Mathematical and Computational Applications

MDPI

Article

An Algorithmic Comparison of the Hyper-Reduction and the Discrete Empirical Interpolation Method for a Nonlinear Thermal Problem

Felix Fritzen [1,*], Bernard Haasdonk [2], David Ryckelynck [3] and Sebastian Schöps [4]

[1] EMMA—Efficient Methods for Mechanical Analysis, Institute of Applied Mechanics (CE), University of Stuttgart, Pfaffenwaldring 7, 70569 Stuttgart, Germany

[2] Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany; haasdonk@mathematik.uni-stuttgart.de

[3] MAT-Centre des Matériaux, MINES ParisTech, PSL Research University, CNRS UMR 7633, BP 87, 91003 Evry, France; david.ryckelynck@mines-paristech.fr

[4] Graduate School of Computational Engineering, Technische Universität Darmstadt, Dolivostraße 15, 64293 Darmstadt, Germany; schoeps@temf.tu-darmstadt.de

* Correspondence: felix.fritzen@mechbau.uni-stuttgart.de; Tel.: +49-(0)711/685-66283

Received: 19 December 2017; Accepted: 7 February 2018; Published: 13 February 2018

Abstract: A novel algorithmic discussion of the methodological and numerical differences of competing parametric model reduction techniques for nonlinear problems is presented. First, the Galerkin reduced basis (RB) formulation is presented, which fails at providing significant gains with respect to the computational efficiency for nonlinear problems. Renowned methods for the reduction of the computing time of nonlinear reduced order models are the Hyper-Reduction and the (Discrete) Empirical Interpolation Method (EIM, DEIM). An algorithmic description and a methodological comparison of both methods are provided. The accuracy of the predictions of the hyper-reduced model and the (D)EIM in comparison to the Galerkin RB is investigated. All three approaches are applied to a simple uncertainty quantification of a planar nonlinear thermal conduction problem. The results are compared to computationally intense finite element simulations.

Keywords: model order reduction (MOR); reduced basis model order reduction (RB MOR); uncertainty quantification (UQ); (discrete) empirical interpolation method (EIM; DEIM); hyper-reduction (HR)

1. Introduction

Numerical models in engineering or natural sciences are getting more and more complex, may be nonlinear and depending on unknown or controllable design-parameters. Simultaneously, simulation settings increasingly move from single-forward simulations to higher-level simulation scenarios. For example, optimization and statistical investigations require multiple solves, interactive applications require real-time simulation response, or slim-computing environments, e.g., simple controllers, require rapid and memory-saving models. For such applications, the field of model reduction has gained increasing attention during the last decade. Their goal is an acceleration of a given numerical model based on the construction of a low-dimensional approximate surrogate model, the so-called reduced order model. Due to the reduced dimension, the computation should ideally be rapid, hence be applicable for the mentioned multi-query, real-time or slim-computing simulation scenarios. Well-known techniques for linear problems comprise Proper Orthogonal Decomposition (POD) [1,2], control-theoretic approaches such as Balanced Truncation, Moment Matching or Hankel-norm approximation [3]. For parametric problems, certified Reduced Basis (RB) methods have been developed [4,5]. Nonlinear problems pose additional challenges. In particular, there exists a well-known drawback with POD, which is that a high-dimensional reconstruction of the reduced solution is required for each evaluation of the nonlinearity. Several approximation techniques exist, which provide a remedy of this problem. Mainly, these approaches are sampling-based techniques such as Empirical Interpolation [6] and discrete variants [7–10] or Hyper-Reduction [11,12]. Note that also further approaches exist, such as Gappy-POD [13], Missing Point Estimation (MPE) [14] or Gauss–Newton with approximated Tensors (GNAT) [15] and Energy Conserving Mesh Sampling and Weighting (ECSW) [16]. Most of those methods identify a subset of the arguments of the nonlinear function. Then, based solely on an evaluation of these few components, they construct an approximation of the full solution.

Nonlinear problems emerge in many applications. For instance, the effective behavior of dissipative microstructured materials needs to be predicted by nonlinear homogenization techniques. These imply a multi-query context in the sense of different loadings (and load paths) applied to the reference volume element in order to obtain the related effective mechanical response. Reduced basis methods combining the purely algorithmic gains of the reduced basis with a reformulation of the problem incorporating micromechanics have shown to be efficient for the prediction of the effective material response, for multi-level FE simulations and for nonlinear multiscale topology optimization (e.g., [17,18]).

In this paper, we analyze and compare two of those efficient methods, namely Discrete Empirical Interpolation (DEIM) and Hyper-Reduction (HR), for the reduction of a non-trivial nonlinear parametric thermal model. The comparison is carried out on the numerical, algorithmical and mathematical level. A new condition for which DEIM and HR are equivalent is given and numerical examples illustrate the application. Similar studies, often with stronger emphasis on the performance, have also been recently conducted by other authors, e.g., [19].

The paper is structured as follows. We introduce the nonlinear parametrized thermal model problem in Section 2. Subsequently, in Section 3, the methods under investigation are introduced and formally compared: as a benchmark, the POD–Galerkin procedure is formulated, then the DEIM and the hyper-reduction technique. The numerical comparison is provided in Section 4. A classical scenario from uncertainty quantification is demonstrated since model order reduction is particularly interesting for such many-query scenarios. Finally, we conclude in Section 5.

1.1. Nomenclature

In the manuscript, the following notation is used: bold face symbols denote vectors (lowercase letters) or matrices (uppercase letters). The spatial gradient operator is denoted by $\nabla\bullet$ and the divergence is expressed by $\nabla\cdot\bullet$. The dependence on spatial coordinates is omitted for simplicity of notation.

2. Nonlinear Reference Problem

2.1. Strong Formulation

In the following, we consider a stationary planar nonlinear heat conduction problem on a plate with a hole (see Figure 1). The problem is parametrized by a vector $\boldsymbol{p}$. The nonlinearity of the problem is introduced by an isotropic Fourier-type heat conductivity $\mu(u;\boldsymbol{p})$ that depends on the current temperature u and the parameter vector. The Dirichlet and Neumann boundary data are denoted by u_* and q_*, respectively. Note that, in the example, $q_* = 0$ is considered. However, the formulation of the weak form is considered with q_* being arbitrary for the sake of generality. The corresponding Dirichlet and Neumann boundaries are denoted by Γ_1 and Γ_2, respectively. The strong formulation of the boundary value problem is

$$-\nabla\cdot(\mu(u;\boldsymbol{p})\,\nabla u) = 0 \text{ in } \Omega, \qquad u = u_*(\boldsymbol{p}) \text{ on } \Gamma_1, \qquad -\mu(u;\boldsymbol{p})\,\nabla u\cdot\boldsymbol{n} = q_* \text{ on } \Gamma_2, \tag{1}$$

where μ is the temperature dependent conductivity.

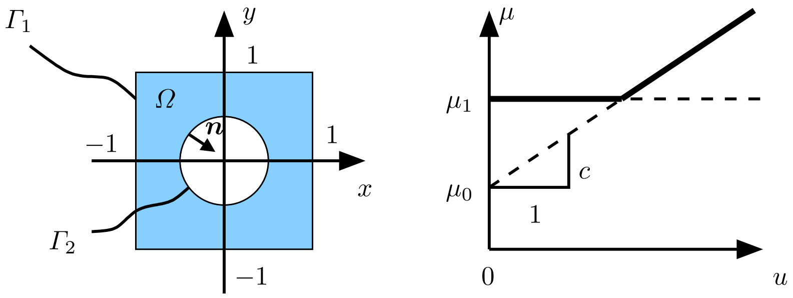


Figure 1. Geometry of the planar benchmark problem (**left**) and nonlinearity of the temperature dependent conductivity μ (**right**).

