, or the computational domain (via a mapping into a reference domain) are parameterized by $\boldsymbol{\alpha}$.

We are interested in projection based ROMs, where for an arbitrary but fixed $\boldsymbol{\alpha} \in \mathcal{A}$ an approximation to $\mathbf{u}$ is sought as a solution to equations projected onto a reduced space. Projection based approaches aim at retaining the structure of the model and thus at preserving the physics present in the high-fidelity model [3]. Among the projection based approaches to model reduction for time-dependent differential equations, Proper Orthogonal Decomposition (POD) and its variants are likely the most widely used ROM technique, which provides tools to represent trajectories of a dynamical system in a low-dimensional, problem-dependent basis [31–33]. We summarize the POD-ROM below for further reference and for the purpose of comparison to our approach in Section 5.

Assume for a moment that $\boldsymbol{\alpha}$ is *fixed*. The POD-ROM computes a representative collection of states $\boldsymbol{\phi}_k(\boldsymbol{\alpha}) = \mathbf{u}(t_k, \boldsymbol{\alpha}) \in \mathbb{R}^M$ at times $0 \le t_1, \ldots, t_N < T$, referred to as snapshots, through high-fidelity numerical simulations. Next, one finds a parameter-specific low-dimensional basis $\{\mathbf{z}_i^{\text{pod}}(\boldsymbol{\alpha})\}_{i=1}^n \subset \mathbb{R}^M$, $n \ll N$, referred to hereafter as the *reduced basis*, such that the projection subspace $\operatorname{span}\{\mathbf{z}_1^{\text{pod}}(\boldsymbol{\alpha}), \ldots, \mathbf{z}_n^{\text{pod}}(\boldsymbol{\alpha})\}$ approximates the snapshot space $\operatorname{span}\{\boldsymbol{\phi}_1(\boldsymbol{\alpha}), \ldots, \boldsymbol{\phi}_N(\boldsymbol{\alpha})\}$ in the best possible way.

To determine the reduced basis, form a matrix of snapshots

$$\Phi_{\text{pod}}(\boldsymbol{\alpha}) = [\boldsymbol{\phi}_1(\boldsymbol{\alpha}), \ldots, \boldsymbol{\phi}_N(\boldsymbol{\alpha})] \in \mathbb{R}^{M \times N}, \tag{2.2}$$

compute its SVD

$$\Phi_{\text{pod}}(\boldsymbol{\alpha}) = \mathrm{U}\Sigma\mathrm{V}^T, \tag{2.3}$$

and define $\mathbf{z}_i^{\text{pod}}(\boldsymbol{\alpha})$, $i = 1, \ldots, n$, to be the first n left singular vectors of $\Phi_{\text{pod}}(\boldsymbol{\alpha})$, i.e., the first n columns of U. Hereafter we denote all matrices with upright capital letters. The singular values in Σ provide information about the approximation power of $\operatorname{span}\{\mathbf{z}_1^{\text{pod}}(\boldsymbol{\alpha}), \ldots, \mathbf{z}_n^{\text{pod}}(\boldsymbol{\alpha})\}$. We refer to [31,34] and references therein for a discussion about algebraically different ways to define POD and their equivalence.

For parameters $\boldsymbol{\alpha}$ varying in $\mathcal{A}$, a parametric POD-ROM builds a global reduced basis by sampling the parameter domain, generating snapshots for each sampled parameter value and proceeding with SVD (2.3) for a cumulative matrix of all snapshots. Possible sampling strategies include using a Cartesian grid in $\mathcal{A}$, Monte-Carlo methods, and greedy algorithms based on a posteriori error estimates; see, e.g., [3,4]. Regardless of the sampling procedure, the resulting basis can accurately reproduce only the data from which it originated. Without parameter-specificity, the basis may lack robustness for out-of-sample parameters, i.e., away from the reference simulations. This is a serious limitation for using POD based ROMs in inverse modeling. We plan to address this limitation by introducing tensorial techniques for finding reduced bases that are both problem- and parameter-specific.

3. Tensorial ROMs

We first consider in Section 3.1 a Cartesian grid-based sampling of the parameter domain $\mathcal{A}$ in the case when $\mathcal{A}$ is the D-dimensional box

$$\mathcal{A} = \bigotimes_{i=1}^{D} [\alpha_i^{\min}, \alpha_i^{\max}], \tag{3.1}$$

and the sampling points are placed at the nodes of a Cartesian grid. Next, we describe three tensorial ROMs (**TROM**s) based on three different tensor decompositions, canonical polyadic (CP, Section 3.5), high order SVD (HOSVD, Section 3.6) and tensor train (TT, Section 3.7).

3.1. Cartesian grid-based parameter sampling

To generate the sampling set $\widehat{\mathcal{A}}$, we distribute n_i nodes $\{\widehat{\alpha}_i^j\}_{j=1,\dots,n_i}$ within each of the intervals $[\alpha_i^{\min}, \alpha_i^{\max}]$ in (3.1) for $i = 1, \dots, D$, and define

$$\widehat{\mathcal{A}} = \left\{ \widehat{\boldsymbol{\alpha}} = (\widehat{\alpha}_1, \dots, \widehat{\alpha}_D)^T \ : \ \widehat{\alpha}_i \in \{\widehat{\alpha}_i^j\}_{j=1,\dots,n_i}, \ i = 1, \dots, D \right\}. \tag{3.2}$$

Hereafter we use hats to denote parameters from the sampling set $\widehat{\mathcal{A}}$, and the cardinality of $\widehat{\mathcal{A}}$ is denoted by

$$K = \prod_{i=1}^{D} n_i. \tag{3.3}$$

The corresponding snapshots $\boldsymbol{\phi}_k(\widehat{\boldsymbol{\alpha}}) = \mathbf{u}(t_k, \widehat{\boldsymbol{\alpha}})$, $\widehat{\boldsymbol{\alpha}} \in \widehat{\mathcal{A}}$, are organized in a multi-dimensional array

$$(\boldsymbol{\Phi})_{:,i_1,\dots,i_D,k} = \boldsymbol{\phi}_k(\widehat{\alpha}_1^{i_1}, \dots, \widehat{\alpha}_D^{i_D}), \tag{3.4}$$

which is a tensor of order $D+2$ and size $M \times n_1 \times \dots \times n_D \times N$. We reserve the first and the last indices of $\boldsymbol{\Phi}$ for the spatial and temporal distributions, respectively. All tensors hereafter are denoted with bold uppercase letters.

Unfolding $\boldsymbol{\Phi}$ along the first index in an $M \times (n_1 \cdots n_D)N$ matrix and applying (truncated) SVD to determine the first n left singular vectors is equivalent to the POD with grid-based parameter sampling. The disadvantage of this approach for ROM construction is that it neglects any information about the dependence of snapshots on parameter variation reflected in the tensor structure of $\boldsymbol{\Phi}$. To preserve this information, we proceed with a compressed approximation $\widetilde{\boldsymbol{\Phi}}$ of $\boldsymbol{\Phi}$ rather than with the low rank approximation of the unfolded matrix.

3.2. Tensor compression and universal space

The notion of a tensor rank and low-rank tensor approximation is ambiguous and later in this section we consider three popular compressed tensor formats. For now we only assume that $\widetilde{\boldsymbol{\Phi}}$ satisfies

$$\left\| \widetilde{\boldsymbol{\Phi}} - \boldsymbol{\Phi} \right\|_F \le \widetilde{\varepsilon} \left\| \boldsymbol{\Phi} \right\|_F \tag{3.5}$$

for some small $\widetilde{\varepsilon} > 0$, where tensor Frobenius norm is simply

$$\| \boldsymbol{\Phi} \|_F := \Big(\sum_{j=1}^{M} \sum_{i_1=1}^{n_1} \cdots \sum_{i_D=1}^{n_D} \sum_{k=1}^{N} \boldsymbol{\Phi}_{j,i_1,\dots,i_D,k}^2 \Big)^{1/2} \tag{3.6}$$

The "low-rank" (compressed) tensor $\widetilde{\boldsymbol{\Phi}}$ is computed during the first, *offline* stage of TROM construction and *a part of* $\widetilde{\boldsymbol{\Phi}}$ is passed on to the second, *online* stage which uses this information about variation of snapshots with respect to changes in parameters to compute a parameter-specific TROM.

We call *universal space* the space $\widetilde{V}$ spanned by the first-mode fibers of $\widetilde{\boldsymbol{\Phi}}$, i.e., $\widetilde{V}$ is the column space of the mode-1 unfolding matrix. For the exact decomposition (i.e., for $\widetilde{\varepsilon} = 0$), $\widetilde{V}$ is the space of all observed system states. In general, $\widetilde{V}$ depends on a compression format, dimension of $\widetilde{V}$ does not exceed M and depends on $\widetilde{\varepsilon}$ and snapshot variation. We shall see that $\widetilde{V}$ does approximate the full space of high-fidelity snapshots, while the CP, HOSVD and TT formats all deliver an orthogonal basis for $\widetilde{V}$. In the online stage of TROM we find a local (parameter-specific) ROM basis by specifying its coordinates in $\widetilde{V}$.

3.3. In-sample prediction

During the online stage we wish to be able to approximately solve (2.1) for an arbitrary parameter $\boldsymbol{\alpha} \in \mathcal{A}$ in an $\boldsymbol{\alpha}$*-specific* reduced basis. For this step we need to introduce the notion of k-mode tensor–vector product $\boldsymbol{\Psi} \times_k \mathbf{a}$ of a tensor $\boldsymbol{\Psi} \in \mathbb{R}^{N_1 \times \cdots \times N_m}$ of order m and a vector $\mathbf{a} \in \mathbb{R}^{N_k}$: the resulting tensor $\boldsymbol{\Psi} \times_k \mathbf{a}$ has order $m-1$ and size $N_1 \times \cdots \times N_{k-1} \times N_{k+1} \times \cdots \times N_m$. Specifically, elementwise

$$(\boldsymbol{\Psi} \times_k \mathbf{a})_{j_1,\dots,j_{k-1},j_{k+1},\dots,j_m} = \sum_{j_k=1}^{N_k} \boldsymbol{\Psi}_{j_1,\dots,j_m} a_{j_k}. \tag{3.7}$$

For a moment, consider some $\widehat{\boldsymbol{\alpha}} = (\widehat{\alpha}_1, \dots, \widehat{\alpha}_D)^T$ from the sampling set $\widehat{\mathcal{A}}$ and define D vectors, $\mathbf{e}^i(\widehat{\boldsymbol{\alpha}}) = \left(e_1^i(\widehat{\boldsymbol{\alpha}}), \dots, e_{n_i}^i(\widehat{\boldsymbol{\alpha}})\right)^T \in \mathbb{R}^{n_i}$, $i = 1, \dots, D$, as

$$e_j^i(\widehat{\boldsymbol{\alpha}}) = \begin{cases} 1 & \text{if } \widehat{\alpha}_i = \widehat{\alpha}_i^j \\ 0 & \text{otherwise,} \end{cases} \qquad j = 1, \dots, n_i. \tag{3.8}$$

In other words, $\mathbf{e}^i(\widehat{\boldsymbol{\alpha}})$ encodes the position of $\widehat{\alpha}_i$ among the grid nodes on $[\alpha_i^{\min}, \alpha_i^{\max}]$, $i = 1, \dots, D$.

Vectors $\mathbf{e}^i(\widehat{\boldsymbol{\alpha}})$ defined above allow us to extract the snapshots corresponding to a particular $\widehat{\boldsymbol{\alpha}} \in \widehat{\mathcal{A}}$. Specifically, we introduce the following *extraction* operation

$$\Phi_e(\widehat{\boldsymbol{\alpha}}) = \boldsymbol{\Phi} \times_2 \mathbf{e}^1(\widehat{\boldsymbol{\alpha}}) \times_3 \mathbf{e}^2(\widehat{\boldsymbol{\alpha}}) \dots \times_{D+1} \mathbf{e}^D(\widehat{\boldsymbol{\alpha}}) \in \mathbb{R}^{M \times N}, \tag{3.9}$$

which extracts from tensor of all snapshots $\boldsymbol{\Phi}$ the matrix of snapshots (2.2) for the particular $\widehat{\boldsymbol{\alpha}} \in \widehat{\mathcal{A}}$, i.e., $\Phi_e(\widehat{\boldsymbol{\alpha}}) = \Phi_{\text{pod}}(\widehat{\boldsymbol{\alpha}})$.

Combining (3.9) with compressed approximation (3.5), we conclude that is should be possible to extract from $\widetilde{\boldsymbol{\Phi}}$ the information about the space spanned by the snapshots $\{\mathbf{u}(t_i, \widehat{\boldsymbol{\alpha}})\}_{i=1}^N$, for a particular $\widehat{\boldsymbol{\alpha}} \in \widehat{\mathcal{A}}$ up to the accuracy of approximation in (3.5). Indeed, let $\boldsymbol{\phi}_i(\widehat{\boldsymbol{\alpha}}) = \mathbf{u}(t_i, \widehat{\boldsymbol{\alpha}})$, $i = 1, \dots, N$, and denote by $\{\mathbf{z}_j(\widehat{\boldsymbol{\alpha}})\}_{j=1}^{\widetilde{N}}$, $\widetilde{N} \le N$, an orthonormal basis for the column space of

$$\widetilde{\Phi}_e(\widehat{\boldsymbol{\alpha}}) = \widetilde{\boldsymbol{\Phi}} \times_2 \mathbf{e}^1(\widehat{\boldsymbol{\alpha}}) \times_3 \mathbf{e}^2(\widehat{\boldsymbol{\alpha}}) \dots \times_{D+1} \mathbf{e}^D(\widehat{\boldsymbol{\alpha}}) \in \mathbb{R}^{M \times N}, \tag{3.10}$$

where $\widetilde{N} = \operatorname{rank}\left(\widetilde{\Phi}_e(\widehat{\boldsymbol{\alpha}})\right)$. Then, it holds

$$\sum_{i=1}^{N} \left\| \boldsymbol{\phi}_i - \sum_{j=1}^{\widetilde{N}} \langle \boldsymbol{\phi}_i, \mathbf{z}_j \rangle \mathbf{z}_j \right\|_{\ell^2}^2 \le \widetilde{\varepsilon}^2 \| \boldsymbol{\Phi} \|_F^2, \tag{3.11}$$

where we use the shortcut $\boldsymbol{\phi}_i = \boldsymbol{\phi}_i(\widehat{\boldsymbol{\alpha}})$, $\mathbf{z}_i = \mathbf{z}_i(\widehat{\boldsymbol{\alpha}})$. To establish (3.11), consider (thin) SVD $\widetilde{\Phi}_e(\widehat{\boldsymbol{\alpha}}) = \widetilde{\mathrm{U}}\widetilde{\Sigma}\widetilde{\mathrm{V}}^T$ and compute

$$\begin{aligned} \sum_{i=1}^{N} \left\| \boldsymbol{\phi}_i - \sum_{j=1}^{\widetilde{N}} \langle \boldsymbol{\phi}_i, \mathbf{z}_j \rangle \mathbf{z}_j \right\|_{\ell^2}^2 &= \left\| (\mathrm{I} - \widetilde{\mathrm{U}}\widetilde{\mathrm{U}}^T) \Phi_e(\widehat{\boldsymbol{\alpha}}) \right\|_F^2 \\ &= \left\| (\mathrm{I} - \widetilde{\mathrm{U}}\widetilde{\mathrm{U}}^T) \left(\Phi_e(\widehat{\boldsymbol{\alpha}}) - \widetilde{\Phi}_e(\widehat{\boldsymbol{\alpha}}) \right) \right\|_F^2 \\ &\le \left\| \mathrm{I} - \widetilde{\mathrm{U}}\widetilde{\mathrm{U}}^T \right\|^2 \left\| \Phi_e(\widehat{\boldsymbol{\alpha}}) - \widetilde{\Phi}_e(\widehat{\boldsymbol{\alpha}}) \right\|_F^2 \le \left\| \Phi_e(\widehat{\boldsymbol{\alpha}}) - \widetilde{\Phi}_e(\widehat{\boldsymbol{\alpha}}) \right\|_F^2 \\ &= \left\| (\boldsymbol{\Phi} - \widetilde{\boldsymbol{\Phi}}) \times_2 \mathbf{e}^1(\widehat{\boldsymbol{\alpha}}) \times_3 \mathbf{e}^2(\widehat{\boldsymbol{\alpha}}) \dots \times_{D+1} \mathbf{e}^D(\widehat{\boldsymbol{\alpha}}) \right\|_F^2 \\ &\le \left\| \boldsymbol{\Phi} - \widetilde{\boldsymbol{\Phi}} \right\|_F^2 \le \widetilde{\varepsilon}^2 \| \boldsymbol{\Phi} \|_F^2, \end{aligned}$$

where we used linearity of (3.7) and the inequality $\|\mathrm{AB}\|_F \le \|\mathrm{A}\| \|\mathrm{B}\|_F$ for matrices A, B, and spectral matrix norm $\| \cdot \|$. We also used $\|\mathrm{P}\| \le 1$ for an orthogonal projection matrix $\mathrm{P} = \mathrm{I} - \widetilde{\mathrm{U}}\widetilde{\mathrm{U}}^T$.

Since $\mathbf{z}_i(\widehat{\boldsymbol{\alpha}}) \in \widetilde{V}$ for all in-sample $\widehat{\boldsymbol{\alpha}}$, the bound in (3.11) provides an estimate on how accurate the true snapshots can be approximated in the universal space. Furthermore, from (3.11) we conclude that given $\widetilde{\boldsymbol{\Phi}}$ and $\widehat{\boldsymbol{\alpha}} \in \widehat{\mathcal{A}}$ we can obtain a parameter-specific (quasi)-optimal reduced basis by taking the first n left singular vectors of $\widetilde{\Phi}_e(\widehat{\boldsymbol{\alpha}})$. Representation power of this basis is determined by $\widetilde{\varepsilon}$ from (3.5) and σ_i, $i > n$, the singular values of $\widetilde{\Phi}_e(\widehat{\boldsymbol{\alpha}})$. If $\widetilde{\varepsilon}$ is sufficiently small, i.e., the snapshot tensor admits an efficient low-rank representation, then the computed reduced basis better represents the snapshot space *for a given* $\widehat{\boldsymbol{\alpha}}$ than the first n left singular vectors of the unfolded snapshot matrix (i.e., better than the POD basis); see numerical results in Section 5 which show up to several orders of accuracy gain for some of the examples.

The arguments above apply only to parameter values $\widehat{\boldsymbol{\alpha}}$ from the sampling set $\widehat{\mathcal{A}} \subset \mathcal{A}$. Next, we consider ROM basis computation for an arbitrary $\boldsymbol{\alpha} \in \mathcal{A}$ that may not necessarily come from the training set $\widehat{\mathcal{A}}$, the so-called *out-of-sample* $\boldsymbol{\alpha}$. Below we explore the option of building ROM basis for an out-of-sample $\boldsymbol{\alpha}$ using interpolation in the parameter space. This approach is based on an assumption of smooth dependence of the solution $\mathbf{u}(t, \boldsymbol{\alpha})$ of (2.1) on $\boldsymbol{\alpha}$. We refer to the corresponding tensorial ROMs as *interpolatory* TROMs.

3.4. Interpolatory TROM

To construct parameter-specific ROM basis for an arbitrary $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_D)^T \in \mathcal{A}$ we introduce the interpolation procedure defined by

$$\mathbf{e}^i : \boldsymbol{\alpha} \to \mathbb{R}^{n_i}, \quad i = 1, \dots, D. \tag{3.12}$$

Entrywise, we write $\mathbf{e}^i(\boldsymbol{\alpha}) = \left(e^i_1(\boldsymbol{\alpha}), \dots, e^i_{n_i}(\boldsymbol{\alpha})\right)^T \in \mathbb{R}^{n_i}$. The interpolation procedure should satisfy the following property. For a smooth function $g : [\alpha_i^{\min}, \alpha_i^{\max}] \to \mathbb{R}$, it holds

$$g(\alpha_i) \approx \sum_{j=1}^{n_i} e^i_j(\boldsymbol{\alpha}) g(\widehat{\alpha}^j_i), \quad i = 1, \dots, D, \tag{3.13}$$

where $\widehat{\alpha}^j_i$, $j = 1, \dots, n_i$, are the grid nodes on $[\alpha_i^{\min}, \alpha_i^{\max}]$.