2.2. Weak Formulation

The weak form of the heat conduction problem (1) is given by

$$a(u, \delta u; \boldsymbol{p}) - l(\delta u; \boldsymbol{p}) = 0 \qquad (\forall \delta u \in \mathcal{V}_0), \tag{2}$$

where

$$a(u, \delta u; \boldsymbol{p}) = \int_{\Omega} \mu(u; \boldsymbol{p}) \, \nabla u \cdot \nabla \delta u \, \mathrm{d}\Omega, \quad \text{and} \qquad l(\delta u; \boldsymbol{p}) = -\int_{\Gamma_2} \delta u \, q_* \, \mathrm{d}\Gamma, \tag{3}$$

with the unknown temperature field $u \in \mathcal{V} = \mathcal{V}_0 + \{\bar{u}_*(\boldsymbol{p})\}$ being sought-after. The function space $\mathcal{V}_0 := \{u \in \mathcal{H}^1(\Omega) : u = 0 \text{ on } \Gamma_1\}$ is referred to as space of test functions vanishing on the Dirichlet boundary Γ_1 and $\bar{u}_*(\boldsymbol{p}) \in \mathcal{H}^1(\Omega)$ is a field defined on the full domain that satisfies the Dirichlet conditions. The Dirichlet conditions on Γ_1 are assumed to depend on two parameters g_x and g_y via

$$u_*(\boldsymbol{p}) := g_\mathrm{x} x + g_\mathrm{y} y, \quad g_\mathrm{x}, g_\mathrm{y} \in [0, 1], \quad x, y \in \Gamma_1. \tag{4}$$

A trivial choice is to set $\bar{u}_*(\boldsymbol{p}) = g_\mathrm{x} x + g_\mathrm{y} y$ in the full domain. This particular choice is considered in the remainder of this study. While the solution space $\mathcal{V}$ depends on the parameters $\boldsymbol{p}$ via $\bar{u}_*$, the space of test functions $\mathcal{V}_0$ is independent of the parameters $\boldsymbol{p}$. For the heat conductivity μ, an explicit dependence on the temperature via the nonlinear constitutive model

$$\mu(u; \boldsymbol{p}) := \max\{\mu_1, \mu_0 + cu\} \tag{5}$$

is assumed (see also Figure 1, right). The parameter vector $\boldsymbol{p}$ in the present context is

$$\boldsymbol{p} := [g_\mathrm{x}, g_\mathrm{y}, c, \mu_0, \mu_1], \tag{6}$$

and attention will be confined to the compact parameter domain

$$\boldsymbol{p} \in \mathcal{P} := [0, 1] \times [0, 1] \times [1, 2] \times \{1\} \times \{0.5\} \subset \mathbb{R}^5. \tag{7}$$

2.3. Discrete Formulation

In order to solve the nonlinear problem (1), nodal Finite Elements (FE) are used in a classical Galerkin formulation (e.g., [20,21]). The space of finite element test functions $\mathcal{V}_0^\mathrm{h} \subset \mathcal{V}_0$ is assumed to be spanned by n linearly independent and continuous ansatz functions φ_i associated with nodes (x_i, y_i) and $\varphi_i(x_j, y_j) = \delta_{ij}$ where $i, j = 1, \ldots, n$. We assume a solution expansion into a linear field $\bar{u}_*$ defined over the full domain Ω via (4) and comply to the parameterized boundary data and a fluctuation term according to

$$u_h(\boldsymbol{p}) := \bar{u}_*(\boldsymbol{p}) + \sum_{j=1}^{n} w_{h,j}(\boldsymbol{p})\varphi_j, \qquad \bar{u}_*(\boldsymbol{p}) := g_{\mathrm{x}} x + g_{\mathrm{y}} y \quad \forall x, y \in \Omega. \tag{8}$$

Here, the coefficient vector $\boldsymbol{w}(\boldsymbol{p}) = \left(w_{h,j}(\boldsymbol{p})\right)_{j=1}^{n} \in \mathbb{R}^n$ contains the nodal temperature fluctuations, which represent the unknowns of the system. They define the coefficient vector containing the nodal temperatures via

$$\boldsymbol{u}(\boldsymbol{w}; \boldsymbol{p}) = \boldsymbol{w}(\boldsymbol{p}) + \bar{\boldsymbol{u}}_*(\boldsymbol{p}), \tag{9}$$

where $\bar{\boldsymbol{u}}_*(\boldsymbol{p})$ is the vector composed of the nodal values of the superimposed linear field $\bar{u}_*(\boldsymbol{p})$. The discrete nonlinear equations that have to be solved for any given parameter vector $\boldsymbol{p}$ are given by

$$r_i(\boldsymbol{w}; \boldsymbol{p}) := a\left(\bar{u}_*(\boldsymbol{p}) + \sum_{j=1}^{n} w_{h,j}\varphi_j,\ \varphi_i; \boldsymbol{p}\right) - l(\varphi_i; \boldsymbol{p}) = 0 \quad \left(\forall i \in \{1, \ldots, n\}\right). \tag{10}$$

In our specific numerical example, the linear functional $l(\varphi_i; \boldsymbol{p})$ is zero due to the chosen homogeneous Neumann conditions. However, problem (10) still has a nontrivial solution due to the inhomogeneous Dirichlet data provided through $\bar{u}_*(\boldsymbol{p})$. Condition (10) can be expressed in a compact representation using the vector notation

$$\boldsymbol{r}(\boldsymbol{w}; \boldsymbol{p}) := \left(r_i(\boldsymbol{w}; \boldsymbol{p})\right)_{i=1}^{n} = \boldsymbol{0}. \tag{11}$$

In order to solve the nonlinear problem (11), expressed component-wise in (10), the finite element method is used to provide the functions for the expansion (8). In the following, $\boldsymbol{N}$ denotes a row vector of finite element ansatz functions for the temperature, and $\boldsymbol{G}$ and $\boldsymbol{G}^T$ are the discrete gradient and divergence matrices, respectively. The FE approximation of the temperature u and its gradient ∇u are given by

$$u_h(\boldsymbol{w}; \boldsymbol{p}) = \boldsymbol{N}\boldsymbol{u}(\boldsymbol{w}; \boldsymbol{p}) = \boldsymbol{N}\boldsymbol{w}(\boldsymbol{p}) + \boldsymbol{N}\bar{\boldsymbol{u}}_*(\boldsymbol{p}), \tag{12}$$

$$\nabla u_h(\boldsymbol{w}; \boldsymbol{p}) = \boldsymbol{G}\boldsymbol{u}(\boldsymbol{w}; \boldsymbol{p}) = \boldsymbol{G}\boldsymbol{w}(\boldsymbol{p}) + \bar{\boldsymbol{g}}(\boldsymbol{p}), \qquad \bar{\boldsymbol{g}}(\boldsymbol{p}) = \begin{bmatrix} g_{\mathrm{x}} \\ g_{\mathrm{y}} \end{bmatrix}. \tag{13}$$

Then, the exact Jacobian of the finite element system is

$$\boldsymbol{J}_*(\boldsymbol{w}; \boldsymbol{p}) = \int_{\Omega} \mu(u_h(\boldsymbol{w}; \boldsymbol{p}); \boldsymbol{p})\, \boldsymbol{G}^T\boldsymbol{G} + \frac{\partial \mu(u_h(\boldsymbol{w}; \boldsymbol{p}); \boldsymbol{p})}{\partial u_h} \left(\boldsymbol{G}^T\boldsymbol{G}\boldsymbol{u}(\boldsymbol{w}; \boldsymbol{p})\right)\boldsymbol{N}\, \mathrm{d}\Omega, \tag{14}$$

where the first part is the classical finite element stiffness matrix and the second part accounts for the thermal sensitivity of the conductivity μ. In the following, a symmetric approximation of $\boldsymbol{J}_*$ given by

$$\boldsymbol{J}(\boldsymbol{w}; \boldsymbol{p}) = \int_{\Omega} \mu(u_h(\boldsymbol{w}; \boldsymbol{p}); \boldsymbol{p})\, \boldsymbol{G}^T\boldsymbol{G}\, \mathrm{d}\Omega \tag{15}$$

is used, i.e., the thermal sensitivity is neglected. The numerical solution of the nonlinear problem (11) is then obtained by a fixed-point scheme using $\boldsymbol{J}$ as an approximation of the differential stiffness matrix $\boldsymbol{J}_*$. The resulting iteration scheme is commonly referred to as successive substitution (Ref. [22], p. 66). Note also that $\boldsymbol{J}$ is well defined for arbitrary thermal fields, while the exact Jacobian is not defined, or more precisely the Jacobian is semi-smooth, for the critical temperature $u_{\mathrm{c}} = (\mu_1 - \mu_0)/c$, which implies a non-differentiability of the conductivity μ.

Algorithm 1 illustrates the nonlinear finite element solution procedure for the considered problem. The comment lines in the algorithm comprise references to computational costs c_{reloc} (computation of u and ∇u), c_{const} (for the constitutive model), c_{rhs} (linked to the residual computation), c_{Jac} (related to the Jacobian) and c_{sol} (for the linear solver). This will become relevant in the comparison section.

Algorithm 1: Finite Element Solution

```
Input :parameters $\boldsymbol{p} \in \mathcal{P}$, scatter/gather operator $\boldsymbol{P}_e$ of element $e$ and weight $v_i$ at the
       integration point $\boldsymbol{x}_i$ $(i = 1, \ldots, n_{\mathrm{gp}})$
Output: nodal temperature vector $\boldsymbol{u}(\boldsymbol{p})$
1  set $\boldsymbol{u}^{(0)} = \bar{\boldsymbol{u}}_*(\boldsymbol{p}); \alpha = 0$ ;                          // initialize
2  set $\boldsymbol{r} = \boldsymbol{0}, \boldsymbol{J} = \boldsymbol{0}$ ;                                // reset r.h.s. and Jacobian
3  for $e = 1, \ldots, n_{\mathrm{el}}$ do                              // $n_{\mathrm{el}}$: number of elements
4      set $\boldsymbol{r}_e = \boldsymbol{0}, \boldsymbol{J}_e = \boldsymbol{0}$ ;                     // reset element r.h.s. and Jacobian
5      for $i = 1, \ldots, n_e^{\mathrm{gp}}$ do                 // $n_e^{\mathrm{gp}}$: number of int. point in element $e$
6          evaluate the FE matrices $\boldsymbol{N}, \boldsymbol{G}$ and $\boldsymbol{G}_e = \boldsymbol{G}\boldsymbol{P}_e$ at the current int. point;    // $c_{\mathrm{reloc}}$
7          compute $u_h^{(\alpha)} \leftarrow \boldsymbol{N}\boldsymbol{u}^{(\alpha)}$ and gradient $\boldsymbol{g}_h^{(\alpha)} \leftarrow \boldsymbol{G}\boldsymbol{u}^{(\alpha)}$ at int. point ;      // $c_{\mathrm{reloc}}$
8          evaluate constitutive model $\mu^{(\alpha)} \leftarrow \mu(u_h^{(\alpha)}; \boldsymbol{p})$ ;                  // $c_{\mathrm{const}}$
9          $\boldsymbol{r}_e \leftarrow \boldsymbol{r}_e + v_i\, \mu^{(\alpha)}\, \boldsymbol{G}_e{}^T \boldsymbol{g}_h^{(\alpha)}$ ;                                   // $c_{\mathrm{rhs}}$
10         $\boldsymbol{J}_e \leftarrow \boldsymbol{J}_e + v_i\, \mu^{(\alpha)}\, \boldsymbol{G}_e{}^T \boldsymbol{G}_e$ ;                                    // $c_{\mathrm{Jac}}$
11     $\boldsymbol{r} \leftarrow \boldsymbol{r} + \boldsymbol{P}_e^T \boldsymbol{r}_e; \boldsymbol{J} \leftarrow \boldsymbol{J} + \boldsymbol{P}_e^T \boldsymbol{J}_e \boldsymbol{P}_e$ ;                     // assembly; $c_{\mathrm{rhs}}, c_{\mathrm{Jac}}$
12 solve $\boldsymbol{J}\delta\boldsymbol{u}^{(\alpha)} = -\boldsymbol{r}$ and update nodal temperatures $\boldsymbol{u}^{(\alpha+1)} \leftarrow \boldsymbol{u}^{(\alpha)} + \delta\boldsymbol{u}^{(\alpha)}; \alpha \leftarrow \alpha + 1$ ; // $c_{\mathrm{sol}}$
13 converged ($\|\delta u_h^{(\alpha)}\|_{L^2(\Omega)} < \epsilon_{\max}$)? $\rightarrow$ end;   else: goto 2
```

2.4. Reduced Basis Ansatz

In the following, an m-dimensional basis of global ansatz functions ψ_k is considered in the reduced framework, where $m \ll n$ is implicitly asserted in order to obtain an actual model order reduction. The functions ψ_k define a linear subspace of $\mathcal{V}_0^{\mathrm{h}}$, which can be characterized by a matrix $\boldsymbol{V} = (V_{ij}) \in \mathbb{R}^{n \times m}$ via

$$\psi_k := \sum_{i=1}^{n} V_{ik} \varphi_i \qquad (k = 1, \ldots, m). \tag{16}$$

The reduced coefficient vector $\boldsymbol{\gamma}(\boldsymbol{p}) = (\gamma_k(\boldsymbol{p}))_{k=1}^m \in \mathbb{R}^m$ defines the temperature fluctations in the reduced setting by the affine relation

$$\tilde{u}(\boldsymbol{p}) = \tilde{u}(\boldsymbol{\gamma}(\boldsymbol{p}); \boldsymbol{p}) := u_*(\boldsymbol{p}) + \tilde{w}(\boldsymbol{p}), \qquad \tilde{w}(\boldsymbol{p}) := \sum_{k=1}^{m} \gamma_k(\boldsymbol{p})\, \psi_k. \tag{17}$$

The corresponding vector of discrete values $\tilde{\boldsymbol{u}}(\boldsymbol{p}) := (\tilde{u}_j(\boldsymbol{p}))_{j=1}^n \in \mathbb{R}^n$ of the temperature with respect to the basis of $\mathcal{V}_0^{\mathrm{h}}$ is given by the linear relation $\tilde{\boldsymbol{u}} = \bar{\boldsymbol{u}}_*(\boldsymbol{p}) + \tilde{\boldsymbol{w}}(\boldsymbol{p})$ with the coefficient vector $\tilde{\boldsymbol{w}}(\boldsymbol{p}) = \boldsymbol{V}\boldsymbol{\gamma}(\boldsymbol{p}) = (\tilde{w}(\boldsymbol{p}))_{j=1}^n$ of the reduced fluctuation field. Hence, the reduced fluctuation $\tilde{w}$ can equivalently be expressed as

$$\tilde{w}(\boldsymbol{p}) = \sum_{j=1}^{n} \tilde{w}_j(\boldsymbol{p}) \varphi_j = \sum_{j=1}^{n} \varphi_j \sum_{k=1}^{m} V_{jk} \gamma_k(\boldsymbol{p}). \tag{18}$$

In the following, the matrix $\boldsymbol{V}$ defining the space of the reduced ansatz functions is assumed to be given by a snapshot POD [23] obtained from full resolution finite element simulations evaluated at s different parameter vectors $\boldsymbol{p}^{(i)}$ $(i = 1, \ldots, s)$. The POD subspace $\tilde{\mathcal{V}}_0^{\mathrm{h}} \subset \mathcal{V}_0^{\mathrm{h}}$ is then obtained as best-approximating m-dimensional subspace with respect to the $L^2(\Omega)$-error, i.e.,

$$\tilde{\mathcal{V}}_0^{\mathrm{h}} = \operatorname*{arg\,min}_{\substack{\tilde{\mathcal{V}} \subset \mathcal{V}_0^{\mathrm{h}} \\ \dim(\tilde{\mathcal{V}})=m}} \sum_{i=1}^{s} \left\| w_h(\boldsymbol{p}^{(i)}) - P_{\tilde{\mathcal{V}}}(w_h(\boldsymbol{p}^{(i)}) \right\|_{L^2(\Omega)}^2, \tag{19}$$

where $P_{\tilde{\mathcal{V}}}$ is the orthogonal projection of the thermal fluctuations onto the subspace $\tilde{\mathcal{V}}$. Numerically, a POD-basis of this space can be obtained by a corresponding matrix eigenvalue problem, e.g., cf. [1,5,24]. Here, we use the $L^2(\Omega)$ inner product for the computation of the symmetric snapshot inner product matrix

$$\boldsymbol{C} = (C_{ij})_{i,j=1}^{s} \in \mathbb{R}^{s \times s}, \qquad C_{ij} := \int_{\Omega} w_h(\boldsymbol{p}^{(i)}) w_h(\boldsymbol{p}^{(j)}) \, \mathrm{d}\Omega. \tag{20}$$