We further consider Lagrangian interpolation of order p: for a given $\boldsymbol{\alpha} \in \mathcal{A}$ let $\widehat{\alpha}^{i_1}_i, \dots, \widehat{\alpha}^{i_p}_i$ be the p closest grid nodes to α_i on $[\alpha_i^{\min}, \alpha_i^{\max}]$, for $i = 1, \dots, D$. Then

$$e^i_j(\boldsymbol{\alpha}) = \begin{cases} \prod\limits_{\substack{m=1,\\ m\neq k}}^{p} (\widehat{\alpha}^{i_m}_i - \alpha_i) \Big/ \prod\limits_{\substack{m=1,\\ m\neq k}}^{p} (\widehat{\alpha}^{i_m}_i - \widehat{\alpha}^{j}_i), & \text{if } j = i_k \in \{i_1, \dots, i_p\}, \\ 0, & \text{otherwise}, \end{cases} \tag{3.14}$$

are the entries of $\mathbf{e}^i(\boldsymbol{\alpha})$ for $j = 1, \dots, n_i$. For the numerical experiments in Section 5 we use $p = 2$ or 3, i.e., linear or quadratic interpolation. The Lagrangian interpolation is not the only option and depending on parameter sampling and solution smoothness other fitting procedures can be more suitable.

Vectors $\mathbf{e}^i$ extend the notion of position vectors $\mathbf{e}^i$ defined in (3.8) for out-of-sample vectors. Indeed, from (3.14) it is easy to see that both vectors coincide if $\widehat{\boldsymbol{\alpha}} \in \widehat{\mathcal{A}}$ and so we use the same notation hereafter. Therefore, we can define a snapshot matrix $\widetilde{\Phi}_e(\boldsymbol{\alpha})$ through the extraction–interpolation procedure:

$$\widetilde{\Phi}_e(\boldsymbol{\alpha}) = \widetilde{\boldsymbol{\Phi}} \times_2 \mathbf{e}^1(\boldsymbol{\alpha}) \times_3 \mathbf{e}^2(\boldsymbol{\alpha}) \dots \times_{D+1} \mathbf{e}^D(\boldsymbol{\alpha}) \in \mathbb{R}^{M\times N}, \tag{3.15}$$

to generalize (3.9) for any $\boldsymbol{\alpha} \in \mathcal{A}$. Note that (3.15) and (3.9) are the same for $\boldsymbol{\alpha} = \widehat{\boldsymbol{\alpha}} \in \widehat{\mathcal{A}} \subset \mathcal{A}$, while (3.15) defines $\widetilde{\Phi}_e(\boldsymbol{\alpha})$ also for out-of-sample parameter vectors. If the low-rank representation of the snapshot tensor is exact, i.e., $\widetilde{\boldsymbol{\Phi}} = \boldsymbol{\Phi}$, then $\widetilde{\Phi}_e(\boldsymbol{\alpha})$ is the interpolation of the snapshot matrices $\Phi_{\text{pod}}(\widehat{\boldsymbol{\alpha}})$.

Once $\boldsymbol{\alpha} \in \mathcal{A}$ is fixed and $\widetilde{\Phi}_e(\boldsymbol{\alpha})$ is given by (3.15), our *parameter-specific reduced basis* $\{\mathbf{z}_i(\boldsymbol{\alpha})\}_{i=1}^n$ is defined as the first n left singular vectors of $\widetilde{\Phi}_e(\boldsymbol{\alpha})$. Later we demonstrate that the coordinates of this basis in the universal space can be calculated quickly (i.e., using only low-dimensional calculations) online without actually computing $\widetilde{\Phi}_e(\boldsymbol{\alpha})$.

In a *non-interpolatory* TROM, a parameter-specific reduced basis can be constructed as follows. Choose $p \geq 2$ and fix $\boldsymbol{\alpha} \in \mathcal{A}$, then let $\widehat{\alpha}^{i_1}_i, \dots, \widehat{\alpha}^{i_p}_i$ be the p closest grid nodes to α_i on $[\alpha_i^{\min}, \alpha_i^{\max}]$, for $i = 1, \dots, D$, similarly to the interpolatory construction above. Define the set

$$\widehat{\mathcal{A}}_p := \left\{ \widehat{\boldsymbol{\alpha}} = (\widehat{\alpha}_1, \dots, \widehat{\alpha}_D)^T \; : \; \widehat{\alpha}_i \in \{\widehat{\alpha}^j_i\}_{j\in\{i_1,\dots,i_p\}}, \; i = 1, \dots, D \right\} \subset \widehat{\mathcal{A}}. \tag{3.16}$$

Then, assemble a large matrix by concatenating $\widetilde{\Phi}_e(\widehat{\boldsymbol{\alpha}})$ for all $\widehat{\boldsymbol{\alpha}} \in \widehat{\mathcal{A}}_p$ and take $\{\mathbf{z}_i(\boldsymbol{\alpha})\}_{i=1}^n$ to be its first n left singular vectors. Of course, hybrid strategies (e.g., interpolation only in some parameter directions) are also possible. For non-interpolatory or hybrid TROMs it is also possible to compute local basis online with only low-dimensional calculations following same steps as considered below.

In the rest of paper we focus on the interpolatory TROM and consider three well-known compressed formats for low rank tensor approximation $\widetilde{\boldsymbol{\Phi}} \approx \boldsymbol{\Phi}$: canonical polyadic (CP), Tucker, a.k.a. higher order SVD (HOSVD), and tensor train (TT) decomposition formats. Note that the notion of tensor rank(s) differs among these formats. When applied to TROM computation, these formats lead to different offline computational costs to build $\widetilde{\boldsymbol{\Phi}}$, different amounts of information transmitted from the offline stage to the online stage (measured by the compression rate, as explained in Section 3.9), and slightly varying amounts of online computations for finding the reduced basis given an incoming $\boldsymbol{\alpha} \in \mathcal{A}$.

3.5. CP-TROM

The first tensor decomposition that we consider is the canonical polyadic decomposition of a tensor into the sum of rank one tensors [20,35–37]. In CP-TROM we approximate $\boldsymbol{\Phi}$ by the sum of R (where R is the so-called canonical tensor rank) direct products of $D+2$ vectors $\mathbf{u}^r \in \mathbb{R}^M$, $\boldsymbol{\sigma}_i^r \in \mathbb{R}^{n_i}$, $i=1,\dots,D$, and $\mathbf{v}^r \in \mathbb{R}^N$,

$$\boldsymbol{\Phi} \approx \widetilde{\boldsymbol{\Phi}} = \sum_{r=1}^{R} \mathbf{u}^r \circ \boldsymbol{\sigma}_1^r \circ \cdots \circ \boldsymbol{\sigma}_D^r \circ \mathbf{v}^r, \tag{3.17}$$

or entry-wise

$$(\widetilde{\boldsymbol{\Phi}})_{j,i_1,\dots,i_D,k} = \sum_{r=1}^{R} u_j^r \sigma_{1,i_1}^r \dots \sigma_{D,i_D}^r v_k^r.$$

CP decomposition often delivers excellent compression. However, there are well-known difficulties in determining the accurate canonical rank R and working with the CP format, see, e.g., [38,39]. Since we are interested in the *approximation* $\widetilde{\boldsymbol{\Phi}}$ to $\boldsymbol{\Phi}$, the alternating least squares (ALS) algorithm [20,40] can be used to minimize $\|\boldsymbol{\Phi}-\widetilde{\boldsymbol{\Phi}}\|_F$ for a specified target canonical rank R to find the approximate factors $\mathbf{u}^r \in \mathbb{R}^M$, $\boldsymbol{\sigma}_i^r \in \mathbb{R}^{n_i}$, and $\mathbf{v}^r \in \mathbb{R}^N$, $r=1,\dots,R$. We further assume $R \le M$, where M is the dimension of high-fidelity snapshots.

Note that the second-mode product of a $D+2$-dimensional rank-one tensor $\mathbf{u}^r \circ \boldsymbol{\sigma}_1^r \circ \cdots \circ \boldsymbol{\sigma}_D^r \circ \mathbf{v}^r$ and a vector $\mathbf{e}^1(\boldsymbol{\alpha}) \in \mathbb{R}^{n_1}$ is the $D+1$-dimensional rank-one tensor $\langle \boldsymbol{\sigma}_1^r, \mathbf{e}\rangle (\mathbf{u}^r \circ \boldsymbol{\sigma}_2^r \circ \cdots \circ \boldsymbol{\sigma}_D^r \circ \mathbf{v}^r)$. Proceeding with this computation for other modes, in the decomposition (3.17) we find that the definition (3.15) yields representation of $\widetilde{\Phi}_e(\boldsymbol{\alpha})$ as the sum of rank one matrices for any $\boldsymbol{\alpha} \in \mathcal{A}$:

$$\widetilde{\Phi}_e(\boldsymbol{\alpha}) = \sum_{r=1}^{R} s_r \mathbf{u}^r \circ \mathbf{v}^r \in \mathbb{R}^{M\times N}, \quad \text{with} \quad s_r = \prod_{i=1}^{D} \langle \boldsymbol{\sigma}_i^r, \mathbf{e}^i(\boldsymbol{\alpha})\rangle \in \mathbb{R}. \tag{3.18}$$

However, (3.18) is *not* the SVD of $\widetilde{\Phi}_e(\boldsymbol{\alpha})$, since vectors $\mathbf{u}^r$ (and $\mathbf{v}^r$) are not necessarily orthogonal. To avoid computing $\widetilde{\Phi}_e(\boldsymbol{\alpha})$ and its SVD online, the following preparatory offline step is required. Organize the vectors $\mathbf{u}^r$ and $\mathbf{v}^r$ from (3.17) into matrices

$$\widehat{\mathrm{U}} = [\mathbf{u}^1,\dots,\mathbf{u}^R] \in \mathbb{R}^{M\times R}, \quad \widehat{\mathrm{V}} = [\mathbf{v}^1,\dots,\mathbf{v}^R] \in \mathbb{R}^{N\times R}, \tag{3.19}$$

and compute the thin QR factorizations

$$\widehat{\mathrm{U}} = \mathrm{U}\mathrm{R}_U, \quad \widehat{\mathrm{V}} = \mathrm{V}\mathrm{R}_V, \tag{3.20}$$

if $R > N$ let further $\mathrm{R}_V = \widehat{\mathrm{V}}$ and $\mathrm{V} = \mathrm{I}$. *The columns of* U *form an orthogonal basis in the universal space* $\widetilde{V}$. Matrix U is stored offline (V is not used and can be dropped), while low-dimensional matrices R_U and R_V together with vectors $\boldsymbol{\sigma}_i^r$ form the *online* part of $\widetilde{\boldsymbol{\Phi}}$,

$$\mathrm{online}(\widetilde{\boldsymbol{\Phi}}) = \left\{\mathrm{R}_U, \mathrm{R}_V \in \mathbb{R}^{R\times R},\ \boldsymbol{\sigma}_i^r \in \mathbb{R}^{n_i},\ i=1,\dots,D\right\}, \tag{3.21}$$

which is transmitted to the online stage.

At the online stage and for any incoming $\boldsymbol{\alpha} \in \mathcal{A}$, we compute the SVD of the $R \times R$ *core* matrix

$$\mathrm{C}(\boldsymbol{\alpha}) = \mathrm{R}_U \mathrm{S}(\boldsymbol{\alpha}) \mathrm{R}_V^T, \tag{3.22}$$

where $\mathrm{S}(\boldsymbol{\alpha}) = \mathrm{diag}(s_1,\dots,s_R)$, with s_r from (3.18): $\mathrm{C}(\boldsymbol{\alpha}) = \mathrm{U}_c \Sigma_c \mathrm{V}_c^T$. Since

$$\widetilde{\Phi}_e(\boldsymbol{\alpha}) = \mathrm{U}\mathrm{C}(\boldsymbol{\alpha})\mathrm{V}^T = (\mathrm{U}\mathrm{U}_c)\, \Sigma_c\, (\mathrm{V}\mathrm{V}_c)^T \tag{3.23}$$

is the SVD of $\widetilde{\Phi}_e(\boldsymbol{\alpha})$, *the first n columns of* U_c, *denoted by* $\left\{\boldsymbol{\beta}_1(\boldsymbol{\alpha}),\dots,\boldsymbol{\beta}_n(\boldsymbol{\alpha})\right\}$, *are the coordinates of the local reduced basis in the universal space* $\widetilde{V}$. The parameter-specific basis in the physical space is then $\{\mathbf{z}_i(\boldsymbol{\alpha})\}_{i=1}^n$, with $\mathbf{z}_i(\boldsymbol{\alpha}) = \mathrm{U}\boldsymbol{\beta}_i(\boldsymbol{\alpha})$. Note that $\mathbf{z}_i(\boldsymbol{\alpha})$ are not actually computed.

Under certain assumptions on F, the dynamical system (2.1) is projected offline onto $\widetilde{V}$ and passed to the online stage, where for any $\boldsymbol{\alpha} \in \mathcal{A}$ it is further projected onto the local basis $\left\{\boldsymbol{\beta}_1(\boldsymbol{\alpha}),\dots,\boldsymbol{\beta}_n(\boldsymbol{\alpha})\right\}$. This avoids any online computations with high-dimensional objects used in high-fidelity simulations; see further discussion in Section 3.10.

We summarize the above in the following algorithm.

Algorithm 1 (*CP-TROM*).

- **Offline stage**.
  Input: snapshot tensor $\boldsymbol{\phi} \in \mathbb{R}^{M\times n_1\times\ldots\times n_D\times N}$, target canonical rank R;
  Output: CP decomposition factors, universal basis matrix $\mathrm{U} \in \mathbb{R}^{M\times R}$, and upper triangular matrices R_U, $\mathrm{R}_V \in \mathbb{R}^{R\times R}$;
  Compute:
  1. Use ALS algorithm to minimize $\|\boldsymbol{\phi}-\widetilde{\boldsymbol{\phi}}\|_F$ to find CP decomposition factors $\mathbf{u}^r$, $\boldsymbol{\sigma}_i^r$, and $\mathbf{v}^r$ of $\widetilde{\boldsymbol{\phi}}$ satisfying (3.17);
  2. Assemble matrices $\widehat{\mathrm{V}}$, $\widehat{\mathrm{U}}$ as in (3.19) and compute their thin QR factorizations (3.20) to obtain R_U, R_V and U.
- **Online stage**.
  Input: online($\widetilde{\Phi}$) as defined in (3.21), reduced space dimension $n \le R$, and parameter vector $\boldsymbol{\alpha} \in \mathcal{A}$;
  Output: Coordinates of the reduced basis in $\widetilde{V}$: $\{\boldsymbol{\beta}_i(\boldsymbol{\alpha})\}_{i=1}^n \subset \mathbb{R}^R$;
  Compute:
  1. Use (3.22) to assemble the core matrix $\mathrm{C}(\boldsymbol{\alpha})$;
  2. Compute the SVD of the core matrix $\mathrm{C}(\boldsymbol{\alpha}) = \mathrm{U}_c\Sigma_c\mathrm{V}_c^T$, with $\mathrm{U}_c = [\widetilde{\mathbf{u}}_1, \widetilde{\mathbf{u}}_2, \ldots, \widetilde{\mathbf{u}}_R]$;
  3. Set $\boldsymbol{\beta}_i(\boldsymbol{\alpha}) = \widetilde{\mathbf{u}}_i$, $i = 1, \ldots, n$.

Note that we do not have direct control over ALS algorithm to enforce a priori CP decomposition accuracy $\|\boldsymbol{\Phi} - \widetilde{\boldsymbol{\Phi}}\|_F < \widetilde{\varepsilon}\|\boldsymbol{\Phi}\|_F$. One option is to rerun the offline stage for different trial values of R. Given that the offline stage is computationally expensive, this may become prohibitive in cases where a desired accuracy $\widetilde{\varepsilon}$ must be strictly enforced. The other two variants of TROM presented below are free from this limitation.

3.6. HOSVD-TROM

As we already mentioned, truncated variant of CP decomposition is not known to satisfy any simple minimization property (unlike the SVD decomposition for matrices). A classical tensor decomposition, known to deliver a (quasi)-minimization property, is the so-called higher order SVD (HOSVD) [41]. In HOSVD-TROM variant we approximate the snapshot tensor with a Tucker tensor [20,42] $\widetilde{\Phi}$ of the form

$$\boldsymbol{\Phi} \approx \widetilde{\boldsymbol{\Phi}} = \sum_{j=1}^{\widetilde{M}}\sum_{q_1=1}^{\widetilde{n}_1}\cdots\sum_{q_D=1}^{\widetilde{n}_D}\sum_{k=1}^{\widetilde{N}}(\mathbf{C})_{j,q_1,\ldots,q_D,k}\mathbf{u}^j \circ \boldsymbol{\sigma}_1^{q_1} \circ\cdots\circ \boldsymbol{\sigma}_D^{q_D} \circ \mathbf{v}^k, \tag{3.24}$$

with $\mathbf{u}^j \in \mathbb{R}^M$, $\boldsymbol{\sigma}_i^{q_i} \in \mathbb{R}^{n_i}$, and $\mathbf{v}^k \in \mathbb{R}^N$. The numbers $\widetilde{M}$, $\widetilde{n}_1$, …, $\widetilde{n}_D$ and $\widetilde{N}$ are referred to as Tucker ranks of $\widetilde{\boldsymbol{\Phi}}$. The HOSVD delivers an efficient compression of the snapshot tensor, provided the size of the *core tensor* $\mathbf{C} \in \mathbb{R}^{\widetilde{M}\times\widetilde{n}_1\times\cdots\times\widetilde{n}_D\times\widetilde{N}}$ is (much) smaller than the size of $\boldsymbol{\Phi}$.