Note that the correlation matrix is computed from the temperature fluctuations w_h only in order to obtain a reduced basis that will comply with the prescribed Dirichlet conditions that are accounted for in $\bar{u}_*(\boldsymbol{p})$. The entries of $\boldsymbol{C}$ are defined by the discrete snapshot solutions $\boldsymbol{w}^{(i)}$ via the unit mass matrix $\boldsymbol{M}$ as follows:

$$\boldsymbol{M} := \int_{\Omega} \boldsymbol{N}^T \boldsymbol{N} \, \mathrm{d}\Omega, \qquad C_{ij} = \left(\boldsymbol{w}^{(i)}\right)^T \boldsymbol{M} \left(\boldsymbol{w}^{(j)}\right). \tag{21}$$

A pseudo-code implementation of the snapshot POD is given by Algorithm 2. Alternatively, a weighted SVD of the snapshot matrix can be used to obtain the same basis. The classical SVD w.r.t. the l^2-norm may be less prone to truncation errors [25], but it will yield a result differing from the optimum in (19), which is based on optimality with respect to $\| \cdot \|_{L^2(\Omega)}$. This is due to the missing consideration of the inner product matrix $\boldsymbol{M}$.

Algorithm 2: Snapshot Proper Orthogonal Decomposition (Snapshot POD)

```
Input  : $s$ snapshot parameters $\boldsymbol{p}^{(i)} \subseteq \mathcal{P}$; reduced dimension $m$ (or: threshold $\delta$);
         unit mass matrix $\boldsymbol{M}$
Output: reduced basis functions $\psi_k$ $(k = 1, \ldots, m)$ and coefficient matrix $\boldsymbol{V}$
for $i = 1, \ldots, s$ do
    solve high-fidelity problem (e.g., via FE solution; Algorithm 1) for parameter $\boldsymbol{p}^{(i)} \rightarrow \boldsymbol{w}^{(i)}$
    store $\boldsymbol{w}^{(i)}$ as $i$-th column of the snapshot matrix $\boldsymbol{S}$
compute snapshot inner product matrix $\boldsymbol{C} := \boldsymbol{S}^T \boldsymbol{M} \boldsymbol{S}$
compute $m$ leading eigenpairs $(\boldsymbol{q}_k, \xi_k)$ of $\boldsymbol{C}$ $(k = 1, \ldots, m)$
set $V_{jk} := (\xi_k)^{-1/2} \sum_{i=1}^{s} \left(\boldsymbol{w}^{(i)}\right)_j (\boldsymbol{q}_k)_i \quad (j = 1, \ldots, n)$ and $\psi_k = \sum_{j=1}^{n} \varphi_j V_{jk}$
```

There are many other techniques to determine reduced projection spaces, e.g., greedy techniques, balanced truncation, Hankel-norm approximation, moment matching and others (e.g., [3]). However, we do not aim at a comparison of projection space techniques, but rather focus on the treatment of the nonlinearities. Therefore, we decide for the POD space as common reduced projection space for all subsequent methods.

Note that, for the given problem, the Dirichlet data can be considered via the linear field $\bar{u}_*(\boldsymbol{p})$ defined in (4). More precisely, $\bar{u}_*$ is parametrized by two independent parameters $g_{\mathrm{x}}, g_{\mathrm{y}}$. Its discrete representation is

$$\bar{\boldsymbol{u}}_*(\boldsymbol{p}) = g_{\mathrm{x}}\bar{\boldsymbol{u}}_{*,1} + g_{\mathrm{y}}\bar{\boldsymbol{u}}_{*,2}, \tag{22}$$

with the components of the vectors $\boldsymbol{u}_{*,1}, \boldsymbol{u}_{*,2}$ defined as

$$(u_{*,1})_{i=1}^n := x_i, \qquad\qquad (u_{*,2})_{i=1}^n := y_i, \tag{23}$$

where x_i and y_i denote the coordinates of node number i. By accounting for $\bar{\boldsymbol{u}}_*(\boldsymbol{p})$ via (22), the Dirichlet data is exactly represented without further algebraic constraints. Additionally, this approach captures exactly constant temperature gradients.

3. Sampling-Based Reductions

3.1. Galerkin Reduced Basis Approximation

A classical Galerkin projection on a POD space is used to provide reference solutions for the HR and the DEIM. In order to solve the weak form of the nonlinear problem for given parameters $g_{\mathrm{x}}, g_{\mathrm{y}}$, a successive substitution is performed in which the local conductivity μ is computed using the previous iteration of the temperature $\tilde{u}^{(\alpha)} := u_*(\boldsymbol{p}) + \tilde{w}^{(\alpha)}$ with $\tilde{w}^{(\alpha)} = \boldsymbol{N}\boldsymbol{V}\boldsymbol{\gamma}^{(\alpha)}$ in the reduced setting, where $\alpha \in \mathbb{N}$ is the iteration number and $\boldsymbol{\gamma}^{(\alpha)}$ is the iterate of the reduced degrees of freedom in iteration α. As in a classical Galerkin approach, we assume the (reduced) test function

$$\tilde{v} := \sum_{i=1}^{n}\sum_{j=1}^{m} \varphi_i\, V_{ij}\lambda_j \tag{24}$$

with arbitrary coefficients λ_j $(j = 1, \dots, m)$ represented by the vector $\boldsymbol{\lambda}$. For convenience, we define the conductivity corresponding to the α-th iterate $\boldsymbol{\gamma}^{(\alpha)}$ of the reduced coefficient vector $\boldsymbol{\gamma}(\boldsymbol{p})$

$$\mu^{(\alpha)} := \mu\left(\tilde{u}(\boldsymbol{\gamma}^{(\alpha)}; \boldsymbol{p}); \boldsymbol{p}\right). \tag{25}$$

Here, $\boldsymbol{\gamma}^{(\alpha)}$ can be interpreted as a constant parameter similar to $\boldsymbol{p}$ during the subsequent iteration, which provides the new iterate $\boldsymbol{\gamma}^{(\alpha+1)}$ as the solution of

$$a(\tilde{u}, \tilde{v}; \boldsymbol{p}) = \sum_{j=1}^{m} \lambda_j \int_{\Omega} \mu^{(\alpha)}\, \nabla\tilde{u}(\boldsymbol{\gamma}^{(\alpha+1)}; \boldsymbol{p}) \cdot \nabla\psi_j \,\mathrm{d}\Omega = 0 \qquad \forall\boldsymbol{\lambda} \in \mathbb{R}^m. \tag{26}$$

Rewriting (26) using the approximated Jacobian $\boldsymbol{J}$ and the residual $\boldsymbol{r}$ leads to the linear system

$$\boldsymbol{V}^T\boldsymbol{J}(\boldsymbol{V}\boldsymbol{\gamma}^{(\alpha)}; \boldsymbol{p})\boldsymbol{V}(\boldsymbol{\gamma}^{(\alpha+1)} - \boldsymbol{\gamma}^{(\alpha)}) + \boldsymbol{V}^T\boldsymbol{r}(\boldsymbol{V}\boldsymbol{\gamma}^{(\alpha)}; \boldsymbol{p}) = \boldsymbol{0}. \tag{27}$$

The projection of the residual onto the reduced basis defines the reduced residual

$$\tilde{\boldsymbol{r}}^{(\alpha)} := \boldsymbol{V}^T\boldsymbol{r}(\boldsymbol{V}\boldsymbol{\gamma}^{(\alpha)}; \boldsymbol{p}). \tag{28}$$

In view of the definition of the reduced basis by $L^2(\Omega)$-orthogonal POD modes and with $\delta\boldsymbol{\gamma}^{(\alpha)} := \boldsymbol{\gamma}^{(\alpha+1)} - \boldsymbol{\gamma}^{(\alpha)}$, one obtains

$$\|\delta\boldsymbol{\gamma}^{(\alpha)}\|_{l^2} = \|\tilde{u}^{(\alpha+1)} - \tilde{u}^{(\alpha)}\|_{L^2(\Omega)} = \|\tilde{w}^{(\alpha+1)} - \tilde{w}^{(\alpha)}\|_{L^2(\Omega)}. \tag{29}$$

This gives rise to the simple convergence criterion $\|\delta\gamma^{(\alpha)}\|_{l^2} < \epsilon_{\max}$, i.e., the iteration should stop upon sufficiently small changes of the temperature field. Algorithm 3 summarizes the online phase of the Galerkin RB method.