In what follows, it is helpful to organize the column vectors from (3.24) into matrices

$$\begin{aligned} &\mathrm{U} = [\mathbf{u}^1, \ldots, \mathbf{u}^{\widetilde{M}}] \in \mathbb{R}^{M\times\widetilde{M}}, \quad \mathrm{V} = [\mathbf{v}^1, \ldots, \mathbf{v}^{\widetilde{N}}] \in \mathbb{R}^{N\times\widetilde{N}}, \\ &\mathrm{S}_i = [\boldsymbol{\sigma}_i^1, \ldots, \boldsymbol{\sigma}_i^{\widetilde{n}_i}]^T \in \mathbb{R}^{\widetilde{n}_i\times n_i}, \quad i = 1, \ldots, D. \end{aligned} \tag{3.25}$$

In contrast with CP decomposition, HOSVD computes vectors $\mathbf{u}^j$, $j = 1, \ldots, \widetilde{M}$, and $\mathbf{v}^k$, $k = 1, \ldots, \widetilde{N}$, that are orthonormal. Therefore, *the columns of* U *form an orthogonal basis in the universal reduced space* $\widetilde{V}$. The dimension of this space is defined by the first Tucker rank, $\dim(\widetilde{V}) = \widetilde{M}$. The information about $\widetilde{\boldsymbol{\Phi}}$ to be transmitted to the online stage includes matrices S_i and the core tensor $\mathbf{C}$. Explicitly,

$$\mathrm{online}(\widetilde{\boldsymbol{\Phi}}) = \left\{\mathbf{C} \in \mathbb{R}^{\widetilde{M}\times\widetilde{n}_1\times\cdots\times\widetilde{n}_D\times\widetilde{N}},\ \mathrm{S}_i \in \mathbb{R}^{n_i\times\widetilde{n}_i},\ i = 1, \ldots, D\right\}. \tag{3.26}$$

To find coordinates of the local basis for $\boldsymbol{\alpha} \in \mathcal{A}$, define the $\boldsymbol{\alpha}$-specific *core matrix* $\mathrm{C}_e(\boldsymbol{\alpha})$ as

$$\mathrm{C}_e(\boldsymbol{\alpha}) = \mathbf{C} \times_2 \left(\mathrm{S}_1\mathbf{e}^1(\boldsymbol{\alpha})\right) \times_3 \left(\mathrm{S}_2\mathbf{e}^2(\boldsymbol{\alpha})\right) \ldots \times_{D+1} \left(\mathrm{S}_D\mathbf{e}^D(\boldsymbol{\alpha})\right) \in \mathbb{R}^{\widetilde{M}\times\widetilde{N}}. \tag{3.27}$$

Using the definition of k-mode product, (3.15) and (3.24), one computes

$$\begin{aligned}\widetilde{\Phi}_e(\boldsymbol{\alpha}) &= \sum_{j=1}^{\widetilde{M}}\sum_{q_1=1}^{\widetilde{n}_1}\cdots\sum_{q_D=1}^{\widetilde{n}_D}\sum_{k=1}^{\widetilde{N}}(\mathbf{C})_{j,q_1,\ldots,q_D,k}\langle\boldsymbol{\sigma}_1^{q_1}, \mathbf{e}^1(\alpha)\rangle\cdot\ldots\cdot\langle\boldsymbol{\sigma}_D^{q_D}, \mathbf{e}^D(\alpha)\rangle\,\mathbf{u}^j\circ\mathbf{v}^k\\ &= \sum_{j=1}^{\widetilde{M}}\sum_{q_1=1}^{\widetilde{n}_1}\cdots\sum_{q_D=1}^{\widetilde{n}_D}\sum_{k=1}^{\widetilde{N}}(\mathbf{C})_{j,q_1,\ldots,q_D,k}\left(\mathrm{S}_1\mathbf{e}^1(\boldsymbol{\alpha})\right)_{q_1}\cdot\ldots\cdot\left(\mathrm{S}_D\mathbf{e}^D(\boldsymbol{\alpha})\right)_{q_D}\mathbf{u}^j\circ\mathbf{v}^k\\ &= \sum_{j=1}^{\widetilde{M}}\sum_{k=1}^{\widetilde{N}}\left(\mathbf{C}\times_2\left(\mathrm{S}_1\mathbf{e}^1(\boldsymbol{\alpha})\right)\times_3\left(\mathrm{S}_2\mathbf{e}^2(\boldsymbol{\alpha})\right)\ldots\times_{D+1}\left(\mathrm{S}_D\mathbf{e}^D(\boldsymbol{\alpha})\right)\right)_{jk}\mathbf{u}^j\circ\mathbf{v}^k = \mathrm{U}\mathrm{C}_e(\boldsymbol{\alpha})\mathrm{V}^T\end{aligned}$$

Consider the thin SVD of the core matrix

$$\mathrm{C}_e(\boldsymbol{\alpha}) = \mathrm{U}_c\Sigma_c\mathrm{V}_c^T. \tag{3.28}$$

Combining this with the representation above we get

$$\widetilde{\Phi}_e(\boldsymbol{\alpha}) = (\mathrm{U}\mathrm{U}_c)\,\Sigma_c\,(\mathrm{V}\mathrm{V}_c)^T\,, \tag{3.29}$$

which is the thin SVD of $\widetilde{\Phi}_e(\boldsymbol{\alpha})$ since both matrices U and V are orthogonal. We conclude that *the coordinates* $\left\{\boldsymbol{\beta}_1(\boldsymbol{\alpha}),\ldots,\boldsymbol{\beta}_n(\boldsymbol{\alpha})\right\}$ *of the local reduced basis in the universal space* $\widetilde{V}$ *are the first n columns of* U_c *from* (3.28). The parameter-specific basis is then $\{\mathbf{z}_i(\boldsymbol{\alpha})\}_{i=1}^n$, with $\mathbf{z}_i(\boldsymbol{\alpha}) = \mathrm{U}\boldsymbol{\beta}_i(\boldsymbol{\alpha})$ (not actually computed at the online stage).

To compute the low-rank HOSVD approximation (3.24) we employ the standard algorithm [41] based on repeated computations of truncated SVD for unfolded matrices. In particular, one may compute $\widetilde{\boldsymbol{\Phi}}$ with either prescribed Tucker ranks or prescribed accuracy $\widetilde{\varepsilon}$. Moreover, for fixed Tucker ranks one can show that the recovered $\widetilde{\boldsymbol{\Phi}}$ satisfies a quasi-minimization property [41] of the form

$$\|\boldsymbol{\Phi} - \widetilde{\boldsymbol{\Phi}}\| \le \sqrt{D+2}\|\boldsymbol{\Phi} - \boldsymbol{\Phi}^{\mathrm{opt}}\| \quad \text{and} \quad \|\boldsymbol{\Phi} - \widetilde{\boldsymbol{\Phi}}\| \le \left(\sum_{i=1}^{D+1}\Delta_i^2\right)^{\frac{1}{2}}, \tag{3.30}$$

where $\boldsymbol{\Phi}^{\mathrm{opt}}$ is the best approximation to $\boldsymbol{\Phi}$ among all Tucker tensors of the given rank (such approximation always exists), and Δ_i measures truncated SVD error on the ith step of the HOSVD. We summarize the above in the following algorithm.

Algorithm 2 (*HOSVD-TROM*).

- **Offline stage.**
  Input: snapshot tensor $\boldsymbol{\phi} \in \mathbb{R}^{M\times n_1\times\ldots\times n_D\times N}$ and target accuracy $\widetilde{\varepsilon}$;
  Output: Compressed tensor ranks, HOSVD decomposition matrices as in (3.25), and core tensor $\mathbf{C}$;
  Compute: Use algorithm [41] with prescribed accuracy $\widetilde{\varepsilon}$ to compute HOSVD decomposition matrices and the core tensor.
- **Online stage.**
  Input: online($\widetilde{\boldsymbol{\phi}}$) as defined in (3.26), reduced space dimension $n \le \min\{\widetilde{M}, \widetilde{N}\}$, and parameter vector $\boldsymbol{\alpha} \in \mathcal{A}$;
  Output: Coordinates of the reduced basis in $\widetilde{V}$: $\{\boldsymbol{\beta}_i(\boldsymbol{\alpha})\}_{i=1}^n \subset \mathbb{R}^{\widetilde{M}}$;
  Compute:

  1. Use the core tensor $\mathbf{C}$ and matrices S_i, $i = 1,\ldots,D$, to assemble the core matrix $\mathrm{C}_e(\boldsymbol{\alpha}) \in \mathbb{R}^{\widetilde{M}\times\widetilde{N}}$ as in (3.27);
  2. Compute the SVD of the core matrix $\mathrm{C}_e(\boldsymbol{\alpha}) = \mathrm{U}_c\Sigma_c\mathrm{V}_c^T$ with $\mathrm{U}_c = [\widetilde{\mathbf{u}}_1,\ldots,\widetilde{\mathbf{u}}_{\widetilde{M}}]$;
  3. Set $\boldsymbol{\beta}_i(\boldsymbol{\alpha}) = \widetilde{\mathbf{u}}_i$, $i = 1,\ldots,n$.

3.7. TT-TROM

A third low-rank tensor decomposition of interest is the *Tensor Train* (TT) decomposition [43]. In TT-TROM we seek to approximate the snapshot tensor with $\widetilde{\boldsymbol{\Phi}}$ in the TT-format, namely

$$\boldsymbol{\Phi} \approx \widetilde{\boldsymbol{\Phi}} = \sum_{j_1=1}^{\widetilde{r}_1} \cdots \sum_{j_{D+1}=1}^{\widetilde{r}_{D+1}} \mathbf{u}^{j_1} \circ \boldsymbol{\sigma}_1^{j_1,j_2} \circ \cdots \circ \boldsymbol{\sigma}_D^{j_D,j_{D+1}} \circ \mathbf{v}^{j_{D+1}}, \tag{3.31}$$

with $\mathbf{u}^{j_1} \in \mathbb{R}^M$, $\boldsymbol{\sigma}_i^{j_i,j_{i+1}} \in \mathbb{R}^{n_i}$, and $\mathbf{v}^{j_{D+1}} \in \mathbb{R}^N$, where the positive integers $\widetilde{r}_i$ are referred to as the *compression ranks* (or TT-ranks) of the decomposition. For higher order tensors the TT format is in general more efficient compared to HOSVD. This may be beneficial for large D, the dimension of parameter space. In [43,44] a stable algorithm for finding $\widetilde{\boldsymbol{\Phi}}$ based on truncated SVD for a sequence of unfolding matrices was introduced and the optimality property similar to (3.30) was proved.

Once an optimal TT approximation (3.31) is computed, we organize the vectors and matrices from (3.31) into matrices

$$\mathrm{U} = [\mathbf{u}^1, \ldots, \mathbf{u}^{\widetilde{r}_1}] \in \mathbb{R}^{M \times \widetilde{r}_1}, \quad \mathrm{V} = [\mathbf{v}^1, \ldots, \mathbf{v}^{\widetilde{r}_{D+1}}] \in \mathbb{R}^{N \times \widetilde{r}_{D+1}}, \tag{3.32}$$

and third order tensors $\mathbf{S}_i \in \mathbb{R}^{\widetilde{r}_i \times n_i \times \widetilde{r}_{i+1}}$, defined entry-wise as

$$(\mathbf{S}_i)_{jkq} = (\boldsymbol{\sigma}_i^{jq})_k, \quad j = 1, \ldots, \widetilde{r}_i, \quad k = 1, \ldots, n_i, \quad q = 1, \ldots, \widetilde{r}_{i+1}, \tag{3.33}$$

for all $i = 1, \ldots, D$. Note that matrix U is orthogonal and so *its columns provide an orthogonal basis in the universal space* $\widetilde{V}$. The dimension of $\widetilde{V}$ is defined by the first TT-rank, $\dim(\widetilde{V}) = \widetilde{r}_1$.

While U is an orthogonal matrix, the columns of V are orthogonal, but not necessarily orthonormal. Thus, we introduce a diagonal scaling matrix

$$\mathrm{W}_c = \operatorname{diag}\left(\|\mathbf{v}^1\|, \ldots, \|\mathbf{v}^{\widetilde{r}_{D+1}}\|\right) \in \mathbb{R}^{\widetilde{r}_{D+1} \times \widetilde{r}_{D+1}}. \tag{3.34}$$

The essential information about $\widetilde{\boldsymbol{\Phi}}$ to be transmitted to the online phase includes $\mathbf{S}_i$ tensors and the scaling factors:

$$\operatorname{online}(\widetilde{\boldsymbol{\Phi}}) = \left\{\mathbf{S}_i \in \mathbb{R}^{\widetilde{r}_i \times n_i \times \widetilde{r}_{i+1}},\ i = 1, \ldots, D,\ \mathrm{W}_c \in \mathbb{R}^{\widetilde{r}_{D+1} \times \widetilde{r}_{D+1}}\right\}. \tag{3.35}$$

To find the coordinates of the local basis, we define the parameter-specific *core matrix* $\mathrm{C}_e(\boldsymbol{\alpha}) \in \mathbb{R}^{\widetilde{r}_1 \times \widetilde{r}_{D+1}}$ as the product

$$\mathrm{C}_e(\boldsymbol{\alpha}) = \prod_{i=1}^{D} \left(\mathbf{S}_i \times_2 \mathbf{e}^i(\boldsymbol{\alpha})\right). \tag{3.36}$$

Using the definition of k-mode product, (3.15) and (3.31), one computes

$$\begin{aligned}
\widetilde{\Phi}_e(\boldsymbol{\alpha}) &= \sum_{j_1=1}^{\widetilde{r}_1} \cdots \sum_{j_{D+1}=1}^{\widetilde{r}_{D+1}} \langle \boldsymbol{\sigma}_1^{j_1,j_2}, \mathbf{e}_1(\boldsymbol{\alpha})\rangle \cdot \ldots \cdot \langle \boldsymbol{\sigma}_D^{j_D,j_{D+1}}, \mathbf{e}_D(\boldsymbol{\alpha})\rangle\, \mathbf{u}^{j_1} \circ \mathbf{v}^{j_{D+1}} \\
&= \sum_{j_1=1}^{\widetilde{r}_1} \cdots \sum_{j_{D+1}=1}^{\widetilde{r}_{D+1}} \left(\mathbf{S}_1 \times_2 \mathbf{e}^1(\boldsymbol{\alpha})\right)_{j_1,j_2} \cdot \ldots \cdot \left(\mathbf{S}_D \times_{D+1} \mathbf{e}^D(\boldsymbol{\alpha})\right)_{j_D,j_{D+1}} \mathbf{u}^{j_1} \circ \mathbf{v}^{j_{D+1}} \\
&= \sum_{j_1=1}^{\widetilde{r}_1} \sum_{j_{D+1}=1}^{\widetilde{r}_{D+1}} \left(\prod_{i=1}^{D} \left(\mathbf{S}_i \times_2 \mathbf{e}^i(\boldsymbol{\alpha})\right)\right)_{j_1 j_{D+1}} \mathbf{u}^{j_1} \circ \mathbf{v}^{j_{D+1}} = \mathrm{U}\mathrm{C}_e(\boldsymbol{\alpha})\mathrm{V}^T.
\end{aligned}$$

Consider the SVD of the rescaled core matrix:

$$\mathrm{C}_e(\boldsymbol{\alpha})\mathrm{W}_c = \mathrm{U}_c \Sigma_c \mathrm{V}_c^T. \tag{3.37}$$

Using this and the above representation of $\widetilde{\Phi}_e(\boldsymbol{\alpha})$ we compute

$$\widetilde{\Phi}_e(\boldsymbol{\alpha}) = \mathrm{U}\mathrm{C}_e(\boldsymbol{\alpha})\mathrm{W}_c\mathrm{W}_c^{-1}\mathrm{V}^T = (\mathrm{U}\mathrm{U}_c)\, \Sigma_c \left(\mathrm{V}\mathrm{W}_c^{-1}\mathrm{V}_c\right)^T. \tag{3.38}$$

The right-hand side of (3.38) is the thin SVD of $\widetilde{\Phi}_e(\boldsymbol{\alpha})$, since matrices U, U_c, $\mathrm{V}\mathrm{W}_c^{-1}$, and V_c are all orthogonal. We conclude that *the coordinates* $\left\{\boldsymbol{\beta}_1(\boldsymbol{\alpha}),\ldots,\boldsymbol{\beta}_n(\boldsymbol{\alpha})\right\}$ *of the local reduced basis in the universal space* $\widetilde{V}$ *are the first* n *columns of* U_c. The parameter-specific basis is then $\{\mathbf{z}_i(\boldsymbol{\alpha})\}_{i=1}^n$, with $\mathbf{z}_i(\boldsymbol{\alpha}) = \mathrm{U}\boldsymbol{\beta}_i(\boldsymbol{\alpha})$ (not actually computed at the online stage).

We summarize the above in the following algorithm.

Algorithm 3 (*TT-TROM*)**.**

- **Offline stage**.
  Input: snapshot tensor $\boldsymbol{\phi} \in \mathbb{R}^{M\times n_1\times\ldots\times n_D\times N}$ and target accuracy $\widetilde{\varepsilon}$;
  Output: Compression ranks, TT decomposition matrices and third order tensors as in (3.32)–(3.33);
  Compute: Use algorithm from [43] with prescribed accuracy $\widetilde{\varepsilon}$ to compute TT decomposition (3.31).
- **Online stage**.
  Input: online($\widetilde{\boldsymbol{\phi}}$) as defined in (3.35), reduced space dimension $n \le \min\{\widetilde{r}_1, \widetilde{r}_{D+1}\}$, and parameter vector $\boldsymbol{\alpha} \in \mathcal{A}$;
  Output: Coordinates of the reduced basis in $\widetilde{V}$: $\{\boldsymbol{\beta}_i(\boldsymbol{\alpha})\}_{i=1}^n \subset \mathbb{R}^{\widetilde{r}_1}$;
  Compute:

  1. Use tensors $\mathbf{S}_i$ to assemble the core matrix $\mathrm{C}_e(\boldsymbol{\alpha}) \in \mathbb{R}^{\widetilde{r}_1\times\widetilde{r}_{D+1}}$ as in (3.36);
  2. Compute the SVD of the scaled core matrix $\mathrm{C}_e(\boldsymbol{\alpha})\mathrm{W}_c = \mathrm{U}_c\Sigma_c\mathrm{V}_c^T$ with $\mathrm{U}_c = [\widetilde{\mathbf{u}}_1,\ldots,\widetilde{\mathbf{u}}_{\widetilde{r}_1}]$;
  3. Set $\boldsymbol{\beta}_i(\boldsymbol{\alpha}) = \widetilde{\mathbf{u}}_i$, $i = 1,\ldots,n$.

3.8. General parameter sampling

Grid-based sampling of parameter space can be computationally expensive or not applicable if the set of admissible parameters $\mathcal{A}$ is not a box (or an image of a box) in Euclidean space. However, interpolatory TROMs introduced above can be extended to accommodate a more general sampling set $\widehat{\mathcal{A}}$. If $\mathcal{A}$ does have the Cartesian structure (a box or an image of a box), then one way to reduce offline computational costs is to compute the snapshots for only a few parameter values from a Cartesian grid $\widehat{\mathcal{A}} \subset \mathcal{A}$. To recover the missing entries of the full snapshot tensor $\boldsymbol{\Phi}$, one may use a low-rank tensor completion method, e.g., one of those studied in [45–49]. The low-rank completion can be performed for any of the three compressed tensor formats considered above. We shall investigate this option elsewhere. In this paper, we consider another (more general) approach.

With a slight abuse of notation, let $\widehat{\mathcal{A}} = \{\widehat{\boldsymbol{\alpha}}_1,\ldots,\widehat{\boldsymbol{\alpha}}_K\} \subset \mathcal{A}$ be a set of sampled parameter values. We assume that $\widehat{\mathcal{A}}$ is a frame in $\mathbb{R}^D$ and so $K \ge D$. Note that K does not obey (3.3) for a general sampling. Given an out-of-sample vector of parameters $\boldsymbol{\alpha} \in \mathcal{A}$, let

$$\mathbf{e} : \boldsymbol{\alpha} \to \mathbb{R}^K \tag{3.39}$$

be the representation of $\boldsymbol{\alpha}$ in $\widehat{\mathcal{A}}$, i.e.,

$$\boldsymbol{\alpha} = \sum_{j=1}^K a_j\widehat{\boldsymbol{\alpha}}_j, \quad \mathbf{e}(\boldsymbol{\alpha}) = (a_1,\ldots,a_K)^T, \tag{3.40}$$

with an additional constraint enforcing uniqueness of the representation.

Similarly to (3.13) for the Cartesian grid case, we assume that for a smooth function $g : \mathcal{A} \to \mathbb{R}$ it holds

$$g(\boldsymbol{\alpha}) \approx \sum_{j=1}^K a_j g(\widehat{\boldsymbol{\alpha}}_j). \tag{3.41}$$

In Section 5.1 we describe one particular choice of $\mathbf{e}(\boldsymbol{\alpha})$ that is used in all numerical experiments reported in Section 5.

To assemble the snapshot tensor, for each $\widehat{\boldsymbol{\alpha}}_j \in \widehat{\mathcal{A}}$, $j = 1,\ldots,K$, collect the snapshot vectors $\mathbf{u}(t_k,\widehat{\boldsymbol{\alpha}}_j) = \left(u_1(t_k,\widehat{\boldsymbol{\alpha}}_j),\ldots,u_M(t_k,\widehat{\boldsymbol{\alpha}}_j)\right)^T$, $k = 1,\ldots,N$, and arrange them in a third order tensor $\boldsymbol{\Phi} \in \mathbb{R}^{M\times K\times N}$ with entries

$$(\boldsymbol{\Phi})_{ijk} = u_i(t_k,\widehat{\boldsymbol{\alpha}}_j), \quad i = 1,\ldots,M, \quad j = 1,\ldots,K, \quad k = 1,\ldots,N. \tag{3.42}$$

Then, for any $\boldsymbol{\alpha} \in \mathcal{A}$, the parameter-specific reduced basis is defined as the first n left singular vectors of

$$\widetilde{\Phi}_e(\boldsymbol{\alpha}) = \widetilde{\boldsymbol{\Phi}} \times_2 \mathbf{e}(\boldsymbol{\alpha}), \tag{3.43}$$

where $\widetilde{\boldsymbol{\Phi}}$ is a low rank approximation of the snapshot tensor $\boldsymbol{\Phi}$ with entries (3.42). The same three compressed tensor formats considered above (CP, HOSVD and TT) can be used for $\widetilde{\boldsymbol{\Phi}}$, with TT format being inferior to HOSVD (for 3D tensors both HOSVD- and TT-decomposition are Tucker tensors). An orthogonal basis in the universal space of $\widetilde{\boldsymbol{\Phi}}$ and coordinates of a local parameter-specific basis in it are computed similarly to the Cartesian grid sampling cases considered previously in Sections 3.5–3.7. The only difference is that all the calculations therein are performed setting $D = 1$ and replacing $\mathbf{e}^1(\boldsymbol{\alpha})$ with $\mathbf{e}(\boldsymbol{\alpha})$. This includes the optimality result (3.30), where the factor $\sqrt{D+2}$ becomes $\sqrt{3}$.