Algorithm 3: Galerkin Reduced Basis Solution (Online Phase)

```
Input  : parameters $\boldsymbol{p} \in \mathcal{P}$; reduced temperature matrix $\boldsymbol{T}_i := \boldsymbol{N}(\boldsymbol{x}_i)\boldsymbol{V}$, gradient matrix
         $\boldsymbol{G}_i := \boldsymbol{G}(\boldsymbol{x}_i)\boldsymbol{V}$ and weight $v_i$ at the integration point $\boldsymbol{x}_i$ $(i = 1, \dots, n_{\mathrm{gp}})$
Output: reduced vector $\gamma$ and nodal temperatures $\tilde{\boldsymbol{u}}$ (optional)
1 set $\gamma^{(0)} = \boldsymbol{0}$; $\alpha = 0$ ;                                       // initialize
2 set $\tilde{\boldsymbol{r}}^{(\alpha)} = \boldsymbol{0}$, $\tilde{\boldsymbol{J}}^{(\alpha)} = \boldsymbol{0}$ ;                        // reset r.h.s. and Jacobian
3 for $i = 1, \dots, n_{\mathrm{gp}}$ do                         // $n_{\mathrm{gp}}$: number of int. points in the mesh
4     compute temperature $\tilde{u}^{(\alpha)} \leftarrow \boldsymbol{T}_i\gamma^{(\alpha)} + u_*$ and gradient $\tilde{\boldsymbol{g}}^{(\alpha)} \leftarrow \boldsymbol{G}_i\gamma^{(\alpha)} + \bar{\boldsymbol{g}}$ ;    // $c_{\mathrm{reloc}}$
5     evaluate constitutive model $\mu^{(\alpha)} \leftarrow \mu(\tilde{u}^{(\alpha)}; \boldsymbol{p})$ ;                  // $c_{\mathrm{const}}$
6     $\tilde{\boldsymbol{r}}^{(\alpha)} \leftarrow \tilde{\boldsymbol{r}}^{(\alpha)} + v_i\,\mu^{(\alpha)}\,\boldsymbol{G}_i^T\tilde{\boldsymbol{g}}^{(\alpha)}$ ;                 // $c_{\mathrm{rhs}}$; direct assembly
7     $\tilde{\boldsymbol{J}}^{(\alpha)} \leftarrow \tilde{\boldsymbol{J}}^{(\alpha)} + v_i\,\mu^{(\alpha)}\,\boldsymbol{G}_i^T\boldsymbol{G}_i$ ;                  // $c_{\mathrm{Jac}}$; direct assembly
8 solve $\tilde{\boldsymbol{J}}^{(\alpha)}\delta\gamma^{(\alpha)} = -\tilde{\boldsymbol{r}}^{(\alpha)}$ and update $\gamma^{(\alpha+1)} \leftarrow \gamma^{(\alpha)} + \delta\gamma^{(\alpha)}$; set $\alpha \leftarrow \alpha + 1$ ;   // $c_{\mathrm{sol}}$
9 converged ($\|\delta\tilde{u}^{(\alpha)}\|_{L^2(\Omega)} = \|\delta\gamma^{(\alpha)}\|_{l^2} < \epsilon_{\max}$)? → end;   else: goto 2
```

The projected system can be interpreted as a finite element method with global, problem specific ansatz functions ψ_k, whereas the classical finite element method uses local and rather general ansatz functions φ_j (e.g., piecewise defined polynomials).

Note that, the solution of (26), (27) is also a minimizer of the potential

$$\Pi(\gamma; \gamma^{(\alpha)}, \boldsymbol{p}) := \frac{1}{2}\int_{\Omega} \mu^{(\alpha)}\,\nabla\tilde{u}(\gamma; \boldsymbol{p}) \cdot \nabla\tilde{u}(\gamma; \boldsymbol{p})\;\mathrm{d}\Omega. \tag{30}$$

Therefore, variational methods can directly be applied to solve the minimization problem and alternative numerical strategies are available. Such variational schemes are also used, e.g., in the context of solid mechanical problems involving internal variables (e.g., [26]).

The Galerkin RB method with a well-chosen reduced basis functions ψ_k (represented via the matrix $\boldsymbol{V}$) can replicate the FEM solution to a high accuracy (see Section 4). It also provides a significant reduction of the memory requirements: instead of $\boldsymbol{u} \in \mathbb{R}^n$, only $\gamma \in \mathbb{R}^m$ needs to be stored. Despite the significant reduction of the number of unknowns from n to m, the Galerkin RB cannot attain substantial accelerations of the nonlinear simulation due to a computationally expensive assembly procedure with complexity $\mathcal{O}(n_{\mathrm{gp}})$ for the residual vector $\boldsymbol{r}$ and for the fixed point operator $\boldsymbol{J}$ (compare c_{rhs} and c_{Jac} in Algorithm 1 and Algorithm 3). Here, n_{gp} is the number of quadrature points in the mesh. However, if the linear systems are not solved with optimal complexity, e.g., using sparse *LU* or Cholesky decompositions with at least $\mathcal{O}(n^2)$, then a reduction of complexity can still be achieved. It shall be pointed out that, for very large n (i.e., for millions of unknowns), the linear solver usually dominates the overall computational expense. Then, the Galerkin RB may provide good accelerations without further modifications.

In order to significantly improve on the computational efficiency while maintaining the reduced number of degrees of freedom (and thus the reduced storage requirements), the Hyper-Reduction [11,12] and the Discrete Empricial Interpolation Method (DEIM, [7–9]) are used. Both methods are specifically designed for the computationally efficient approximation of the nonlinearity of PDEs.

3.2. Discrete Empirical Interpolation Method (DEIM)

The empirical interpolation method (EIM) was introduced by [6] to approximate parametric or nonlinear functions by separable interpolants. This technique is meanwhile standard in the reduced basis methodology for parametric PDEs. Discrete versions of the EIM for instationary problems have been introduced as empirical operator interpolation [7,8,27] or alternatively (and in some cases equivalently) as discrete empirical interpolation (DEIM) [9,10]. In particular, a posteriori [8,27,28] and a priori [10] error control is possible under certain assumptions (see also [5]).

We present a formulation for the present stationary problem. Instead of approximating a continuous field variable, the goal of the discrete versions is to provide an approximation $\tilde{\boldsymbol{r}}$ for the vectorial nonlinearity of the nodal residual vector $\boldsymbol{r}$ of the form

$$\tilde{\boldsymbol{r}}(\boldsymbol{u};\boldsymbol{p}) := \boldsymbol{U}(\boldsymbol{P}^T\boldsymbol{U})^{-1}\boldsymbol{P}^T\boldsymbol{r}(\boldsymbol{u};\boldsymbol{p}), \tag{31}$$

where the columns of $\boldsymbol{U} \in \mathbb{R}^{n\times M}$ are called *collateral reduced basis* and $\boldsymbol{P} = [\boldsymbol{e}_{i_1},\ldots,\boldsymbol{e}_{i_M}] \in \mathbb{R}^{n\times M}$ is a sampling matrix with interpolation indices (also known as *magic points*) $i_1,\ldots,i_M \in \{1,\ldots,n\}$, with $\boldsymbol{e}_i$ being the i-th unit vector. By multiplication of (31) with $\boldsymbol{P}^T$, we verify that

$$(\tilde{\boldsymbol{r}}(\boldsymbol{u};\boldsymbol{p}))_{i_j} = \boldsymbol{r}(\boldsymbol{u};\boldsymbol{p}))_{i_j}, \quad j = 1,\ldots,M. \tag{32}$$

In this sense, the approximation acts as an interpolation within the set of magic points.