3.9. Complexity and compression analysis

Projection-based parametric ROM framework consists in general of the following steps.

(i) High-fidelity simulations of (2.1) to generate the snapshot tensor $\boldsymbol{\Phi}$;
(ii) **Offline stage**: computing the compressed approximation $\widetilde{\boldsymbol{\Phi}}$ to $\boldsymbol{\Phi}$ in one of low-rank tensor formats;
(iii) Passing the online($\widetilde{\boldsymbol{\Phi}}$) part of the compressed tensor to the online stage;
(iv) **Online stage**: using online($\widetilde{\boldsymbol{\Phi}}$) to compute the coordinates of the parameter-specific reduced basis for an input $\boldsymbol{\alpha}$;
(v) Solving (2.1) projected onto the reduced space.

Since steps (i) and (v) are common for all projection-based ROM approaches, we focus below on the computational and storage/transmission costs invoked in steps (ii)–(iv). The necessary details on step (v) are included in Section 3.10.

First, we discuss briefly the computational costs at the more expensive offline stage. For CP-TROM, the standard algorithm for finding $\widetilde{\boldsymbol{\Phi}}$ in CP format (3.17) is the ALS method [40] which for a *given* CP rank R iteratively fits a rank R tensor $\widetilde{\Phi}$ by solving on each iteration $D+2$ least squares problems for the factors $\mathbf{u}^r$, $\boldsymbol{\sigma}_i^r$, $i = 1, \dots, D$, and $\mathbf{v}^r$, $r = 1, \dots, R$. While straightforward to implement, the method is sensitive to the choice of initial guess and may converge slowly. We refer the reader to [20] for a guidance on the literature on improving the efficiency of ALS and possible alternatives. On the other hand, computing $\widetilde{\boldsymbol{\Phi}}$ in either HOSVD or TT formats relies on finding truncated SVDs for matrix unfoldings of $\boldsymbol{\Phi}$ [41,43]. Therefore, the computational complexity and cost of step (ii) for HOSVD- and TT-TROM is essentially the same as that of standard POD-ROM.

Second, to measure the amount of information transmitted to the online stage at step (iii), we introduce the *compression factor* CF, defined as

$$\mathrm{CF} = \frac{\#(\boldsymbol{\Phi})}{\#(\mathrm{online}(\widetilde{\boldsymbol{\Phi}}))}, \tag{3.44}$$

where we denote by $\#(\boldsymbol{\Psi})$ the number of floating point numbers needed to store a tensor $\boldsymbol{\Psi}$. Specifically, $\#(\boldsymbol{\Phi}) = MKN$ is simply the total number of entries in $\boldsymbol{\Phi}$, while $\#(\mathrm{online}(\widetilde{\boldsymbol{\Phi}}))$ is the number of entries needed to store all the factors passed to the online stage, as defined in (3.21), (3.26) and (3.35) for CP-, HOSVD- and TT-TROMs, which we summarize in Table 3.1.

Table 3.1 shows that the compression factor is largely determined by the compression ranks. In turn, the ranks depend on $\widetilde{\varepsilon}$ and variability of observed states.

Third, the computational complexity of finding $\boldsymbol{\alpha}$-specific reduced basis in step (iv) is determined by the interpolation procedure and the computation of first n left singular vectors of the core matrix. Since vectors $\mathbf{e}^i(\boldsymbol{\alpha})$ contain very few non-zero entries, e.g., $p = 2$ or 3 of them for the Cartesian sampling, the number of operations for computing core matrices $\mathrm{C}(\boldsymbol{\alpha})$ for CP-, HOSVD- and TT-TROM is

$$O\left(R^2\right), \quad O\Big(\widetilde{M}\widetilde{N}\prod_{i=1}^{D}\widetilde{n}_i\Big), \quad \text{and} \quad O\Big(\sum_{i=2}^{D}\widetilde{r}_{i-1}\widetilde{r}_i\widetilde{r}_{i+1}\Big), \tag{3.45}$$

respectively. CP-, HOSVD- and TT-TROM algorithms proceed to compute the SVD of small core matrices of sizes $R \times R$, $\widetilde{M} \times \widetilde{N}$ and $\widetilde{r}_1 \times \widetilde{r}_{D+1}$, respectively. If a reduced basis in the physical space is desired, then one finds

Table 3.1
Number of entries needed to store online($\widetilde{\boldsymbol{\Phi}}$) for CP, HOSVD and TT formats.

Format	#(online($\widetilde{\boldsymbol{\Phi}}$))	
	Cartesian grid-based	General
CP	$R\left(\sum_{i=1}^{D} n_i + R + 1\right)$	$R\left(K + R + 1\right)$
HOSVD	$\widetilde{N}\widetilde{M}\prod_{i=1}^{D}\widetilde{n}_i + \sum_{i=1}^{D}\widetilde{n}_i n_i$	$\widetilde{N}\widetilde{M}\widetilde{n}_1 + \widetilde{n}_1 K$
TT	$\widetilde{r}_{D+1} + \sum_{i=1}^{D}\widetilde{r}_i n_i \widetilde{r}_{i+1}$	$\widetilde{r}_2 + \widetilde{r}_1 K \widetilde{r}_2$

its vectors as linear combinations of columns of U, which requires $O(MRn)$, $O(M\widetilde{M}n)$ or $O(M\widetilde{r}_1 n)$ operations for CP-, HOSVD- or TT-TROM, respectively. Section 3.10 discusses how these costs can be avoided at the online phase. We note that for a fixed compression accuracy $\widetilde{\epsilon}$, it is often observed in practice that the corresponding ranks of HOSVD and TT formats satisfy $\widetilde{M} \simeq \widetilde{r}_1$, $\widetilde{N} \simeq \widetilde{r}_{D+1}$.

In summary, the computational costs of the offline stage for TROMs are comparable to those of POD-ROM for a multi-parameter problem. At the online stage complexity of all preparatory steps depends only on compressed tensor ranks rather than the size of the snapshot tensor $\boldsymbol{\Phi}$. The amount of information transmitted from offline to online stages is determined by the compressed tensor ranks, as should be clear from Table 3.1.

3.10. TROM evaluation

Besides finding a suitable reduced basis, a fast evaluation of the reduced model for any incoming $\boldsymbol{\alpha} \in \mathcal{A}$ is required for a reduced modeling scheme to be effective. Efficient implementation of a projected parametric model is a well-known challenge that have been addressed in the literature with various approaches; see, e.g., [3,4,11,50–54]. The tensorial approach presented here does not directly contribute to resolving this issue, but it does not make it harder either and so techniques known from the literature can be adapted in the TROM framework.

For example, assume that $F(t, \mathbf{u}, \boldsymbol{\alpha})$ from (2.1) has an affine dependence on parameters and linear dependence on $\mathbf{u}$:

$$F(t, \mathbf{u}, \boldsymbol{\alpha}) = \sum_{i=1}^{P} f_i(\boldsymbol{\alpha})\mathrm{A}_i\mathbf{u},$$

with some $f_i : \mathcal{A} \to \mathbb{R}$ and parameter-independent $\mathrm{A}_i \in \mathbb{R}^{M\times M}$. We assume that P is not too large, at least independent of other dimensions. Then the *offline* stage of model reduction consists of projecting matrices onto the universal space by computing $\widehat{\mathrm{A}}_i = \mathrm{U}^T\mathrm{A}_i\mathrm{U}$, where U is an orthogonal basis matrix for $\widetilde{V}$ provided by the tensor decompositions. The new matrices $\widehat{\mathrm{A}}_i$ have the reduced size $T_r \times T_r$, with $T_r \in \{R, \widetilde{M}, \widetilde{r}_1\}$ for CP-, HOSVD- and TT-TROMs, respectively.

For each of TROMs, denote by $\mathrm{U}_c(n)$ the matrix of the first n columns of U_c, left singular vectors of α-specific core matrices. During the *online* stage one solves the system projected further on the parameter-specific local basis:

$$\mathbf{v}_t = \sum_{i=1}^{P} f_i(\boldsymbol{\alpha})\mathrm{U}_c^T(n)\widehat{\mathrm{A}}_i\mathrm{U}_c(n)\mathbf{v},$$

where $\mathbf{v}(t)$ is the trajectory in a space spanned by the columns of $\mathrm{U}_c(n) \in \mathbb{R}^{T_r\times n}$, i.e., the corresponding physical states are given by $\mathbf{u}(t) = \mathrm{U}\mathrm{U}_c(n)\mathbf{v}(t)$. We see that online computations depend only on reduced dimensions (tensor ranks) and the small dimension n of parameter-specific basis. This observation can be extended to the case when F has a low order polynomial non-linearity with respect to $\mathbf{u}$. For example, quadratic nonlinear terms, as in Burgers or Navier–Stokes equations, can be evaluated in $O(T_r^2)$ operations on each time step given a vector $\mathbf{v}$ in the local reduced basis.

To evaluate more general nonlinear terms, one can use a hyper-reduction technique such as the discrete empirical interpolation method (DEIM) [11]. In this approach, the nonlinear term is approximated in a basis of its snapshots.

As an example, consider $F(t, \mathbf{u}, \boldsymbol{\alpha}) = \mathrm{A}\mathbf{u} + f(t, \mathbf{u}(t), \boldsymbol{\alpha})$, with $\mathrm{A}\mathbf{u}$ representing linear and $f(t, \mathbf{u}(t), \boldsymbol{\alpha})$ representing the non-linear part of F. Define the snapshots $\mathbf{f}_i = f(t_i, \mathbf{u}(t_i), \boldsymbol{\alpha}_i)$, $i = 1, \dots, N_{\text{DEIM}}$ for some "greedy" choice of parameters and time instances and high-fidelity solution $\mathbf{u}$. Denote by Q an orthogonal basis matrix for $\operatorname{span}\{\mathbf{f}_1, \dots, \mathbf{f}_{N_{\text{DEIM}}}\}$, then DEIM approximates

$$f(t, \mathbf{u}, \boldsymbol{\alpha}) \approx \mathrm{Q}(\mathrm{PQ})^{-1}\mathrm{P}f(t, \mathbf{u}, \boldsymbol{\alpha}),$$

where P^T is an $N_{\text{DEIM}} \times M$ "selection" matrix such that for any $f \in \mathbb{R}^M$, $\mathrm{P}f$ contains N_{DEIM} selected entries of f. A particular P corresponds to the choice of spatial interpolation nodes, cf. [11]. In TROM one may pre-compute $\widehat{\mathrm{Q}} = \mathrm{U}^T\mathrm{Q}$ and $\widehat{\mathrm{A}} = \mathrm{U}^T\mathrm{A}\mathrm{U}$ during the offline stage, then solve at the online stage

$$\mathbf{v}_t = \mathrm{U}_c^T(n)\widehat{\mathrm{A}}\mathrm{U}_c(n)\mathbf{v} + \mathrm{U}_c^T(n)\widehat{\mathrm{Q}}(\mathrm{PQ})^{-1}\mathrm{P}f(t, \mathrm{U}\mathrm{U}_c(n)\mathbf{v}, \boldsymbol{\alpha}),$$

with costs depending on compressed tensor ranks, n, and the dimension of DEIM space, but not on the dimensions of high-fidelity simulations. It is an interesting question, whether the tensor technique can be applied to make the DEIM space parameter-specific for more efficient reduce online computations. We plan to address this question elsewhere.

4. Prediction analysis

In this section we assess the prediction power of the reduced basis $\mathcal{Z}_n(\boldsymbol{\alpha}) = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$ consisting of the first n left singular vectors of $\widetilde{\Phi}_e(\boldsymbol{\alpha})$ from (3.15), for a parameter $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_D)^T \in \mathcal{A}$, not necessarily from a sampling set; i.e., $[\mathbf{z}_1, \dots, \mathbf{z}_n] = \mathrm{U}\mathrm{U}_c(n)$.

For the discussion below we also need the following notation. Given an $\boldsymbol{\alpha} \in \mathcal{A}$, we denote by $\boldsymbol{\psi}_i = \mathbf{u}(t_i, \boldsymbol{\alpha}) \in \mathbb{R}^M$, $i = 1, \dots, N$, the snapshots of a high-fidelity solution to (2.1) and let $\Psi(\boldsymbol{\alpha}) = [\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_N] \in \mathbb{R}^{M \times N}$ be the corresponding snapshot matrix. Note that in practice the snapshots for out-of-sample parameters are not available, so the matrix $\Psi(\boldsymbol{\alpha})$ should be treated as unknown.

We estimate the prediction power of $\mathcal{Z}_n(\boldsymbol{\alpha})$ in terms of the quantity

$$E_n(\boldsymbol{\alpha}) = \frac{1}{NM}\sum_{i=1}^{N}\left\| \boldsymbol{\psi}_i - \sum_{j=1}^{n}\langle \boldsymbol{\psi}_i, \mathbf{z}_j\rangle \mathbf{z}_j \right\|_{\ell^2}^2, \tag{4.1}$$

which measures how accurate the solution $\mathbf{u}(t, \boldsymbol{\alpha})$ at time instances t_i can be represented in the reduced basis for the arbitrary but fixed $\boldsymbol{\alpha} \in \mathcal{A}$. The scaling $1/(NM)$ accounts for the variation of dimensions N and M, which may correspond to the number of temporal and spatial degrees of freedom, respectively, if (2.1) comes from a discretization of a parabolic PDE defined in a spatial domain Ω. In this case and for uniform grids, the quantity in (4.1) is consistent with the $L^2(0, T, L^2(\Omega))$ norm.

Below we prove an estimate for $E_n(\boldsymbol{\alpha})$ in terms of $\widetilde{\varepsilon}$ from (3.5), the singular values of $\widetilde{\Phi}_e(\boldsymbol{\alpha})$ and interpolation properties of $\mathbf{e}^i(\boldsymbol{\alpha})$, $i = 1, \dots, D$. To make use of the latter, we introduce the following quantities related to the interpolation procedure. For Cartesian grid-based sampling we define the maximum grid step

$$\delta_i = \max_{1 \le j \le n_i - 1}\left|\widehat{\alpha}_j^i - \widehat{\alpha}_{j+1}^i\right|, \quad i = 1, \dots, D. \tag{4.2}$$

Relation (3.14) implies that the interpolation procedure (3.12)–(3.14) is of order p, i.e., for any sufficiently smooth $f : [\alpha_i^{\min}, \alpha_i^{\max}] \to \mathbb{R}$ it holds

$$\sup_{a \in [\alpha_i^{\min}, \alpha_i^{\max}]}\left| f(a) - \sum_{j=1}^{n_i} e_j^i(a\mathbf{e}_i)\, f(\widehat{\alpha}_i^j)\right| \le C_a \|f^{(p)}\|_{C([\alpha_i^{\min}, \alpha_i^{\max}])}\delta_i^p, \tag{4.3}$$

for $i = 1, \dots, D$, where $\mathbf{e}_i \in \mathbb{R}^{n_i}$ is the ith column of an $n_i \times n_i$ identity matrix. The constant C_a does not depend on f. We let $\delta^p = \sum_{i=1}^{D}\delta_i^p$ and also assume that the interpolation procedure is stable in the sense that

$$\left(\sum_{j=1}^{n_i}\left|e_j^i(a\mathbf{e}_i)\right|^2\right)^{\frac{1}{2}} \le C_e, \tag{4.4}$$

with some C_e independent of $a \in [\alpha_i^{\min}, \alpha_i^{\max}]$ and $i = 1, \dots, D$. For the example of linear interpolation with $p = 2$ and $\alpha_i^{\min}$, $\alpha_i^{\max}$ included among the grid nodes, bounds (4.3) and (4.4) hold with $C_a = \frac{1}{8}$ and $C_e = 1$.

To estimate $E_n(\boldsymbol{\alpha})$, consider the SVD of $\widetilde{\Phi}_e(\boldsymbol{\alpha}) \in \mathbb{R}^{M\times N}$ given by

$$\widetilde{\Phi}_e(\boldsymbol{\alpha}) = \widetilde{\mathrm{U}}\widetilde{\Sigma}\widetilde{\mathrm{V}}^T, \quad \text{with} \quad \widetilde{\Sigma} = \mathrm{diag}(\widetilde{\sigma}_1, \dots, \widetilde{\sigma}_N). \tag{4.5}$$

Then $\mathrm{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_n] \in \mathbb{R}^{M\times n}$ is build as the first n columns of $\widetilde{\mathrm{U}}$, the reduced basis vectors, i.e., $\mathrm{Z} = \mathrm{U}\mathrm{U}_c$. Then,

$$\begin{aligned}
E_n(\boldsymbol{\alpha}) &= \frac{1}{NM}\left\|(\mathrm{I} - \mathrm{Z}\mathrm{Z}^T)\,\Psi(\boldsymbol{\alpha})\right\|_F^2 \\
&\le \frac{1}{NM}\left(\left\|(\mathrm{I} - \mathrm{Z}\mathrm{Z}^T)(\Psi(\boldsymbol{\alpha}) - \widetilde{\Phi}_e(\boldsymbol{\alpha}))\right\|_F + \left\|(\mathrm{I} - \mathrm{Z}\mathrm{Z}^T)\widetilde{\Phi}_e(\boldsymbol{\alpha})\right\|_F\right)^2 \\
&\le \frac{1}{NM}\left(\left\|\,\Psi(\boldsymbol{\alpha}) - \widetilde{\Phi}_e(\boldsymbol{\alpha})\right\|_F + \left\|(\mathrm{I} - \mathrm{Z}\mathrm{Z}^T)\widetilde{\Phi}_e(\boldsymbol{\alpha})\right\|_F\right)^2,
\end{aligned} \tag{4.6}$$

where we used triangle inequality and $\|\mathrm{I} - \mathrm{Z}\mathrm{Z}^T\| \le 1$ for the spectral norm of the projector. For the last term in (4.6), we observe

$$\left\|(\mathrm{I} - \mathrm{Z}\mathrm{Z}^T)\widetilde{\Phi}_e(\boldsymbol{\alpha})\right\|_F = \left\|\widetilde{\mathrm{U}}\ \mathrm{diag}(0, \dots, 0, \widetilde{\sigma}_{n+1}, \dots, \widetilde{\sigma}_N)\ \widetilde{\mathrm{V}}^T\right\|_F = \left(\sum_{j=n+1}^{N} \widetilde{\sigma}_j^2\right)^{\frac{1}{2}}. \tag{4.7}$$

To handle the first term of (4.6), consider the extraction

$$\Phi_e(\boldsymbol{\alpha}) = \boldsymbol{\Phi} \times_2 \mathbf{e}^1(\boldsymbol{\alpha}) \times_3 \mathbf{e}^2(\boldsymbol{\alpha}) \dots \times_{D+1} \mathbf{e}^D(\boldsymbol{\alpha}) \tag{4.8}$$

and proceed using the triangle inequality

$$\left\|\,\Psi(\boldsymbol{\alpha}) - \widetilde{\Phi}_e(\boldsymbol{\alpha})\right\|_F \le \left\|\,\Psi(\boldsymbol{\alpha}) - \Phi_e(\boldsymbol{\alpha})\right\|_F + \left\|\,\Phi_e(\boldsymbol{\alpha}) - \widetilde{\Phi}_e(\boldsymbol{\alpha})\right\|_F. \tag{4.9}$$

We use the stability of interpolation (4.4) and (3.5) to bound the second term of (4.9). Specifically,

$$\begin{aligned}
\left\|\,\Phi_e(\boldsymbol{\alpha}) - \widetilde{\Phi}_e(\boldsymbol{\alpha})\right\|_F &= \left\|(\boldsymbol{\Phi} - \widetilde{\boldsymbol{\Phi}}) \times_2 \mathbf{e}^1(\boldsymbol{\alpha}) \times_3 \mathbf{e}^2(\boldsymbol{\alpha}) \dots \times_{D+1} \mathbf{e}^D(\boldsymbol{\alpha})\right\|_F \\
&\le \left\|\,\boldsymbol{\Phi} - \widetilde{\boldsymbol{\Phi}}\right\|_F \|\mathbf{e}^1(\boldsymbol{\alpha})\|_{\ell^2}\|\mathbf{e}^2(\boldsymbol{\alpha})\|_{\ell^2} \dots \|\mathbf{e}^D(\boldsymbol{\alpha})\|_{\ell^2} \\
&\le (C_e)^D \left\|\,\boldsymbol{\Phi} - \widetilde{\boldsymbol{\Phi}}\right\|_F \le (C_e)^D\,\widetilde{\varepsilon}\,\|\boldsymbol{\Phi}\|_F\,.
\end{aligned} \tag{4.10}$$

It remains to handle the first term in (4.9). At this point we need more precise assumptions on the smoothness of $\mathbf{u}(t, \boldsymbol{\alpha})$, the solution of (2.1). In particular,

$$\mathbf{u} \in C([0, T] \times \overline{\mathcal{A}})^M, \qquad \frac{\partial^{\mathbf{j}}\mathbf{u}}{\partial\alpha_1^{j_1}\dots\alpha_D^{j_D}} \in C([0, T] \times \overline{\mathcal{A}})^M, \quad |\mathbf{j}| \le p. \tag{4.11}$$

We note that (4.11) is guaranteed to hold if the unique solution to (2.1) exists on $(0, T_1)$ for all $\boldsymbol{\alpha} \in \mathcal{A}_1$, with $T < T_1$ and $\overline{\mathcal{A}} \subset \mathcal{A}_1$ and F is continuous with continuous partial derivatives in components of $\mathbf{u}$ and $\boldsymbol{\alpha}$ of order up to p [55].