The identification of the interpolation points is an incremental procedure, which is performed during the offline phase. We assume the existence of a set of training snapshots $\mathcal{Y} := \{\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{n_{\text{train}}}\} \subset \mathbb{R}^n$ with $\dim(\operatorname{span}(\mathcal{Y})) \geq M$.

Then, a POD of these snapshots results, see e.g., [10], in the collateral basis vectors $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_M$ and we define $\boldsymbol{U}_l := [\boldsymbol{u}_1,\ldots,\boldsymbol{u}_l]$ for $l = 1,\ldots,M$. The algorithm for the point selection is initialized with $\boldsymbol{P}_0 = [\,]$, $I_0 := \emptyset$, $\boldsymbol{U}_0 = [\,]$ and then computes for $l = 1\ldots,M$

$$\boldsymbol{q}_l := \boldsymbol{u}_l - \boldsymbol{U}_{l-1}(\boldsymbol{P}_{l-1}^T\boldsymbol{U}_{l-1})^{-1}\boldsymbol{P}_{l-1}^T\boldsymbol{u}_l, \tag{33}$$

$$i_l := \arg\max_{i\in\{1,\ldots,n\}} |(\boldsymbol{q}_l)_i|, \tag{34}$$

$$\boldsymbol{P}_l := [\boldsymbol{P}_{l-1},\boldsymbol{e}_{i_l}], I_l := I_{l-1} \cup \{i_l\}. \tag{35}$$

Finally, we set $\boldsymbol{P} := \boldsymbol{P}_M$, $\boldsymbol{U} := \boldsymbol{U}_M$ and $I := I_M$, which concludes the construction of $\tilde{\boldsymbol{r}}$. Intuitively, in each iteration, the interpolation error $\boldsymbol{q}_l$ for the current POD basis vector $\boldsymbol{u}_l$ is determined and the vector entry i_l with maximum absolute value is identified, which gives the next index. Regularity of the matrix $\boldsymbol{P}^T\boldsymbol{U}$ is required for a well-defined interpolation. This condition is automatically satisfied under the aforementioned assumption of a sufficiently rich set of snapshots $\mathcal{Y}$. As training set $\mathcal{Y}$, one can either use samples of the nonlinearity [8], or use snapshots of the state vector or combinations thereof. In contrast to the instationary case, we may not use only training snapshots of $\boldsymbol{r}$: As the residual $\boldsymbol{r}$ is zero for all snapshots, we would try to find a basis for a zero-dimensional space $\operatorname{span}(\mathcal{Y}) = 0$, which is not possible for $M > 0$. However, the residual at the intermediate (non-equilibrium) iterates is non-zero and this is also a good target quantity for the (D)EIM, as these terms appear on the right-hand side of the linear system during the fixed point iteration. Hence, a reasonable set $\mathcal{Y}$ is obtained via

$$\begin{aligned}\mathcal{Y} = [\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{n_{\text{train}}}] = [&\boldsymbol{r}(\boldsymbol{u}^{(0)}(\boldsymbol{p}^{(1)});\boldsymbol{p}^{(1)}),\ldots,\boldsymbol{r}(\boldsymbol{u}^{(\alpha_1)}(\boldsymbol{p}^{(1)});\boldsymbol{p}^{(1)}),\\ &\ldots,\boldsymbol{r}(\boldsymbol{u}^{(\alpha_s)}(\boldsymbol{p}^{(s)});\boldsymbol{p}^{(s)})],\end{aligned} \tag{36}$$

where $\alpha_1,\ldots,\alpha_s$ are the number of fixed point iterations of the full simulation scheme for parameters $\boldsymbol{p}^{(1)},\ldots,\boldsymbol{p}^{(s)}$.

Inserting $\tilde{\boldsymbol{r}}$ from (31) for the nonlinearity into the full problem and projection by left-multiplication with a weight matrix $\boldsymbol{W} \in \mathbb{R}^{n \times m}$ yields the POD-DEIM reduced m-dimensional nonlinear system for the unknown γ

$$\boldsymbol{W}^T\boldsymbol{U}(\boldsymbol{P}^T\boldsymbol{U})^{-1}\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{u}}_*(\boldsymbol{p})+\boldsymbol{V}\gamma;\boldsymbol{p}) = 0. \tag{37}$$

This low-dimensional nonlinear problem is iteratively solved by a fixed point procedure, i.e., at the current approximation $\gamma^{(\alpha)}$, we solve the linearized problem for $\delta\gamma^{(\alpha)}$

$$\boldsymbol{W}^T\boldsymbol{U}(\boldsymbol{P}^T\boldsymbol{U})^{-1}\boldsymbol{P}^T\boldsymbol{J}(\bar{\boldsymbol{u}}_*(\boldsymbol{p})+\boldsymbol{V}\gamma^{(\alpha)};\boldsymbol{p})\boldsymbol{V}\delta\gamma^{(\alpha)} = -\boldsymbol{W}^T\boldsymbol{U}(\boldsymbol{P}^T\boldsymbol{U})^{-1}\boldsymbol{P}^T\boldsymbol{r}(\bar{\boldsymbol{u}}_*(\boldsymbol{p})+\boldsymbol{V}\gamma^{(\alpha)};\boldsymbol{p}) \tag{38}$$

and set $\gamma^{(\alpha+1)} := \gamma^{(\alpha)} + \delta\gamma^{(\alpha)}$. As in the previous sections, if $M < m$, this linear system cannot be solved uniquely. In that case, an alternative would be to solve a residual least-squares problem, similar to the GNAT-procedure, cf. [15]. Note that the assembly of this system does not involve any high-dimensional operations, as the product of the first four matrices on the left- and right-hand side can be precomputed as a small matrix $\boldsymbol{X} := \boldsymbol{W}^T\boldsymbol{U}(\boldsymbol{P}^T\boldsymbol{U})^{-1}$. Then, the terms $\boldsymbol{P}^T\boldsymbol{J}, \boldsymbol{P}^T\boldsymbol{r}$ also do not require any high-dimensional operations, as the multiplication with $\boldsymbol{P}^T(\cdot) = (\cdot)_I$ corresponds to evaluation of the "magic" rows of the Jacobian and right-hand side, respectively. Typically, in discretized PDEs, these M rows only depend on few $\bar{M}$ entries of the unknown variable (e.g., the DOFs related to neighboring elements). This number $\bar{M}$ is typically bounded by a certain multiple of M due to regularity constraints on the mesh [8].

In (38), what is required for the collateral basis $\boldsymbol{U}$ can be recognized: If $\boldsymbol{J}(\boldsymbol{V}\gamma;\boldsymbol{p})\boldsymbol{V}\delta\gamma \in \operatorname{colspan}(\boldsymbol{U})$ and $\boldsymbol{r}(\boldsymbol{V}\gamma;\boldsymbol{p}) \in \operatorname{colspan}(\boldsymbol{U})$, then we are exactly solving the Galerkin–POD reduced linearized system

$$\boldsymbol{W}^T\boldsymbol{J}(\boldsymbol{V}\gamma;\boldsymbol{p})\boldsymbol{V}\delta\gamma = -\boldsymbol{W}^T\boldsymbol{r}(\boldsymbol{V}\gamma;\boldsymbol{p}). \tag{39}$$

Hence, this gives a guideline for an alternative reasonable choice of the training set $\mathcal{Y}$, namely consisting of snapshots of both (columns of) $\boldsymbol{J}$ or $\boldsymbol{J}\boldsymbol{V}$ and $\boldsymbol{r}$.