Using interpolation property (4.3), we compute

$$\begin{aligned}
\left(\boldsymbol{\Phi} \times_2 \mathbf{e}^1(\boldsymbol{\alpha})\right)_{:,i_2,\dots,i_D,k} &= \sum_{j=1}^{n_1} e_j^1(\boldsymbol{\alpha})\mathbf{u}(t_k, \widehat{\alpha}_1^j, \widehat{\alpha}_2^{i_2}, \dots, \widehat{\alpha}_D^{i_D}) \\
&= \mathbf{u}(t_k, \alpha_1, \widehat{\alpha}_2^{i_2} \dots, \widehat{\alpha}_D^{i_D}) + \Delta^1_{:,i_2,\dots,i_D,k},
\end{aligned} \tag{4.12}$$

with the remainder term obeying a component-wise bound

$$|\Delta^1_{:,i_2,\dots,i_D,k}| \le C_a \sup_{a\in[\alpha_1^{\min},\alpha_1^{\max}]} \left|\frac{\partial^p\mathbf{u}}{\partial\alpha_1^p}(t_k, a, \widehat{\alpha}_2^{i_2}, \dots, \widehat{\alpha}_D^{i_D})\right| \delta_1^p, \tag{4.13}$$

where the absolute value of vectors is understood entry-wise. Analogously, we compute

$$\begin{aligned}\left(\boldsymbol{\Phi} \times_2 \mathbf{e}^1(\boldsymbol{\alpha}) \times_3 \mathbf{e}^2(\boldsymbol{\alpha})\right)_{:,i_3,\dots,i_D,k} &= \left((\boldsymbol{\Phi} \times_2 \mathbf{e}^1(\boldsymbol{\alpha})) \times_2 \mathbf{e}^2(\boldsymbol{\alpha})\right)_{:,i_3,\dots,i_D,k} \\ &= \sum_{j=1}^{n_2} e^2_j(\boldsymbol{\alpha}) \left(\mathbf{u}(t_k, \alpha_1, \widehat{\alpha}_2^{j}, \widehat{\alpha}_3^{i_3}, \dots, \widehat{\alpha}_D^{i_D}) + \Delta^1_{:,j,i_3,\dots,i_D,k}\right) \\ &= \mathbf{u}(t_k, \alpha_1, \alpha_2, \widehat{\alpha}_3^{i_3}, \dots, \widehat{\alpha}_D^{i_D}) + \Delta^2_{:,i_3,\dots,i_D,k} + \sum_{j=1}^{n_2} e^2_j(\boldsymbol{\alpha}) \Delta^1_{:,j,i_3,\dots,i_D,k},\end{aligned}$$

with a component-wise bound for the remainder

$$\begin{aligned}&\left|\Delta^2_{:,i_3,\dots,i_D,k} + \sum_{j=1}^{n_2} e^2_j(\boldsymbol{\alpha}) \Delta^1_{:,j,i_3,\dots,i_D,k}\right| \\ &\qquad \le C_a \sup_{a\in[\alpha_2^{\min},\alpha_2^{\max}]} \left|\frac{\partial^p \mathbf{u}}{\partial \alpha_2^p}(t_k, \alpha_1, a, \widehat{\alpha}_3^{i_3}, \dots, \widehat{\alpha}_D^{i_D})\right| \delta_2^p \\ &\qquad + C_e\, C_a \sup_{a\in[\alpha_1^{\min},\alpha_1^{\max}]} \left|\frac{\partial^p \mathbf{u}}{\partial \alpha_1^p}(t_k, a, \widehat{\alpha}_2^{i_2}, \dots, \widehat{\alpha}_D^{i_D})\right| \delta_1^p.\end{aligned} \tag{4.14}$$

Applying the same argument repeatedly, we obtain

$$\begin{aligned}(\Phi_e(\boldsymbol{\alpha}))_{:,k} &= \left(\boldsymbol{\Phi} \times_2 \mathbf{e}^1(\boldsymbol{\alpha}) \times_3 \mathbf{e}^2(\boldsymbol{\alpha}) \dots \times_{D+1} \mathbf{e}^D(\boldsymbol{\alpha})\right)_{:,k} \\ &= \mathbf{u}(t_k, \alpha_1, \alpha_2, \dots, \alpha_D) + \Delta_{:,k} = (\Psi(\boldsymbol{\alpha}))_{:,k} + \Delta_{:,k},\end{aligned} \tag{4.15}$$

with a component-wise bound for the remainder

$$\begin{aligned}\left|\Delta_{:,k}\right| \le C_a\Bigg(&\sup_{a\in[\alpha_D^{\min},\alpha_D^{\max}]} \left|\frac{\partial^p \mathbf{u}}{\partial \alpha_D^p}(t_k, \alpha_1, \dots, \alpha_{D-1}, a)\right| \delta_D^p + \cdots \\ + (C_e)^{D-2} &\sup_{a\in[\alpha_2^{\min},\alpha_2^{\max}]} \left|\frac{\partial^p \mathbf{u}}{\partial \alpha_2^p}(t_k, \alpha_1, a, \widehat{\alpha}_3^{i_3}, \dots, \widehat{\alpha}_D^{i_D})\right| \delta_2^p \\ + (C_e)^{(D-1)} &\sup_{a\in[\alpha_1^{\min},\alpha_1^{\max}]} \left|\frac{\partial^p \mathbf{u}}{\partial \alpha_1^p}(t_k, a, \widehat{\alpha}_2^{i_2}, \dots, \widehat{\alpha}_D^{i_D})\right| \delta_1^p\Bigg).\end{aligned}$$

Using the definition of the Frobenius norm, we arrive at

$$\|\,\Psi(\boldsymbol{\alpha}) - \Phi_e(\boldsymbol{\alpha})\|_F \le \sqrt{NM}\; C_a C_{\mathbf{u}} \max\left\{(C_e)^{(D-1)}, 1\right\}\; \delta^p, \tag{4.16}$$

where $C_{\mathbf{u}}$ depends only on the smoothness of $\mathbf{u}$ with respect to the variations of parameters $\boldsymbol{\alpha}$. More precisely, we can take $C_{\mathbf{u}} = \|\mathbf{u}\|_{C(0,T;C^p(\mathcal{A}))}$, which is bounded due to assumption (4.11).

Summarizing (4.6)–(4.16), we proved the following result.

Theorem 4.1. *Assume the solution* $\mathbf{u}$ *to* (2.1) *satisfies* (4.11), $\widehat{\mathcal{A}}$ *is a Cartesian grid in parameter domain* $\mathcal{A}$. *Then for any* $\boldsymbol{\alpha} \in \mathcal{A}$ *the interpolatory TROM reduced basis* $\mathcal{Z}(\boldsymbol{\alpha}) = \{\mathbf{z}_1, \dots, \mathbf{z}_n\}$ *delivers the following representation estimate*

$$\begin{aligned}&\frac{1}{3NM} \sum_{i=1}^{N} \left\| \mathbf{u}(t_i, \boldsymbol{\alpha}) - \sum_{j=1}^{n} \left\langle \mathbf{u}(t_i, \boldsymbol{\alpha}), \mathbf{z}_j \right\rangle \mathbf{z}_j \right\|_{\ell^2}^2 \\ &\quad \le \frac{1}{NM}\left((C_e)^{2D} \widetilde{\varepsilon}^2\, \|\boldsymbol{\Phi}\|_F + \sum_{i=n+1}^{N} \widetilde{\sigma}_i^2\right) + C_a C_{\mathbf{u}} \max\left\{(C_e)^{2(D-1)}, 1\right\} \delta^{2p},\end{aligned} \tag{4.17}$$

with $C_{\mathbf{u}} = \|\mathbf{u}\|_{C(0,T;C^p(\mathcal{A}))}$ *independent of* $\boldsymbol{\alpha}$, *sampling grid and* n.

We summarize here the definitions of quantities that appear in Theorem 4.1: N is a number of time steps for snapshot collection, while M is the spatial dimension of snapshots, i.e., $\mathbf{u}(t_i, \boldsymbol{\alpha}) \in \mathbb{R}^M$, $i = 1, \dots, N$; $\widetilde{\varepsilon}$ is the relative accuracy of the snapshot tensor compression from (3.5); $\widetilde{\sigma}_i$ are the singular values of $\widetilde{\Phi}_e(\boldsymbol{\alpha})$ from (3.15) (note that in TROMs we have an access to $\widetilde{\sigma}_i$ as the singular values of core matrices (3.22), (3.28) and (3.37) for CP-, HOSVD-, and TT-TROM, respectively); $\delta = \left(\sum_{i=1}^{D} \delta_i^p\right)^{\frac{1}{p}}$ is the grid step parameter of the Cartesian grid in $\mathcal{A}$; p is both the number of nearest grid points and the order of interpolation of the interpolation procedure (3.12)–(3.14); C_e is the interpolation stability constant from (4.4); and D is the dimension of parameter space.

For the general parameter sampling, prediction power analysis follows the same lines as above, simply setting $D = 1$. However, the order of interpolation p is slightly more difficult to formalize, so instead of (4.3)–(4.4) we rather assume

$$\sup_{\boldsymbol{\alpha}\in\mathcal{A}} \left| f(\boldsymbol{\alpha}) - \sum_{j=1}^{K} (\mathbf{e}(\boldsymbol{\alpha}))_j f(\widehat{\boldsymbol{\alpha}}_j) \right| \le \|f\|_{C^p(\mathcal{A})}\delta, \qquad \left(\sum_{j=1}^{K} |(\mathbf{e}(\boldsymbol{\alpha}))_j|^2\right)^{\frac{1}{2}} \le C_e \tag{4.18}$$

with some δ depending on $\widehat{\mathcal{A}}$. The prediction estimate then becomes

$$\begin{aligned} &\frac{1}{3NM} \sum_{i=1}^{N} \left\| \mathbf{u}(t_i, \boldsymbol{\alpha}) - \sum_{j=1}^{n} \langle \mathbf{u}(t_i, \boldsymbol{\alpha}), \mathbf{z}_j \rangle \mathbf{z}_j \right\|_{\ell^2}^2 \\ &\le \frac{1}{NM} \left((C_e)^2 \widetilde{\varepsilon}^2 \, \|\boldsymbol{\Phi}\|_F + \sum_{i=n+1}^{N} \widetilde{\sigma}_i^2 \right) + C_{\mathbf{u}} \max\{(C_e)^2, 1\}\delta^2, \end{aligned} \tag{4.19}$$

with $C_{\mathbf{u}} = \|\mathbf{u}\|_{C(0,T;C^p(\mathcal{A}))}$.

We finally, note that the feasibility of a sufficiently accurate lower rank representation of $\boldsymbol{\Phi}$ depends on the smoothness of $\mathbf{u}$ as a function of $\mathbf{x}$, t and $\boldsymbol{\alpha}$. This question can be addressed by considering tensor decompositions of multivariate functions, e.g. [23,56]. For these functional CP, HOSVD and hierarchical Tucker (including TT) formats, the dependence of compression ranks on $\widetilde{\varepsilon}$ from (3.5) and the regularity (smoothness) of $\mathbf{u}$ was studied in [57–61]. This compression property for multivariate functions was exploited to effectively represent solutions of parametric elliptic PDEs using tensor formats in [62–66] among other publications.

5. Numerical experiments

We perform several numerical experiments to assess the performance of the three TROM approaches and compare them to the conventional POD-ROM. The testing in Section 5.2 is performed for a dynamical system originating from a discretization of linear parameter-dependent heat equation. In Section 5.3 a similar set of tests is carried out for a time-dependent parameterized advection–diffusion system.

5.1. General parameter sampling interpolation scheme

For the numerical examples in the general parameter sampling setting we employ the following interpolation scheme. Fix an integer $q \ge D + 1$ and let $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_D) \in \mathcal{A}$ be an out-of-sample parameter vector. The interpolation scheme is based on the weighted minimum norm fit over q nearest neighbors of $\boldsymbol{\alpha}$ in the sampling set. Thus, we denote by $\widehat{\boldsymbol{\alpha}}_{i_1}, \dots, \widehat{\boldsymbol{\alpha}}_{i_q}$ the q closest parameter samples in $\widehat{\mathcal{A}}$ to $\boldsymbol{\alpha}$ and set $d_k = \|\widehat{\boldsymbol{\alpha}}_{i_k} - \boldsymbol{\alpha}\| > 0$, $k = 1, \dots, q$. Next, define the weighting matrix

$$\mathrm{D} = \mathrm{diag}(d_1^{-1}, \dots, d_q^{-1}) \in \mathbb{R}^{q \times q}.$$

Also, assemble the matrix

$$\mathrm{X} = \begin{bmatrix} (\widehat{\boldsymbol{\alpha}}_{i_1})_1 & (\widehat{\boldsymbol{\alpha}}_{i_2})_1 & \cdots & (\widehat{\boldsymbol{\alpha}}_{i_q})_1 \\ (\widehat{\boldsymbol{\alpha}}_{i_1})_2 & (\widehat{\boldsymbol{\alpha}}_{i_2})_2 & \cdots & (\widehat{\boldsymbol{\alpha}}_{i_q})_2 \\ \vdots & \vdots & \vdots & \vdots \\ (\widehat{\boldsymbol{\alpha}}_{i_1})_D & (\widehat{\boldsymbol{\alpha}}_{i_2})_D & \cdots & (\widehat{\boldsymbol{\alpha}}_{i_q})_D \\ 1 & 1 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{D+1 \times q}.$$

Solve the weighted minimum norm fitting problem to obtain

$$\widehat{\mathbf{a}} = \mathrm{D}(\mathrm{XD})^{\dagger} \begin{bmatrix} \boldsymbol{\alpha} \\ 1 \end{bmatrix} \in \mathbb{R}^q. \tag{5.1}$$

Note that the last row of X and the last entry of $(\boldsymbol{\alpha}, 1)^T$ enforces the condition that the entries of $\widehat{\mathbf{a}}$ sum to one. Meanwhile, the presence of the weighting matrix puts more emphasis on the neighbors of $\boldsymbol{\alpha}$ that are closest to it.

Once $\widehat{\mathbf{a}}$ is obtained from (5.1), we define a_j, $j = 1, \dots, K$, the entries of $\mathbf{e}(\boldsymbol{\alpha})$ as

$$a_j = \begin{cases} \widehat{a}_k, & \text{if } j = i_k \in \{i_1, \dots, i_q\} \\ 0, & \text{otherwise} \end{cases}$$

Clearly, such construction enforces representation (3.40).

5.2. Parameterized heat equation

We first assess performance of the three TROM approaches on a dynamical system resulting from the discretization of a heat equation

$$w_t = \Delta w, \tag{5.2}$$

in a rectangular domain with three holes $\Omega = \Omega_r \setminus (\Omega_1 \cup \Omega_2 \cup \Omega_3) \subset \mathbb{R}^2$, where $\Omega_r = [0, 10] \times [0, 4]$, and the holes are $\Omega_1 = [1, 3] \times [1, 3]$, $\Omega_2 = [4, 6] \times [1, 3]$, $\Omega_3 = [7, 9] \times [1, 3]$. The PDE and geometry of Ω follow that of [67], while the boundary conditions are modified from those used in [67], as described below.

We parametrize the system with $D = 4$ parameters that enter the boundary conditions. Convection boundary conditions are enforced on the left side of the rectangle $\Gamma_o = 0 \times [0, 4]$ and on the boundaries of each hole $\partial\Omega_j$, $j = 1, 2, 3$. Explicitly,

$$(\mathbf{n} \cdot \nabla w + \alpha_1 (w - 1))|_{\Gamma_o} = 0, \tag{5.3}$$

and

$$\left(\mathbf{n} \cdot \nabla w + \frac{1}{2} w\right)\Big|_{\partial\Omega_j} = \frac{1}{2}\alpha_{j+1}, \quad j = 1, 2, 3, \tag{5.4}$$

i.e., the first parameter in $\boldsymbol{\alpha} \in \mathbb{R}^4$ is Biot number at Γ_o with a fixed outside temperature $t_o = 1$, while the other three parameters are the temperatures at $\partial\Omega_j$, $j = 1, 2, 3$, respectively, with Biot numbers equal to $\frac{1}{2}$ on all three hole boundaries. The rest of the boundary of Ω is assumed to be insulated

$$(\mathbf{n} \cdot \nabla w)|_{\partial\Omega_r \setminus \Gamma_o} = 0. \tag{5.5}$$

In (5.3)–(5.5), $\mathbf{n}$ is the outer unit normal. Observe that the boundary conditions (5.3)–(5.5) can be combined into

$$(\mathbf{n} \cdot \nabla w + q(\mathbf{x}, \boldsymbol{\alpha}) w)|_{\partial\Omega} = g(\mathbf{x}, \boldsymbol{\alpha}),$$

for the appropriate choices of $q(\mathbf{x}, \boldsymbol{\alpha})$ and $g(\mathbf{x}, \boldsymbol{\alpha})$ defined on $\partial\Omega$ with $\boldsymbol{\alpha} \in \mathcal{A}$, a parameter domain that we take to be the 4D box $\mathcal{A} = [0.01, 0.5] \times [0, 0.9]^3$. The initial temperature is taken to be zero throughout Ω.

The system (5.2)–(5.5) is discretized with P_2 finite elements on a quasi-uniform triangulation of Ω resulting in $M = 3,562$ spatial degrees of freedom. The choice of standard nodal basis functions $\{\theta_j(\mathbf{x})\}_{j=1}^{M}$ defines the mass $\mathrm{M} \in \mathbb{R}^{M \times M}$ and stiffness $\mathrm{K} \in \mathbb{R}^{M \times M}$ matrices, as well as boundary terms $\mathrm{Q}(\boldsymbol{\alpha}) \in \mathbb{R}^{M \times M}$, $\mathbf{g}(\boldsymbol{\alpha}) \in \mathbb{R}^M$ with entries given by

$$(\mathrm{Q})_{ij}(\boldsymbol{\alpha}) = \int_{\partial\Omega} q(\mathbf{x}, \boldsymbol{\alpha}) \theta_j(\mathbf{x}) \theta_i(\mathbf{x}) ds_{\mathbf{x}}, \quad (\mathbf{g})_j(\boldsymbol{\alpha}) = \int_{\partial\Omega} g(\mathbf{x}, \boldsymbol{\alpha}) \theta_j(\mathbf{x}) ds_{\mathbf{x}},$$

for $i, j = 1, \dots, M$.

The vector-valued function of nodal values $\mathbf{u}(t, \boldsymbol{\alpha}) : [0, T) \times \mathcal{A} \to \mathbb{R}^M$ solves

$$\mathrm{M}\mathbf{u}_t + (\mathrm{K} + \mathrm{Q}(\boldsymbol{\alpha}))\, \mathbf{u} = \mathbf{g}(\boldsymbol{\alpha}), \tag{5.6}$$

i.e., it satisfies the dynamical system of the form (2.1) with

$$F(t, \mathbf{u}, \boldsymbol{\alpha}) = -\mathrm{M}^{-1} (\mathrm{K} + \mathrm{Q}(\boldsymbol{\alpha}))\, \mathbf{u} + \mathrm{M}^{-1}\mathbf{g}(\boldsymbol{\alpha})$$

and the initial condition $\mathbf{u}(0, \boldsymbol{\alpha}) = \mathbf{u}_0 = \mathbf{0} \in \mathbb{R}^M$ corresponding to zero initial temperature condition for w.

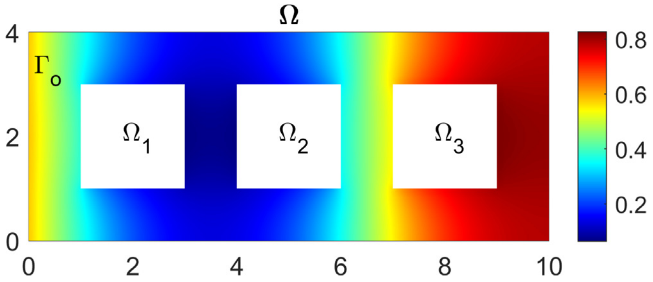

Fig. 5.1. Domain Ω and the solution $w(T, \mathbf{x}, \boldsymbol{\alpha})$ of the heat Eq. (5.2)–(5.5) corresponding to $\boldsymbol{\alpha} = (0.5, 0, 0, 0.9)^T$.

We compute the snapshots $\boldsymbol{\phi}_k = \mathbf{u}(t_k, \boldsymbol{\alpha})$ by time-stepping (5.6) at $t_k = 0.2k$, $k = 1, 2, \ldots, N$, with $N = 100$ time steps and $T = 20$ using Crank–Nicolson scheme. Setting $\Theta(\mathbf{x}) = [\theta_1(\mathbf{x}), \ldots, \theta_M(\mathbf{x})]$ allows to express the solution $w(t, \mathbf{x}, \boldsymbol{\alpha})$ of (5.2)–(5.5) as $w(t, \mathbf{x}, \boldsymbol{\alpha}) = \Theta(\mathbf{x})\mathbf{u}(t, \boldsymbol{\alpha})$, hence the solution snapshots are

$$w(t_k, \mathbf{x}, \boldsymbol{\alpha}) = \Theta(\mathbf{x})\mathbf{u}(t_k, \boldsymbol{\alpha}). \tag{5.7}$$

The setting is illustrated in Fig. 5.1, where we display the domain Ω along with solution $w(T, \mathbf{x}, \boldsymbol{\alpha})$ corresponding to parameter values $\boldsymbol{\alpha} = (0.5, 0, 0, 0.9)^T$.

For an arbitrary but fixed $\boldsymbol{\alpha} \in \mathcal{A}$ let $\mathrm{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{R}^{M \times n}$ be a matrix with columns being vectors constituting the reduced basis, i.e. $\mathrm{Z} = \mathrm{U}\mathrm{U}_c$ for TROM. Then, the projection ROM of (5.6) is

$$\widetilde{\mathrm{M}}\widetilde{\mathbf{u}}_t + \left(\widetilde{\mathrm{K}} + \widetilde{\mathrm{Q}}(\boldsymbol{\alpha})\right)\widetilde{\mathbf{u}} = \widetilde{\mathbf{g}}(\boldsymbol{\alpha}), \tag{5.8}$$

where

$$\begin{aligned} &\widetilde{\mathrm{M}} = \mathrm{Z}^T\mathrm{M}\mathrm{Z} \in \mathbb{R}^{n\times n}, \quad \widetilde{\mathrm{K}} = \mathrm{Z}^T\mathrm{K}\mathrm{Z} \in \mathbb{R}^{n\times n}, \\ &\widetilde{\mathrm{Q}}(\boldsymbol{\alpha}) = \mathrm{Z}^T\mathrm{Q}(\boldsymbol{\alpha})\mathrm{Z} \in \mathbb{R}^{n\times n}, \quad \widetilde{\mathbf{g}}(\boldsymbol{\alpha}) = \mathrm{Z}^T\mathbf{g}(\boldsymbol{\alpha}) \in \mathbb{R}^{n}, \end{aligned}$$

and the initial condition is $\widetilde{\mathbf{u}}(0, \boldsymbol{\alpha}) = \mathrm{Z}^T\mathbf{u}_0 = \mathbf{0} \in \mathbb{R}^n$. As discussed in Section 3.10, the evaluation of (5.8) can be effectively split between the offline and online stages. Solving (5.8) for $\widetilde{\mathbf{u}}(t, \boldsymbol{\alpha})$ allows to recover the approximate solution at times t_k as

$$\widetilde{w}(t_k, \mathbf{x}, \boldsymbol{\alpha}) = \Theta(\mathbf{x})\, \mathrm{Z}\, \widetilde{\mathbf{u}}(t_k, \boldsymbol{\alpha}) \approx w(t_k, \mathbf{x}, \boldsymbol{\alpha}). \tag{5.9}$$

5.2.1. In-sample prediction and compression study

We begin TROM assessment with in-sample prediction and compression study for the linear parabolic system described Section 5.2. To measure TROM predictive power and to compare it to that of POD-ROM, we sample $\mathcal{A}$ uniformly in each direction with $n_1 \times n_2 \times n_3 \times n_4 = 9 \times 5 \times 5 \times 5$ samples, for a total of $K = 1,125$ samples in the set $\widehat{\mathcal{A}} = \{\widehat{\boldsymbol{\alpha}}_1, \ldots, \widehat{\boldsymbol{\alpha}}_K\}$. For each of the three TROMs and for POD-ROM we compute the following in-sample prediction error

$$E_{L^2(\widehat{\mathcal{A}})} = \left(\frac{1}{MNK} \sum_{j=1}^{K} \left\| (\mathrm{I} - \mathrm{Z}\mathrm{Z}^T)\,\Phi_e(\widehat{\boldsymbol{\alpha}}_j) \right\|_F^2 \right)^{1/2}, \tag{5.10}$$

to quantify the ability of the CP-TROM, HOSVD-TROM, TT-TROM local bases and POD-ROM basis to represent original snapshots for in-sample parameter values. Note that $E^2_{L^2(\widehat{\mathcal{A}})}$ is the quantity from (3.11) averaged over $\widehat{\mathcal{A}}$ and scaled by $(MN)^{-1}$.

We report in Tables 5.1 and 5.2 the in-sample prediction errors and compression factors CF defined in (3.44) for Cartesian grid-based and general samplings, respectively. For general sampling we organize the sampling parameters from the Cartesian grid in 1D array leading to snapshot tensors or order 3. Experimenting with the same number of randomly sampled parameters showed very similar compression rates and in-sample prediction errors and so those are not reported here. The results are reported in Tables 5.1 and 5.2 for a number of decreasing values of $\widetilde{\varepsilon}$ and correspondingly increasing n, such that $n \le \min(\widetilde{N}, \widetilde{r}_{D+1})$, where $\widetilde{N}$ and $\widetilde{r}_{D+1}$ are the last Tucker and compression

Table 5.1
Cartesian grid-based sampling in-sample prediction and compression study results reporting prediction errors $E_{L^2(\widehat{\mathcal{A}})}$ for HOSVD, TT, POD, the number of compressed tensors elements transmitted to the online stage, and the corresponding compression factors CF. $\widetilde{\boldsymbol{\Phi}}$ ranks comprise: Tucker ranks for HOSVD-TROM in format $[\widetilde{M}, \widetilde{n}_1, \widetilde{n}_2, \widetilde{n}_3, \widetilde{n}_4, \widetilde{N}]$, and compression ranks for TT-TROM in format $[\widetilde{r}_1, \widetilde{r}_2, \widetilde{r}_3, \widetilde{r}_4, \widetilde{r}_5]$.

$\widetilde{\varepsilon}$		1e−4	1e−5	1e−6	1e−7	1e−9
n		12	16	19	23	30
$E_{L^2(\widehat{\mathcal{A}})}$	HOSVD	3.45e−05	3.85e−06	2.78e−07	5.61e−08	–
	TT	6.15e−05	5.58e−06	5.21e−07	5.03e−08	5.41e−10
	POD	1.67e−03	9.06e−04	5.84e−04	3.05e−04	7.76e−05
$\widetilde{\boldsymbol{\Phi}}$ ranks	HOSVD	[34, 4, 2, 2, 2, 12]	[46, 5, 2, 2, 2, 16]	[57, 6, 2, 2, 2, 20]	[66, 7, 2, 2, 2, 23]	–
	TT	[34, 35, 30, 21, 12]	[46, 48, 41, 29, 16]	[57, 61, 51, 36, 19]	[69, 74, 60, 43, 23]	[99, 97, 80, 57, 30]
#online($\widetilde{\boldsymbol{\Phi}}$)	HOSVD	13 122	29 515	54 804	85 101	–
	TT	20 382	37 993	53 877	86 022	156 607
CF	HOSVD	3.05e+4	1.36e+4	7.31e+3	4.71e+3	–
	TT	1.97e+4	1.05e+4	7.42e+3	4.66e+3	2.56e+3

Table 5.2
General parameter sampling in-sample prediction and compression study results reporting prediction errors $E_{L^2(\widehat{\mathcal{A}})}$ for HOSVD, TT, POD, the number of compressed tensors elements transmitted to the online stage, and the corresponding compression factors CF. $\widetilde{\boldsymbol{\Phi}}$ ranks comprise: Tucker ranks for HOSVD-TROM in format $[\widetilde{M}, \widetilde{N}]$, and compression ranks for TT-TROM in format $[\widetilde{r}_1, \widetilde{r}_2]$.

$\widetilde{\varepsilon}$		1e−4	1e−5	1e−6	1e−7	1e−9
n		12	16	19	23	30
$E_{L^2(\widehat{\mathcal{A}})}$	HOSVD	5.69e−05	5.29e−06	4.66e−07	7.93e−08	–
	TT	5.63e−05	5.95e−06	5.72e−07	5.56e−08	5.43e−09
	POD	2.05e−03	1.12e−03	5.84e−04	3.05e−04	8.46e−05
$\widetilde{\boldsymbol{\Phi}}$ ranks	HOSVD	[33, 11]	[45, 16]	[56, 19]	[65, 23]	–
	TT	[32, 11]	[43, 15]	[55, 19]	[66, 23]	[95, 29]
#online($\widetilde{\boldsymbol{\Phi}}$)	HOSVD	11 904	18 450	26 268	36 680	–
	TT	396 011	725 640	1 175 644	1 633 522	3 099 404
CF	HOSVD	3.37e+4	2.17e+4	1.53e+4	1.09e+4	–
	TT	1.01e+3	5.52e+2	3.41e+2	2.45e+2	1.29e+2

ranks, respectively, for HOSVD- and TT-TROM. Available compression algorithm failed to deliver the accuracy of 1e-9 for HOSVD, so we report only TT statistics for this extreme value of $\widetilde{\varepsilon}$. Note that we leave $E_{L^2(\widehat{\mathcal{A}})}$ for CP-ROM out since there is no direct way to control its relative error $\widetilde{\varepsilon}$, as discussed at the end of Section 3.5. Instead, compression factors and canonical ranks for CP-TROM are illustrated in Fig. 5.2.

We observe in Tables 5.1 and 5.2 that both HOSVD- and TT-TROM outperform POD-ROM in terms of prediction error up to four orders of magnitude for larger n and correspondingly small $\widetilde{\varepsilon}$. This result is consistent across both Cartesian grid-based and general parameter samplings. The #online($\widetilde{\boldsymbol{\Phi}}$) values tell us that for Cartesian based sampling, HOSVD and TT are comparable in terms of memory and data transmission requirements with HOSVD doing somewhat better for lower representation accuracy for the snapshot tensor of order 6. Compression achieved varies in $\widetilde{\varepsilon}$ (as should expected) and gives more than 3 orders of saving even for the finest available representation accuracy. If the Cartesian structure of $\widehat{\mathcal{A}}$ is abandoned and snapshots are organized in tensors of order 3, then HOSVD format has a clear advantage over TT in terms of compression achieved; see #online($\widetilde{\boldsymbol{\Phi}}$) and CF statistics in Table 5.2.

For CP-TROM we display in Fig. 5.2 compression factors CF and canonical ranks R for a number of values of $\widetilde{\varepsilon}$. For comparison we also show on the same plots CF for HOSVD, which shows that HOSVD-TROM is on par with CP-TROM if Cartesian sampling allows to organize snapshots in a higher order tensor. However, we were able to compute $\widetilde{\boldsymbol{\Phi}}$ in CP compressed format only up to moderate values of $\widetilde{\varepsilon}$, since the corresponding CP rank was

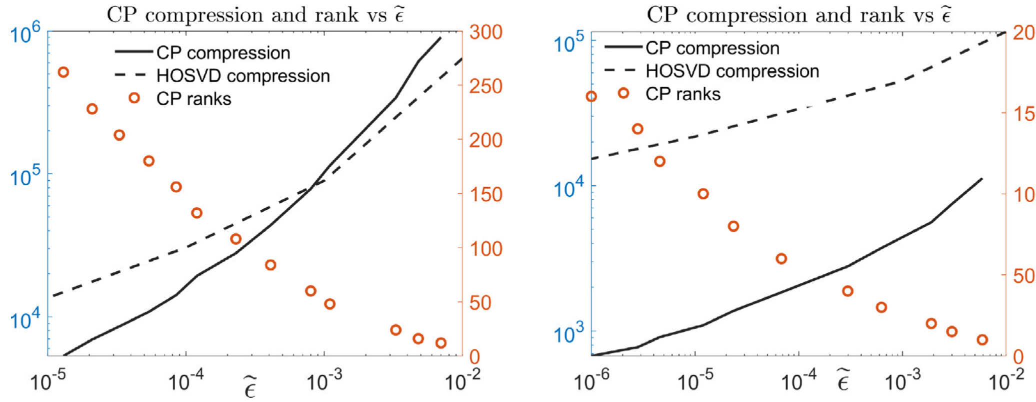

Fig. 5.2. CP-TROM compression factor CF and canonical rank R as functions of $\widetilde{\varepsilon}$. Left: Cartesian grid-based sampling; Right: general sampling.

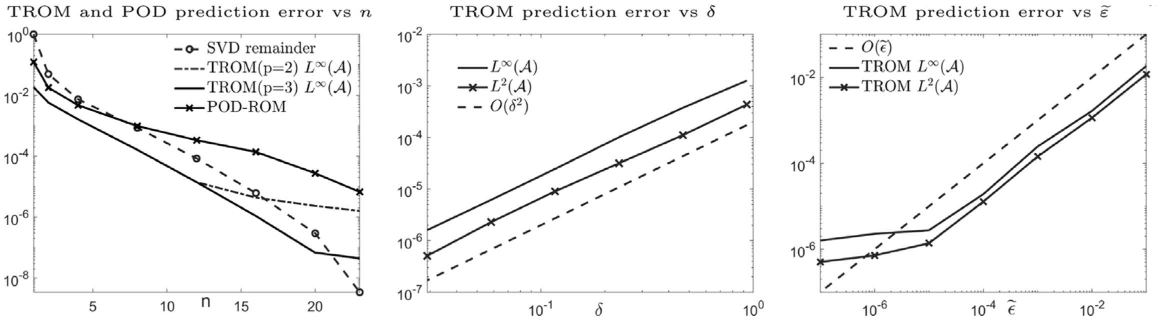

Fig. 5.3. Out-of-sample prediction error of TROM versus ROM basis dimension n (left plot), parameter mesh size δ (middle plot) and tensor compression accuracy $\widetilde{\varepsilon}$ (right plot).

growing fast as $\widetilde{\varepsilon}$ decreases (see the left plot in Fig. 5.2). Smaller $\widetilde{\varepsilon}$ become feasible with CP if the snapshots are organized in tensors of order 3, but in this case HOSVD-TROM achieves much better compression than CP-TROM (see the right plot in Fig. 5.2). We conclude that for this example with a relatively small number of parameters ($D = 4$) HOSVD-TROM appears to be the best performing TROM. We finally note that for the same compression accuracy $\widetilde{\varepsilon}$ (if it was achieved) CP-TROM demonstrated very similar in-sample prediction error as HOSVD- and TT-ROMs. This observation largely carries over to out-of-sample representation studied next.

5.2.2. *Out-of-sample prediction study*

To quantify the ability of the CP-, HOSVD- and TT-TROM local bases to represent the solution of (5.2) for arbitrary out-of-sample parameter values, we use $E_{L^2(\mathcal{A})}$ which is defined as in (5.10) but with $\boldsymbol{\alpha}$ (in place of $\widehat{\boldsymbol{\alpha}}$) running through a large number of random points from $\mathcal{A}$. We also use

$$E_{L^\infty(\mathcal{A})} = \sup_{\boldsymbol{\alpha}\in\mathcal{A}} \left(\frac{1}{MN} \left\| (\mathrm{I} - \mathrm{Z}\mathrm{Z}^T) \Phi_e(\boldsymbol{\alpha}) \right\|_F^2 \right)^{1/2},$$

for the maximum of representation error over the parameter domain. An estimate of $E_{L^\infty(\mathcal{A})}$ is given by Theorem 4.1. Regarding constants appearing in (4.17) we note that for our choice of the uniform grid in $\mathcal{A}$ and $p = 2, 3$ one computes $C_e = 1$, $C_a = \frac{1}{8}$ ($p = 2$) and $C_a = \frac{1}{48}$ ($p = 2$), while constant $C_{\mathbf{u}}$ is hard to evaluate. Experimentally we found that the grid in the fourth parameter should be finer than for the first three parameters to balance the observed error suggesting that the solution is less smooth as a function of α_4. To study the error dependence on the parameter mesh size δ, we reduce the number of parameters to two, letting $\alpha_1 = \alpha_2 = \alpha_3$ in (5.4).

In Fig. 5.3 we plot $E_{L^\infty(\mathcal{A})}$ and $E_{L^2(\mathcal{A})}$ versus ROM basis dimension n, parameter mesh size δ and tensor compression accuracy $\widetilde{\varepsilon}$. The results were computed using 100 randomly distributed parameters from $\mathcal{A}$ to evaluate the error quantities for HOSVD-TROM. For TT-TROM and CP-TROM the error dependence on n, δ and $\widetilde{\varepsilon}$ was virtually the same and are omitted. Variants of TROM of course may differ by complexity. While the cost of offline phase of computing $\widetilde{\boldsymbol{\Phi}}$ varies significantly depending on the format, the online distribution of costs was persistent for all three formats showing between 30% and 40% of online time spent on finding U_c (coordinates of local basis), less then 5% on projection to $\boldsymbol{\alpha}$-specific coordinates, and about 60% of time on the integration of the projected system (this last step is common with standard POD approach).

The left plot in Fig. 5.3 shows the $E_{L^\infty(\mathcal{A})}$ error for different values of n and compares it to the error of POD-ROM. We use $\widetilde{\varepsilon} =$ 1e-7 and 65 × 33 parameter grid. Such fine compression accuracy and grid allows us to isolate the effect of the second term on the right hand side of (4.17). Indeed, we see that the error curve follows closely the graph of "SVD remainder" (the maximum over all $\boldsymbol{\alpha}$ of the second term on the right-hand side of (4.17)) until the interpolation error starts dominating for larger n. As can be expected, the interpolation error for $p = 3$ starts affecting $E_{L^\infty(\mathcal{A})}$ for larger n than the interpolation error for $p = 2$.

The middle plot in Fig. 5.3 demonstrates that both $E_{L^\infty(\mathcal{A})}$ and $E_{L^2(\mathcal{A})}$ for TROM decrease as $O(\delta^2)$ for $p = 2$ (computed with $\widetilde{\varepsilon} =$ 1e-7 and $n = \widetilde{N}$) just as predicted by (4.17). Likewise, the right plot in Fig. 5.3 gives evidence for $O(\widetilde{\varepsilon})$ decrease of $E_{L^\infty(\mathcal{A})}$ and $E_{L^2(\mathcal{A})}$ (computed with 65 × 33 parameter grid and $n = \widetilde{N}$) in accordance to (4.17) as long the first term on the right hand side of (4.17) dominates.

5.2.3. Out-of-sample TROMs vs. POD-ROM performance: heat equation

Performance of TROM and POD-ROM may vary for different out-of-sample parameter values. Therefore, assessment of TROM vs. POD-ROM performance in this section is conducted in a statistical setting. The quantities of interest are computed for $N_r \gg 1$ out-of -sample realizations of $\boldsymbol{\alpha}^{(r)} \in \mathcal{A}$, $r = 1, 2, \ldots, N_r$, where we use $N_r = 200$ for the numerical studies below. Realizations $\boldsymbol{\alpha}^{(r)} = \left(\alpha_1^{(r)}, \alpha_2^{(r)}, \alpha_3^{(r)}, \alpha_4^{(r)}\right)^T$ are drawn at random from $\mathcal{A} = [0.01, 0.5] \times [0, 0.9]^3$ with each α_i distributed uniformly on $[\alpha_i^{\min}, \alpha_i^{\max}]$, $i = 1, 2, 3, 4$.