In the Galerkin projection case, one can choose $\boldsymbol{W} = \boldsymbol{V}$. This is the choice that we pursue in the experiments to make the procedure more similar to the other reduction approaches. The offline phase of the DEIM is summarized in Algorithm 4 and an algorithm of the online phase is provided in Algorithm 5.

Algorithm 4: Offline Phase of the Discrete Empirical Interpolation Method (DEIM)

```
Input  : collateral basis functions $\boldsymbol{u}_k$ (and related matrix $\boldsymbol{U}$) from POD of snapshots $\mathcal{Y}$
Output: sampling matrix $\boldsymbol{P}$; magic point index set $I := \{i_1, \ldots, i_M\}$;
         reduced basis $(\hat{\boldsymbol{q}}_l)_{l=1,\ldots,M}$ (optional)
set $\boldsymbol{P}_0 \leftarrow [\,], \boldsymbol{U}_0 = [\,], I_0 := \emptyset$ ;                      // initialize
for $l = 1 \ldots, M$ do
    $\boldsymbol{U}_l \leftarrow [\boldsymbol{u}_1, \ldots \boldsymbol{u}_l]$ ;                                  // truncated POD matrix
    $\boldsymbol{q}_l \leftarrow \boldsymbol{u}_l - \boldsymbol{U}_{l-1}(\boldsymbol{P}_{l-1}^T\boldsymbol{U}_{l-1})^{-1}\boldsymbol{P}_{l-1}^T\boldsymbol{u}_l$ ;         // interpolation residual
    $i_l \leftarrow \arg\max_{i\in\{1,\ldots,n\}} |(\boldsymbol{q}_l)_i|$ ;                        // maximum of residual
    $\hat{\boldsymbol{q}}_l \leftarrow \boldsymbol{q}_l / (\boldsymbol{q}_l)_{i_l}$ ;                                    // normalization
    $\boldsymbol{P}_l \leftarrow [\boldsymbol{P}_{l-1}, \boldsymbol{e}_{i_l}],\ I_l := I_{l-1} \cup \{i_l\}$ ;          // extend projection/magic points
set $\boldsymbol{P} := \boldsymbol{P}_M, I := I_M$.
```

Algorithm 5: Online Phase of the Discrete Empirical Interpolation Method (DEIM)

```
Input  : parameters $\boldsymbol{p} \in \mathcal{P}$ reduced basis $\boldsymbol{V}$, POD-DEIM sampling matrix
         $\boldsymbol{X} := \boldsymbol{W}^T\boldsymbol{U}(\boldsymbol{P}^T\boldsymbol{U})^{-1}$ and magic point index set $I$
Output : reduced vector $\gamma$ and nodal temperatures $\tilde{\boldsymbol{u}}$ (optional)
1 set $\gamma^{(0)} = \mathbf{0}$; $\tilde{\boldsymbol{u}}^{(0)} = \bar{\boldsymbol{u}}_*(\boldsymbol{p})$; $\alpha = 0$ ;                  // initialize
2 $\bar{\boldsymbol{J}} \leftarrow (\boldsymbol{J}(\tilde{\boldsymbol{u}}^{(\alpha)};\boldsymbol{p})\boldsymbol{V})_I$ ;     // $c_{\mathrm{Jac}}$; evaluate $M$ rows of right-projected Jacobian
3 $\bar{\boldsymbol{r}} \leftarrow (\boldsymbol{r}(\tilde{\boldsymbol{u}}^{(\alpha)};\boldsymbol{p}))_I$ ;          // $c_{\mathrm{rhs}}$; evaluate $M$ rows of right hand side
4 solve $\boldsymbol{X}\bar{\boldsymbol{J}}\delta\gamma^{(\alpha)} = -\boldsymbol{X}\bar{\boldsymbol{r}}$ ;                // $c_{\mathrm{sol}}$; fixpoint iter. for $\delta\gamma^{(\alpha)}$
5 update $\gamma^{(\alpha+1)} \leftarrow \gamma^{(\alpha)} + \delta\gamma^{(\alpha)}$;                             // update
6 compute $\boldsymbol{P}_{\bar{M}}\tilde{\boldsymbol{u}}^{(\alpha+1)} \leftarrow \boldsymbol{P}_{\bar{M}}\tilde{\boldsymbol{u}}^{(\alpha)} + \boldsymbol{P}_{\bar{M}}\boldsymbol{V}\delta\gamma^{(\alpha)}$ and set $\alpha \leftarrow \alpha + 1$ ;   // $c_{\mathrm{reloc}}$
7 converged ($\|\delta\tilde{u}^{(\alpha)}\|_{L^2(\Omega)} = \|\delta\gamma^{(\alpha)}\|_{l^2} < \epsilon_{\max}$)? → end;   else: goto 2
```

3.3. Hyper-Reduction (HR)

In order to improve the numerical efficiency, the Hyper-Reduction method [11] introduces a Reduced Integration Domain (RID) denoted $\Omega_Z \subset \Omega$. The RID depends on the reduced basis. It is constructed by offline algebraic operations. The hyper-reduced equations are a Petrov–Galerkin formulation of the equilibrium equations, obtained by using truncated test functions having zero values outside the RID. The vector form of the reduced equations is similar to the one obtained by the Missing Point Estimation method [14] proposed for the Finite Volume Method. The strength of the Hyper-Reduction is its ability to reduce mechanical models in material science while keeping the formulation of the constitutive equations unchanged [12]. The smaller the RID, the lower the computational complexity and the higher the approximation errors. These points have been developed in previous papers dealing with various mechanical problems (e.g., [29,30]).

The offline procedure of the Hyper-Reduction method involves two steps. The first step is the construction of the Reduced Integration Domain Ω_Z. For the present benchmark test, the RID is the union of a subdomain denoted by Ω^u generated from the reduced vector gradients $(\nabla\psi_k)_{k=1,\ldots,m}$, and a domain denoted by Ω^+ corresponding to a set of neighboring elements to the previous subdomain. Usually, in the Hyper-reduction method, the user can select an additional subdomain of Ω in order to extend the RID over a region of interest. This subdomain is denoted by Ω_{user}. In the sequel, to get small RIDs, Ω_{user} is empty. The set Ω^u consists of aggregated contributions $\Omega^u_k, k = 1, \ldots, m$ from all the reduced vectors:

$$\Omega_Z := \Omega^u \cup \Omega^+ \cup \Omega_{user}, \qquad\qquad \Omega^u := \cup_{k=1}^m \Omega^u_k. \tag{40}$$

To give the full details of the procedure, we introduce the domain partition in finite elements denoted $(\Omega^{\mathrm{e}}_j \subset \Omega)_{j=1,\ldots n_{\mathrm{el}}}$: $\Omega = \cup_{j=1}^{n_{\mathrm{el}}} \Omega^{\mathrm{e}}_j$, where n_{el} is the number of elements in the mesh. The domain Ω^u_k is the element where the maximum $L^2(\Omega)$ norm of the reduced vectors $\nabla\widetilde{\psi}_k$ is reached. In [11], $\nabla\widetilde{\psi}_k$ was set equal to $\nabla\psi_k$. Here, when applying the DEIM to $(\nabla\psi_k)_{k=1,\ldots,m}$, the interpolation residuals provide a new reduced basis $(\boldsymbol{q}_k)_{k=1,\ldots,m}$ (cf. Algorithm 6) related to temperature gradients. In this paper, $\nabla\widetilde{\psi}_k$ is the output reduced basis produced by the DEIM, when it is applied to $(\nabla\psi_k)_{k=1,\ldots,m}$. Other procedures, for the RID construction, are available in previous papers on hyper-reduction, e.g., [11,12]. The element selection reads for $k = 1, \ldots, m$:

$$\Omega^u_k = \underset{\Omega^{\mathrm{e}}_j, j=1,\ldots,n_{\mathrm{el}}}{\arg\max} \left\|\nabla\widetilde{\psi}_k\right\|_{L^2(\Omega^{\mathrm{e}}_j)}, \tag{41}$$

where $\|.\|_{L^2(\Omega^{\mathrm{e}}_j)}$ is the L^2 norm restricted to the element Ω^{e}_j. Several layers of surrounding elements denoted Ω^+ can be added to Ω^u.