For statistical tests we use the following quantities to measure performance of TROM. First, we introduce the relative $L^\infty(0, T, L^2(\Omega))$ ROM solution error

$$R_X(\boldsymbol{\alpha}) = \frac{\max_{k=1,\ldots,N} \|\widetilde{w}(t_k, \mathbf{x}, \boldsymbol{\alpha}) - w(t_k, \mathbf{x}, \boldsymbol{\alpha})\|_{L^2(\Omega)}}{\max_{k=1,\ldots,N} \|w(t_k, \mathbf{x}, \boldsymbol{\alpha})\|_{L^2(\Omega)}} \approx \frac{\sup_{t\in[0,T]} \|\widetilde{w}(t, \mathbf{x}, \boldsymbol{\alpha}) - w(t, \mathbf{x}, \boldsymbol{\alpha})\|_{L^2(\Omega)}}{\sup_{t\in[0,T]} \|w(t, \mathbf{x}, \boldsymbol{\alpha})\|_{L^2(\Omega)}}, \tag{5.11}$$

which we compute for each realization $\boldsymbol{\alpha}^{(r)}$, $r = 1, 2, \ldots, N_r$, for both POD-ROM and each of the three TROMs with X∈ {POD, CP, HOSVD, TT}. The true and reduced order snapshots for (5.11) are computed as in (5.7) and (5.9), respectively. We report the mean, minimum, and standard deviation of the three relative *gain* distributions

$$G_{\mathrm{X}}^{(r)} = \frac{R_{\mathrm{POD}}\left(\boldsymbol{\alpha}^{(r)}\right)}{R_{\mathrm{X}}\left(\boldsymbol{\alpha}^{(r)}\right)}, \quad r = 1, 2, \ldots, N_r, \tag{5.12}$$

for X∈ {CP, HOSVD, TT}, which quantify the error decrease of CP-TROM, HOSVD-TROM and TT-TROM, respectively, relative to POD-ROM. We study the dependency of (5.12) with respect to K, the number of sampled parameter values in $\mathcal{A}$, and n, the dimension of the reduced space. The results are reported for both Cartesian grid-based sampling and general parameter sampling.

We present in Table 5.3 the dependence of relative gain statistics on the values of K (statistics in the table were computed setting $\widetilde{\varepsilon} = 10^{-5}$ as the targeted accuracy of HOSVD-TROM, TT-TROM, and $R = 250$ as the targeted rank for CP-TROM). We observe that as K increases, TROMs become both more accurate on average and more robust. The robustness is observed in both the increase of the minimum relative gain and the decrease of its standard deviation. On average, for $K = 3,430$, all three TROMs are almost 40 times more accurate compared to POD-ROM. The performance difference between the TROMs themselves is basically negligible for this particular study.

The performance of TROMs in the example above is limited by the relatively small value of $n = 10$. The effect of increasing n to 20 while keeping $K = 3,430$ is shown in Table 5.4 (statistics in the table were computed setting $\widetilde{\varepsilon} = 10^{-7}$ as the targeted accuracy of HOSVD-TROM, TT-TROM, and $R = 250$ as the targeted rank for CP-TROM). While the worst case scenario stays relatively unchanged, the average accuracy gain by TROMs is over two orders of magnitude. An outlier here is CP-TROM that underperforms HOSVD- and TT-TROM in case of Cartesian grid-based parameter sampling. Aside from that, performance difference of TROMs for general and Cartesian samplings is negligible. It is also interesting to see if the interpolation of approximate snapshots

Table 5.3
Statistics of relative gain (5.12) for various values of K, the number of sampled parameter values in $\mathcal{A}$. The study is performed with $n = 10$.

		General parameter sampling			Cartesian grid-based sampling		
K	G_X	CP	HOSVD	TT	CP	HOSVD	TT
	mean	**24.17**	**24.17**	**24.17**	**24.76**	**25.08**	**25.08**
$135 =$	min	0.49	0.49	0.49	0.56	0.56	0.56
5×3^3	std	17.33	17.33	17.33	16.88	17.31	17.32
	mean	**34.60**	**34.60**	**34.61**	**35.21**	**35.52**	**35.51**
$1000 =$	min	2.80	2.80	2.80	1.72	1.72	1.72
8×5^3	std	15.36	15.36	15.37	15.03	15.11	15.11
	mean	**38.14**	**38.15**	**38.15**	**37.80**	**38.80**	**38.80**
$3430 =$	min	3.81	3.81	3.81	4.20	4.45	4.43
10×7^3	std	14.20	14.21	14.22	12.96	13.61	13.62

Table 5.4
Statistics of relative gain (5.12) for $n = 10$ and 20. The study is performed for $K = 10 \times 7 \times 7 \times 7 = 3430$.

		General parameter sampling			Cartesian grid-based sampling		
n	G_X	CP	HOSVD	TT	CP	HOSVD	TT
	mean	**38.14**	**38.15**	**38.15**	**37.80**	**38.80**	**38.80**
10	min	3.81	3.81	3.81	4.20	4.45	4.43
	std	14.20	14.21	14.22	12.96	13.61	13.62
	mean	**155.00**	**158.97**	**161.75**	**49.80**	**155.65**	**154.03**
20	min	4.59	4.54	4.55	1.51	5.26	5.23
	std	513.33	530.50	557.86	39.48	551.92	541.88

from the lower-rank tensor $\widetilde{\boldsymbol{\Phi}}$ alone, i.e., without finding a reduced local basis and solving the projected problem, gives reasonable approximation to high-fidelity solutions for out-of-sample parameters. Such interpolation-only predicted solution for incoming $\boldsymbol{\alpha} \in \mathcal{A}$ is given by columns of $\widetilde{\Phi}(\boldsymbol{\alpha})$. Repeating the experiment with Cartesian grid-based sampling and other parameters the same as used for results in Table 5.3 and Table 5.4, we find that the mean relative gain of HOSVD-tROM compared to interpolation-only approach is {2.10, 1.51, 1.13} for $n = 10$, $K = \{135, 1000, 3430\}$ and {7.64, 7.60, 7.54} for $n = 20$ and same values of K. The numbers were very close for other two TROMs. We see that TROM based on solving projected problem in general gives more accurate results then pure interpolation, especially if more vectors are included in the reduced basis. If n is fixed, then for sufficiently fine sampling the interpolation-only approach delivers the same (or even better) accuracy.

We conclude the numerical study for the parameterized heat equation with a comparison between the three TROMs and another variant of POD-ROM often used in practice, the so-called greedy POD or POD-Greedy approach to computing the reduced basis [68]. Replacing the conventional POD-ROM computation with POD-Greedy algorithm, we perform the same out-of-sample performance study as presented in Table 5.4. The resulting relative gain statistics are reported in Table 5.5. Qualitatively, the results are very similar to those in Table 5.4. However, quantitatively we observe 20% to 40% increased relative gain for all TROMs. This is consistent with the fact that POD-Greedy reduced basis is sub-optimal compared to the conventional POD-ROM reduced basis computed from the snapshots corresponding to all parameter values in $\widehat{\mathcal{A}}$.

In terms of computational performance for this specific setting, POD-Greedy algorithm was found to be significantly slower in the offline stage than all three TROM approaches and the conventional POD-ROM even if one includes into the offline cost of TROM and POD-ROM the computation of KN snapshots $\boldsymbol{\phi}_k(\widehat{\boldsymbol{\alpha}}_j)$, $k = 1, \dots, N$, $j = 1, \dots, K$. Indeed, the bulk of computational cost of POD-Greedy approach is in the evaluation of error estimator that has to be performed at each of its n iterations for all $\widehat{\boldsymbol{\alpha}}_j$, $j = 1, \dots, K$, to determine the sample with the largest error. In turn, each evaluation of the estimator requires the computation of the residual of the chosen time-stepping scheme (e.g., Crank–Nicolson) that needs $O(N)$ matrix–vector products of a dense $M \times n$ matrix and a vector in $\mathbb{R}^n$. Thus, error estimator evaluations alone account for $O(KNMn^2)$ operations of POD-Greedy offline stage. We note that in other situations, when computing the snapshots is much more expensive compared to the evaluation of

Table 5.5
Statistics of relative gain (5.12) for POD-ROM with POD-Greedy reduced basis for $n = 10$ and 20. The study is performed for $K = 10 \times 7 \times 7 \times 7 = 3430$.

		General parameter sampling			Cartesian grid-based sampling		
n	G_X	CP	HOSVD	TT	CP	HOSVD	TT
10	mean	**53.14**	**53.14**	**53.14**	**54.02**	**54.12**	**54.12**
	min	7.01	7.01	7.01	7.43	7.42	7.42
	std	18.63	18.63	18.63	18.03	18.03	18.03
20	mean	**188.57**	**195.57**	**198.77**	**62.18**	**193.66**	**191.86**
	min	7.39	7.35	7.36	1.87	8.51	8.47
	std	571.50	607.44	634.94	49.07	649.81	639.61

the error estimator (e.g., when the matrices of systems to be solved on each time step are dense), both POD-ROM and TROM may benefit from a greedy approach to parameter sampling.

5.3. Advection–diffusion PDE

In the second numerical example, for assessing performance of the three TROM approaches we are interested in a case with a higher order of parameter space compared to $D = 4$ in Section 5.2. To that end we set up a dynamical system resulting from the discretization of a linear advection–diffusion equation

$$w_t = \nu \Delta w - \boldsymbol{\eta}(\mathbf{x}, \boldsymbol{\alpha}) \cdot \nabla w + f(\mathbf{x}), \tag{5.13}$$

in a unit square domain $\Omega = [0, 1] \times [0, 1] \subset \mathbb{R}^2$, $\mathbf{x} = (x_1, x_2)^T \in \Omega$. Here ν is a constant diffusion coefficient, $\boldsymbol{\eta} : \Omega \times \mathcal{A} \to \mathbb{R}^2$ is the advection field and $f(\mathbf{x})$ is a Gaussian source

$$f(\mathbf{x}) = \frac{1}{2\pi\sigma_s^2} \exp\left(-\frac{(x_1 - x_1^s)^2 + (x_2 - x_2^s)^2}{2\sigma_s^2}\right), \tag{5.14}$$

where we take $\sigma_s = 0.05$, $x_1^s = x_2^s = 0.25$. We enforce homogeneous Neumann boundary conditions and zero initial condition

$$(\mathbf{n} \cdot \nabla w)|_{\partial\Omega} = 0, \quad w(0, \mathbf{x}, \boldsymbol{\alpha}) = 0. \tag{5.15}$$

The model is parameterized with $D = 9$ parameters with only the advection field $\boldsymbol{\eta}$ depending on $\boldsymbol{\alpha} \in \mathbb{R}^9$. The advection field is given as follows

$$\boldsymbol{\eta}(\mathbf{x}, \boldsymbol{\alpha}) = \begin{pmatrix} \eta_1(\mathbf{x}, \boldsymbol{\alpha}) \\ \eta_2(\mathbf{x}, \boldsymbol{\alpha}) \end{pmatrix} = \begin{pmatrix} \cos\alpha_9 \\ \sin\alpha_9 \end{pmatrix} + \frac{1}{\pi} \begin{pmatrix} \partial_{x_2} h(\mathbf{x}, \boldsymbol{\alpha}) \\ -\partial_{x_1} h(\mathbf{x}, \boldsymbol{\alpha}) \end{pmatrix}, \tag{5.16}$$

where $h(\mathbf{x})$ is the cosine trigonometric polynomial

$$\begin{aligned} h(\mathbf{x}, \boldsymbol{\alpha}) = \quad & \alpha_1 \cos(\pi x_1) + \alpha_2 \cos(\pi x_2) + \alpha_3 \cos(\pi x_1) \cos(\pi x_2) \\ & + \alpha_4 \cos(2\pi x_1) + \alpha_5 \cos(2\pi x_2) + \alpha_6 \cos(2\pi x_1) \cos(\pi x_2) \\ & + \alpha_7 \cos(\pi x_1) \cos(2\pi x_2) + \alpha_8 \cos(2\pi x_1) \cos(2\pi x_2). \end{aligned} \tag{5.17}$$

Here α_9 determines the angle of the dominant advection direction, while parameters α_i, $i = 1, \dots, 8$, introduce perturbations into the advection field. The parameter domain is a 9D box, see Section 5.3.1 for details.

The system (5.13)–(5.15) is discretized using P_2 finite elements on a grid with either $M = 1,893$ or $M = 4,797$ nodes (depending on a particular experiment) using the standard nodal basis functions $\{\theta_j(\mathbf{x})\}_{j=1}^M$ that define the mass $\mathrm{M} \in \mathbb{R}^{M \times M}$, stiffness $\mathrm{K} \in \mathbb{R}^{M \times M}$ and advection $\mathrm{H}(\boldsymbol{\alpha}) \in \mathbb{R}^{M \times M}$ matrices, and the source vector $\mathbf{f} \in \mathbb{R}^M$. The vector-valued function of nodal values $\mathbf{u}(t, \boldsymbol{\alpha}) : [0, T) \times \mathcal{A} \to \mathbb{R}^M$ solves

$$\mathrm{M}\mathbf{u}_t + (\mathrm{K} + \mathrm{H}(\boldsymbol{\alpha}))\,\mathbf{u} = \mathbf{f}, \tag{5.18}$$

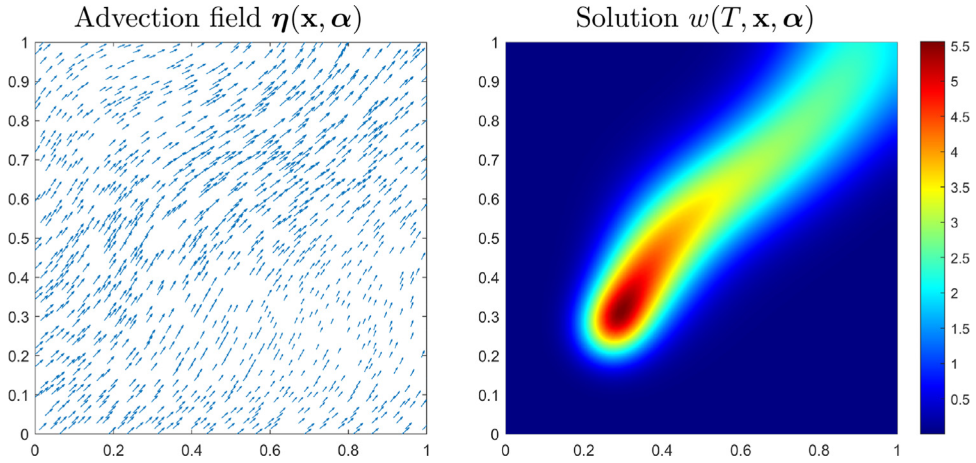

Fig. 5.4. Advection field (5.16) (left) and the solution $w(T, \mathbf{x}, \boldsymbol{\alpha})$ of (5.13)–(5.15) (right) corresponding to a random realization of $\boldsymbol{\alpha}$ from $\mathcal{B}$ and $\nu = 0.01$.

i.e., it satisfies the dynamical system of the form (2.1) with

$$F(t, \mathbf{u}, \boldsymbol{\alpha}) = -\mathrm{M}^{-1}\left(\mathrm{K} + \mathrm{H}(\boldsymbol{\alpha})\right)\mathbf{u} + \mathrm{M}^{-1}\mathbf{f}, \tag{5.19}$$

and the initial condition $\mathbf{u}(0, \boldsymbol{\alpha}) = \mathbf{0} \in \mathbb{R}^M$.

Similarly to the experiments in Section 5.2, we compute the snapshots $\boldsymbol{\phi}_k = \mathbf{u}(t_k, \boldsymbol{\alpha})$ by time-stepping (5.18) at $t_k = (1/30)k$, $k = 1, 2, \ldots, N$, with $N = 30$ time steps and $T = 1$ using Crank–Nicolson scheme. Then, the physical solution snapshots have the form (5.7). The setting is illustrated in Fig. 5.4 where we display the advection field for a random realization of $\boldsymbol{\alpha} \in \mathcal{A}$ and the corresponding solution $w(T, \mathbf{x}, \boldsymbol{\alpha})$.

Projection ROM of (5.18) is obtained similarly to that of (5.6) using the matrix of reduced basis vectors $\mathrm{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{R}^{M\times n}$. Specifically,

$$\widetilde{\mathrm{M}}\widetilde{\mathbf{u}}_t + \left(\widetilde{\mathrm{K}} + \widetilde{\mathrm{H}}(\boldsymbol{\alpha})\right)\widetilde{\mathbf{u}} = \widetilde{\mathbf{f}}, \tag{5.20}$$

where $\widetilde{\mathrm{M}}$ and $\widetilde{\mathrm{K}}$ are defined as in Section 5.2, whereas $\widetilde{\mathrm{H}}(\boldsymbol{\alpha}) = \mathrm{Z}^T\mathrm{H}(\boldsymbol{\alpha})\mathrm{Z} \in \mathbb{R}^{n\times n}$, $\widetilde{\mathbf{f}} = \mathrm{Z}^T\mathbf{f} \in \mathbb{R}^n$, and the initial condition is $\widetilde{\mathbf{u}}(0, \boldsymbol{\alpha}) = \mathbf{0} \in \mathbb{R}^n$. Efficient evaluation of (5.20) was discussed in Section 3.10. Solving (5.20) for $\widetilde{\mathbf{u}}(t, \boldsymbol{\alpha})$ allows to compute the approximate solution snapshots $\widetilde{w}(t_k, \mathbf{x}, \boldsymbol{\alpha})$ exactly as in (5.9).

5.3.1. Out-of-sample TROM performance: advection–diffusion equation

The testing of TROMs for out-of-sample parameters for the advection–diffusion system (5.13)–(5.15) is performed similarly to that for the heat equation in Section 5.2.3. In particular, we compute the average over $\mathcal{A}$ of the relative $L^\infty(0, T; L^2(\Omega))$ and $L^2(0, T; H^1(\Omega))$ errors for $\widetilde{w}$ and use the statistical behavior of the TROM vs. POD gain (5.12) to compare the three TROM variants to conventional POD-ROM.

First, we test TROM in the following setting. The flow behavior is balanced between advection and diffusion, with a diffusion coefficient $\nu = 0.1$ and $M = 1893$. The parameter domain is $\mathcal{A} = [-0.05, 0.05]^8 \times [0.1\pi, 0.3\pi]$ sampled at a Cartesian grid with $K = 3^8 \times 9 = 59{,}049$ points to obtain the sampling set $\widehat{\mathcal{A}}$. We draw $N_r = 200$ out-of-sample realizations $\boldsymbol{\alpha}^{(r)}$ from $\mathcal{A}$ with each $\alpha_i^{(r)}$, $i = 1, \ldots, 9$, distributed uniformly in its corresponding interval. The averaged relative $L^\infty(0, T; L^2(\Omega))$ finite element error of HOSVD-TROM is evaluated as $N_r^{-1}\sum_{r=1}^{N_r} R_{\mathrm{HOSVD}}(\boldsymbol{\alpha}^{(r)})$ with $R_{\mathrm{HOSVD}}(\boldsymbol{\alpha})$ defined in (5.11). The average relative $L^2(0, T; H^1(\Omega))$ finite element error is computed in the same way after modifying $R_{\mathrm{HOSVD}}(\boldsymbol{\alpha})$ accordingly. The TROM vs. POD gain statistic was defined in (5.12).

For such large values of K and therefore large $\boldsymbol{\Phi}$, $\#(\boldsymbol{\Phi}) = 3.3534\mathrm{e}{+}09$, the algorithm we use for finding CP decomposition turns out to be the most memory-intensive and runs out of memory. Thus, in what follows we only report the results for HOSVD- and TT-TROM approaches.

We present in Table 5.6 the averaged relative error of the HOSVD-TROM finite element solutions (for TT-TROM the errors were very close and so are skipped) and the behavior of the gain statistics when tensor compression error

Table 5.6
Tensor compression ranks, averaged relative error of the HOSVD-TROM finite element solutions, statistics of the gain (5.12) for $\nu = 0.1$, $K = 3^8 \times 9 = 59,049$.

	$n = 10$, $\widetilde{\varepsilon} = 10^{-5}$			$n = 12$, $\widetilde{\varepsilon} = 10^{-6}$			$n = 13$, $\widetilde{\varepsilon} = 10^{-7}$		
HOSVD ranks	[78,3,3,3,3,3, 3,3,3,6,11]			[116,3,3,3,3,3, 3,3,3,7,12]			[153,3,3,3,3,3, 3,3,3,9,13]		
TT ranks	[77,99,117,121,121, 107,85,62,37,11]			[116,162,204,219,218, 186,139,91,50,12]			[179,251,330,364,362, 298,209,126,63,14]		
TROM FE error									
$L^\infty(0,T;L^2(\Omega))$	1.15e−03			8.36e−04			7.95e−04		
$L^2(0,T;H^1(\Omega))$	9.86e−04			9.46e−04			9.14e−04		
G_X	mean	std	min	mean	std	min	mean	std	min
HOSVD	**9.17**	5.87	3.11	**11.32**	10.42	1.16	**10.69**	9.06	1.28
TT	**9.13**	5.85	3.11	**11.40**	10.61	1.17	**10.65**	9.06	1.22

$\widetilde{\varepsilon}$ decreases, while simultaneously increasing n to be slightly less than or equal to Tucker rank $\widetilde{N}$ for HOSVD or compression rank $\widetilde{r}_{D+1}$ for TT, respectively. We observe in Table 5.6 a relatively weak dependence of the errors and TROM vs. POD gain mean on the choice of $\widetilde{\varepsilon}$ and n, hence, we conclude that higher tensor compression error and smaller n are more beneficial, since they correspond to higher compression factors and possible faster run times for the offline stage of TROM algorithms.

For the second example we choose an advection-dominated flow with a smaller diffusion coefficient $\nu = 0.01$ and $M = 4,797$. The parameter domain is $\mathcal{A} = [-0.01, 0.01]^8 \times [0.1\pi, 0.5\pi]$ sampled on a Cartesian grid with $K = 20 \times 2^8 = 5,120$ points to obtain $\widehat{\mathcal{A}}$. This gives the snapshot tensor with $\#(\boldsymbol{\Phi}) = 1.2280\text{e+}09$ entries. We draw $N_r = 100$ out-of-sample realizations $\boldsymbol{\alpha}^{(r)}$ from $\mathcal{A}$ with each $\alpha_i^{(r)}$, $i = 1, \dots, 9$, distributed uniformly in its corresponding interval.

Table 5.7 shows tensor compression ranks, number of elements passed to the online stage, the averaged relative error of the HOSVD-TROM finite element solutions, the relative gain statistics for three different levels of tensor compression error $\widetilde{\varepsilon} = 10^{-3}, 10^{-5}, 10^{-7}$ with three different reduced space dimensions n for each case. We observe that it is possible to achieve performance that is very close to the best one with $\widetilde{\varepsilon}$ as large as 10^{-3}, provided n is large enough. Similarly to the results for $\nu = 0.1$, we suggest that for the given problem, discretization and parameter sampling, the finite element error is dominated by the interpolation error of the TROM and using a tighter compression threshold or larger n does not lead to more accurate ROM solutions. It seems beneficial to use low-accuracy tensor decompositions to save on both the computation and storage, while not losing much in terms of relative gain compared to more expensive options. As expected, the TT format becomes more cost-efficient for the higher parameter space dimension D.

Overall, while the accuracy increase of HOSVD- and TT-TROM over POD-ROM is still substantial in the advection–diffusion setting with $D = 9$ parameters, it is smaller than the one for the heat equation considered in Section 5.2. This is most probably caused by larger variability of the snapshots with respect to parameter variation making the problem a good candidate for non-interpolatory TROM (not studied here).

6. Conclusions

Summarizing the findings of the paper, the tensorial projection ROM for parametric dynamical systems builds on several new ideas:
(i) To approximately represent the set of observed snapshots, it uses low-rank tensor formats, rather than a truncated SVD of the snapshot matrix. The corresponding tensor decompositions provide POD-type universal basis while preserving information about solution variation with respect to parameters.
(ii) This additional information is used to find a local (parameter-specific) ROM basis for any incoming parameter that is not necessarily from the training/sampling set.
(iii) The local basis can be represented by its coordinates in the universal low-dimensional basis allowing an effective split of the ROM evaluation between the online and offline phases.

An interpolation procedure was suggested to extract the information about parameter dependence of the solutions, and thus of the ROM spaces, from the low-rank tensor decompositions. Online stage uses fast linear algebra with

Table 5.7

Tensor compression ranks, number of elements passed to the online stage, averaged relative error of the HOSVD-TROM finite element solutions, and statistics of relative gain (5.12) for $\nu = 0.01$, $K = 20 \times 2^8 = 5120$.

$\widetilde{\varepsilon} = 10^{-3}$									
HOSVD ranks	[76,2,2,2,2,2,2,2,2,11,12]						$\#(\text{online}(\widetilde{\boldsymbol{\Phi}})) = 2\,568\,444$		
TT ranks	[75,77,79,76,77,75,71,67,61,11]						$\#(\text{online}(\widetilde{\boldsymbol{\Phi}})) = 100\,747$		
n	5			8			10		
TROM FE error									
$L^\infty(0,T;L^2(\Omega))$	3.45e−2			7.07e−3			5.58e−3		
$L^2(0,T;H^1(\Omega))$	5.59e−2			1.10e−2			5.86e−3		
G_X	mean	std	min	mean	std	min	mean	std	min
HOSVD	**6.95**	1.10	5.41	**22.56**	6.97	12.02	**32.66**	24.70	10.00
TT	**6.95**	1.09	5.41	**22.54**	6.94	11.97	**31.83**	23.58	9.99
$\widetilde{\varepsilon} = 10^{-5}$									
HOSVD ranks	[184,2,2,2,2,2,2,2,2,15,18]						$\#(\text{online}(\widetilde{\boldsymbol{\Phi}})) = 12\,718\,412$		
TT ranks	[183,235,288,305,319,295,246,187,128,18]						$\#(\text{online}(\widetilde{\boldsymbol{\Phi}})) = 1\,110\,964$		
n	5			10			15		
TROM FE error									
$L^\infty(0,T;L^2(\Omega))$	3.45e−2			5.56e−3			5.51e−3		
$L^2(0,T;H^1(\Omega))$	5.59e−2			5.80e−3			5.08e−3		
G_X	mean	std	min	mean	std	min	mean	std	min
HOSVD	**6.95**	1.10	5.41	**33.34**	25.90	10.01	**19.45**	16.33	5.60
TT	**6.95**	1.10	5.41	**33.34**	25.90	10.01	**19.45**	16.33	5.60
$\widetilde{\varepsilon} = 10^{-7}$									
HOSVD ranks	[476,2,2,2,2,2,2,2,2,18,25]						$\#(\text{online}(\widetilde{\boldsymbol{\Phi}})) = 54\,835\,592$		
TT ranks	[528,618,753,858,866,725,520,341,211,25]						$\#(\text{online}(\widetilde{\boldsymbol{\Phi}})) = 6\,975\,287$		
n	5			10			15		
TROM FE error									
$L^\infty(0,T;L^2(\Omega))$	3.44e−2			5.56e−3			5.51e−3		
$L^2(0,T;H^1(\Omega))$	5.59e−2			5.80e−3			5.08e−3		
G_X	mean	std	min	mean	std	min	mean	std	min
HOSVD	**6.95**	1.10	5.41	**33.34**	25.90	10.01	**19.45**	16.33	5.60
TT	**6.95**	1.10	5.41	**33.34**	25.90	10.01	**19.45**	16.33	5.60

complexity depending only on the compression ranks. Non-interpolatory or hybrid approaches are also possible and in fact can produce even more accurate and robust TROMs. We will study these options elsewhere. For interpolatory TROMs, Theorem 4.1 proves an estimate on the representation power of the local ROM bases. Numerical experiment with parameterized heat equation supported the estimate and illustrated the role of each of its terms.

Three popular compressed tensor formats were considered to represent the low-rank tensor in the TROM. Of course, other low-rank tensor decompositions can be used within the general framework of TROM. Out of the three tested, we found HOSVD to be most user-friendly and cost-efficient provided either the dimension of the parameter space is not too large or no Cartesian structure is exploited in organizing the snapshots. Otherwise, TT-TROM provides necessary tools to handle higher-dimensional parameter spaces. We also observed that the accuracy of TROMs crucially depend on n, $\widetilde{\varepsilon}$ and parameter domain sampling, but not as much on the particular low-rank tensor format employed.

Finally, for higher-dimensional parameter spaces a grid-based sampling of the parameter domain becomes prohibitively expensive in terms of offline computation costs. Significant offline costs also incur for problems with less smooth dependence of solution on parameters, which would require a denser sampling, and for problems where each high-fidelity solve is expensive because of fine spatial or temporal resolution. We see several ways to develop TROMs addressing these challenges: (i) use a sophisticated sampling, e.g., based on a greedy strategy, and organize the snapshots in 3D tensors, (ii) to benefit from Cartesian structure and higher order tensor decompositions, apply a tensor completion method to find a low-rank representation of the snapshot tensor sampled at a few nodes of the parameter grid, and (iii) combine TROMs with compressed formats to represent high-fidelity snapshots. We leave these options for a future research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

M.O. was supported in part by the U.S. National Science Foundation under awards DMS-2011444 and DMS-1953535. This material is based upon research supported in part by the U.S. Office of Naval Research under award number N00014-21-1-2370 to A.M. The authors thank Vladimir Druskin, Traian Iliescu, and Vladimir Kazeev for their comments on the first draft of this paper.

References

[1] A.C. Antoulas, D.C. Sorensen, S. Gugercin, A survey of model reduction methods for large-scale systems, Tech. Rep., 2000.

[2] S. Gugercin, A.C. Antoulas, A survey of model reduction by balanced truncation and some new results, Internat. J. Control 77 (8) (2004) 748–766.

[3] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM Rev. 57 (4) (2015) 483–531.

[4] J.S. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, Vol. 590, Springer, 2016.

[5] T. Bui-Thanh, K. Willcox, O. Ghattas, Model reduction for large-scale systems with high-dimensional parametric input space, SIAM J. Sci. Comput. 30 (6) (2008) 3270–3288.

[6] U. Baur, C. Beattie, P. Benner, S. Gugercin, Interpolatory projection methods for parameterized model reduction, SIAM J. Sci. Comput. 33 (5) (2011) 2489–2518.

[7] P. Benner, L. Feng, A robust algorithm for parametric model order reduction based on implicit moment matching, in: Reduced Order Methods for Modeling and Computational Reduction, Springer, 2014, pp. 159–185.

[8] S.L. Brunton, J.L. Proctor, J.N. Kutz, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. 113 (15) (2016) 3932–3937.

[9] J.L. Lumley, The structure of inhomogeneous turbulent flows, Atmospheric Turbulence Radio Wave Propag. (1967).

[10] L. Sirovich, Turbulence and the dynamics of coherent structures. I. Coherent structures, Quart. Appl. Math. 45 (3) (1987) 561–571.

[11] S. Chaturantabut, D.C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM J. Sci. Comput. 32 (5) (2010) 2737–2764.

[12] C.W. Rowley, Model reduction for fluids, using balanced proper orthogonal decomposition, Int. J. Bifurcation Chaos 15 (03) (2005) 997–1013.

[13] F. Chinesta, A. Ammar, E. Cueto, Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models, Arch. Comput. Methods Eng. 17 (4) (2010) 327–350.

[14] F. Chinesta, R. Keunings, A. Leygue, The Proper Generalized Decomposition for Advanced Numerical Simulations: A Primer, Springer Science & Business Media, 2013.

[15] J.L. Eftang, A.T. Patera, E.M. Rønquist, An “hp” certified reduced basis method for parametrized elliptic partial differential equations, SIAM J. Sci. Comput. 32 (6) (2010) 3170–3200.

[16] J.L. Eftang, D.J. Knezevic, A.T. Patera, An hp certified reduced basis method for parametrized parabolic partial differential equations, Math. Comput. Model. Dyn. Syst. 17 (4) (2011) 395–422.

[17] D. Amsallem, M.J. Zahr, C. Farhat, Nonlinear model order reduction based on local reduced-order bases, Internat. J. Numer. Methods Engrg. 92 (10) (2012) 891–916.

[18] D. Amsallem, C. Farhat, Interpolation method for adapting reduced-order models and application to aeroelasticity, AIAA J. 46 (7) (2008) 1803–1813.

[19] N.T. Son, A real time procedure for affinely dependent parametric model order reduction using interpolation on grassmann manifolds, Internat. J. Numer. Methods Engrg. 93 (8) (2013) 818–833.

[20] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (3) (2009) 455–500.

[21] L. Grasedyck, D. Kressner, C. Tobler, A literature survey of low-rank tensor approximation techniques, GAMM-Mitt. 36 (1) (2013) 53–78.

[22] N.D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E.E. Papalexakis, C. Faloutsos, Tensor decomposition for signal processing and machine learning, IEEE Trans. Signal Process. 65 (13) (2017) 3551–3582.

[23] A. Nouy, Low-rank tensor methods for model order reduction, 2015, arXiv preprint arXiv:1511.01555.

[24] A. Nouy, Low-rank methods for high-dimensional approximation and model order reduction, in: P. Benner, A. Cohen, M. Ohlberger, K. Willcox (Eds.), Model Reduction and Approximation, SIAM, Philadelphia, PA, 2017, pp. 171–226.

[25] P. Benner, A. Onwunta, M. Stoll, Low-rank solution of unsteady diffusion equations with stochastic coefficients, SIAM/ASA J. Uncertainty Quantif. 3 (1) (2015) 622–649.

[26] P. Benner, S. Dolgov, A. Onwunta, M. Stoll, Low-rank solvers for unsteady Stokes–Brinkman optimal control problem with random data, Comput. Methods Appl. Mech. Engrg. 304 (2016) 26–54.

[27] P. Benner, S. Dolgov, A. Onwunta, M. Stoll, Solving optimal control problems governed by random Navier-Stokes equations using low-rank methods, 2017, arXiv preprint arXiv:1703.06097.

[28] K. Lee, H.C. Elman, B. Sousedik, A low-rank solver for the Navier–Stokes equations with uncertain viscosity, SIAM/ASA J. Uncertain. Quantif. 7 (4) (2019) 1275–1300.

[29] D. Kressner, C. Tobler, Low-rank tensor Krylov subspace methods for parametrized linear systems, SIAM J. Matrix Anal. Appl. 32 (4) (2011) 1288–1316.

[30] S. Kastian, D. Moser, L. Grasedyck, S. Reese, A two-stage surrogate model for Neo-Hookean problems based on adaptive proper orthogonal decomposition and hierarchical tensor approximation, Comput. Methods Appl. Mech. Engrg. 372 (2020) 113368.

[31] G. Kerschen, J.-c. Golinval, A.F. Vakakis, L.A. Bergman, The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview, Nonlinear Dynam. 41 (1) (2005) 147–169.

[32] M. Rathinam, L.R. Petzold, A new look at proper orthogonal decomposition, SIAM J. Numer. Anal. 41 (5) (2003) 1893–1925.

[33] Y. Liang, W. Lin, H. Lee, S. Lim, K. Lee, H. Sun, Proper orthogonal decomposition and its applications—Part II: Model reduction for MEMS dynamical analysis, J. Sound Vib. 256 (3) (2002) 515–532.

[34] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, C. Wu, Proper orthogonal decomposition and its applications—Part I: Theory, J. Sound Vib. 252 (3) (2002) 527–544.

[35] F.L. Hitchcock, The expression of a tensor or a polyadic as a sum of products, J. Math. Phys. 6 (1–4) (1927) 164–189.

[36] J.D. Carroll, J.-J. Chang, Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition, Psychometrika 35 (3) (1970) 283–319.

[37] H.A. Kiers, Towards a standardized notation and terminology in multiway analysis, J. Chemom. J. Chemom. Soc. 14 (3) (2000) 105–122.

[38] J. Håstad, Tensor rank is NP-complete, J. Algorithms 11 (4) (1990) 644–654.

[39] V. De Silva, L.-H. Lim, Tensor rank and the ill-posedness of the best low-rank approximation problem, SIAM J. Matrix Anal. Appl. 30 (3) (2008) 1084–1127.

[40] R.A. Harshman, Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis, 1970.

[41] L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM J. Matrix Anal. Appl. 21 (4) (2000) 1253–1278.

[42] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311.

[43] I.V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput. 33 (5) (2011) 2295–2317.

[44] I. Oseledets, E. Tyrtyshnikov, TT-cross approximation for multidimensional arrays, Linear Algebra Appl. 432 (1) (2010) 70–88.

[45] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2012) 208–220.

[46] S. Gandy, B. Recht, I. Yamada, Tensor completion and low-n-rank tensor recovery via convex optimization, Inverse Problems 27 (2) (2011) 025010.

[47] B. Huang, C. Mu, D. Goldfarb, J. Wright, Provable low-rank tensor recovery, Optim. Online 4252 (2) (2014) 455–500.

[48] M. Yuan, C.-H. Zhang, On tensor completion via nuclear norm minimization, Found. Comput. Math. 16 (4) (2016) 1031–1068.

[49] J.A. Bengua, H.N. Phien, H.D. Tuan, M.N. Do, Efficient tensor completion for color image and video recovery: Low-rank tensor train, IEEE Trans. Image Process. 26 (5) (2017) 2466–2479.

[50] S. Brenner, L. Scott, The Mathematical Theory of Finite Element Methods, second edn., Springer, New York, 2002.

[51] M.A. Grepl, Y. Maday, N.C. Nguyen, A.T. Patera, Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations, ESAIM Math. Model. Numer. Anal. 41 (3) (2007) 575–605.

[52] M. Drohmann, B. Haasdonk, M. Ohlberger, Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation, SIAM J. Sci. Comput. 34 (2) (2012) A937–A969.

[53] K. Carlberg, C. Farhat, J. Cortial, D. Amsallem, The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows, J. Comput. Phys. 242 (2013) 623–647.

[54] B. Kramer, K.E. Willcox, Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition, AIAA J. 57 (6) (2019) 2297–2307.

[55] P. Hartman, Ordinary Differential Equations, Vol. 590, John Wiley and Sons, 1964.

[56] W. Hackbusch, Tensor Spaces and Numerical Tensor Calculus, Vol. 42, Springer, 2012.

[57] M. Griebel, H. Harbrecht, Analysis of tensor approximation schemes for continuous functions, Found. Comput. Math. (2021) 1–22.

[58] L. Trefethen, Multivariate polynomial approximation in the hypercube, Proc. Amer. Math. Soc. 145 (11) (2017) 4837–4844.

[59] R. Schneider, A. Uschmajew, Approximation rates for the hierarchical tensor format in periodic Sobolev spaces, J. Complexity 30 (2) (2014) 56–71.

[60] W. Hackbusch, B.N. Khoromskij, Tensor-product approximation to operators and functions in high dimensions, J. Complexity 23 (4–6) (2007) 697–714.

[61] V.N. Temlyakov, Estimates for the best bilinear approximations of periodic functions, Trudy Matematicheskogo Inst. Imeni Steklova 181 (1988) 250–267.

[62] B.N. Khoromskij, C. Schwab, Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs, SIAM J. Sci. Comput. 33 (1) (2011) 364–385.

[63] A. Cohen, R. Devore, C. Schwab, Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDE's, Anal. Appl. 9 (01) (2011) 11–47.

[64] J. Ballani, D. Kressner, M.D. Peters, Multilevel tensor approximation of PDEs with random data, Stoch. Partial Differ. Equ. Anal. Comput. 5 (3) (2017) 400–427.

[65] M. Eigel, M. Pfeffer, R. Schneider, Adaptive stochastic Galerkin FEM with hierarchical tensor representations, Numer. Math. 136 (3) (2017) 765–803.

[66] S.V. Dolgov, V.A. Kazeev, B.N. Khoromskij, Direct tensor-product solution of one-dimensional elliptic equations with parameter-dependent coefficients, Math. Comput. Simulation 145 (2018) 136–155.

[67] M.A. Grepl, A.T. Patera, A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations, ESAIM Math. Model. Numer. Anal. 39 (1) (2005) 157–181.

[68] A.T. Patera, G. Rozza, et al., Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations, 2007.