The second step of the offline Hyper-Reduction procedure is the generation of truncated test functions that are zero outside of the RID. The truncation operator P_Z is defined for all $u_h \in \mathcal{V}_0^h$ by

$$P_Z(u_h) := \sum_{i\in\mathcal{F}} \varphi_i \, u_{h,i}, \quad \mathcal{F} = \Big\{ i \in \{1,\ldots,n\} \,\Big|\, \int\limits_{\Omega\setminus\Omega_Z} \varphi_i \, \varphi_i \, \mathrm{d}\Omega = 0 \Big\}. \tag{42}$$

Here, $\mathcal{F}$ is the set of indices of internal points, i.e., inner FE nodes, in Ω_Z, which are related to the available FE equilibrium for predictions, which are forecasted only over Ω_Z, i.e., for all $w \in \mathcal{V}_0^h$ holds with $\Gamma_2^Z := \Gamma_2 \cap \partial\Omega_Z$

$$a(u_h, P_Z(w); \boldsymbol{p}) - l(P_Z(w)) = \int\limits_{\Omega_Z} \mu(u_h; \boldsymbol{p}) \, \nabla u_h \cdot \nabla(P_Z(w)) \, \mathrm{d}\Omega - \int\limits_{\Gamma_2^Z} P_Z(w) q_* \, \mathrm{d}\Gamma. \tag{43}$$

The operator P_Z can be represented by a truncated projector denoted $\boldsymbol{Z}$. More precisely, if $\mathcal{F} = \{i_1, i_2, \ldots, i_l\}$ with $l := \mathrm{card}(\mathcal{F})$, then

$$\boldsymbol{Z} := [\boldsymbol{e}_{i_1}, \boldsymbol{e}_{i_2}, \ldots, \boldsymbol{e}_{i_l}] \in \mathbb{R}^{n\times l}, \qquad P_Z(u_h) := \sum_{i=1}^{n} \varphi_i \, (\boldsymbol{Z}\,\boldsymbol{Z}^T \, \boldsymbol{u})_i \tag{44}$$

with $\boldsymbol{e}_i \in \mathbb{R}^n$ the i-th unit vector. Therefore, the hyper-reduced form of the linearized prediction step is: for a given $\gamma^{(\alpha)}$, find $\delta\gamma^{(\alpha)}$ such that,

$$\boldsymbol{V}^T \, \boldsymbol{Z}\,\boldsymbol{Z}^T \, \boldsymbol{J}(\boldsymbol{V}\gamma^{(\alpha)}; \boldsymbol{p}) \, \boldsymbol{V} \, \delta\gamma^{(\alpha)} = -\boldsymbol{V}^T \, \boldsymbol{Z}\,\boldsymbol{Z}^T \, \boldsymbol{r}(\boldsymbol{V}\gamma^{(\alpha)}; \boldsymbol{p}), \tag{45}$$

where $\boldsymbol{J}$ is given by (15) and $\gamma^{(\alpha+1)} := \gamma^{(\alpha)} + \delta\gamma^{(\alpha)}$.

In addition to $\boldsymbol{Z}$, we introduce also the operator $\bar{\boldsymbol{Z}} \in \mathbb{R}^{n\times\bar{l}}$ that is a truncated projection operator onto the $\bar{l}$ points contained in the RID. In practice, the discrete unknowns are computed at these $\bar{l} \geq l$ points in order to compute the residual at the inner points l. Note that often $\bar{l}$ is significantly larger than l, especially if the RID consists of disconnected (scattered) regions.

The complexity of the products related to the fixed point operator $\boldsymbol{J}$ on the left-hand side term scale with $2\zeta l m + 2 l m^2$, where ζ is the maximum number of non-zero entries per row of $\boldsymbol{J}$. For the right-hand side, the computational complexity is $2lm$. For both products, the complexity reduction factors are n/l. To obtain a well-posed hyper-reduced problem, one requires to fulfill the following condition $l \geq m$. If this condition is not fulfilled, the linear system of Equation (45) is rank deficient. In case of rank deficiency, one has to add more surrounding elements to the RID. The closer l to m, and the lower m, the less complex is the solution of the hyper-reduced formulation. The RID construction must generate a sufficiently large RID. If not, the convergence can be hampered, the number of correction steps can be increased and, moreover, the accuracy of the prediction can suffer. When $\Omega_Z = \Omega$, then $\boldsymbol{Z}$ is the identity matrix and the hyper-reduced formulation coincides with the usual system obtained by the Galerkin projection. An a posteriori error estimator for hyper-reduced approximations has recently been proposed in [31] for generalized standard materials.

The offline phase of the hyper-reduction is summarized in Algorithm 6 and an algorithm of the online phase is provided in Algorithm 7.

3.4. Methodological Comparison

We comment on some formal commonalities and differences between the HR and the (D)EIM.

We first note that both methods reproduce the Galerkin–POD case, if $l = M = n$. For the HR, this means that the RID is the full domain, which implies that $\boldsymbol{Z}$ is a square permutation matrix, hence being invertible and yielding $\boldsymbol{Z}\boldsymbol{Z}^T = \boldsymbol{I}$, thus (45) reduces to the POD–Galerkin reduced system (27). For the (D)EIM, this implies that the magic points consist of all grid points. We similarly obtain that $\boldsymbol{P}$ and $\boldsymbol{U}$ are invertible and thus $\boldsymbol{U}(\boldsymbol{P}^T\boldsymbol{U})^{-1}\boldsymbol{P}^T = \boldsymbol{I}$ and (38) also reproduces the POD–Galerkin reduced system (27).

```
Algorithm 6: Offline Phase of the Hyper-Reduction (HR)
  Input  : POD basis functions $\psi_k$ (and related matrix $\boldsymbol{V}$) from snapshot POD, global matrix
           $\boldsymbol{G} \in \mathbb{R}^{(n_{\mathrm{gp}}D)\times n}$ ($D$ : space dimension) for the gradient of the FE ansatz functions,
           the number $\ell$ of additional layers of elements used to extend the RID and the
           user-defined subdomain $\Omega_{user} \subset \Omega$
  Output: reduced integration domain $\Omega_Z$; truncated identities $\boldsymbol{Z}, \bar{\boldsymbol{Z}}$
 1  set $\boldsymbol{U} = \boldsymbol{G}\boldsymbol{V}, \widetilde{\boldsymbol{U}} = [\,], \boldsymbol{P} = [\,]$ ;      // initialize gradient at all int. points from RB
    // $\boldsymbol{u}_k \in \mathbb{R}^{n_{\mathrm{gp}}D}$ is a column vector containing $\nabla\psi_k$ at all $n_{\mathrm{gp}}$ int. points
 2  for $l = 1, \dots, m$ do                 // select interpolation points for mode gradients
 3      $\boldsymbol{q} \leftarrow \boldsymbol{u}_l - \widetilde{\boldsymbol{U}}(\boldsymbol{P}^T\widetilde{\boldsymbol{U}})^{-1}\boldsymbol{P}^T\boldsymbol{u}_l$;                     // project onto current sampling sites
 4      $j \leftarrow \arg\max_{i\in\{1,\dots,n_{\mathrm{gp}}D\}} |q_i|$ ;          // pick gradient component with largest amplitude
 5      $\boldsymbol{P} \leftarrow [\boldsymbol{P}\,\boldsymbol{e}_j]$,   $\widetilde{\boldsymbol{U}} \leftarrow [\boldsymbol{u}_1 \dots \boldsymbol{u}_l]$ ;                 // enrich projection and sampling set
 6      $\Omega_Z \leftarrow \Omega_Z \cup$ ElementContaining$(j)$
 7  for $k = 1, \dots, \ell$ do                                        // add element layers
 8      $\Omega_Z \leftarrow$ GrowOneElementLayer$(\Omega_Z)$
 9  $\Omega_Z \leftarrow \Omega_Z \cup \Omega_{user}$ ;                              // zone of interest is considered
10  construct $\bar{\boldsymbol{Z}}$ and $\boldsymbol{Z}$ based on all/internal-only points of $\Omega_Z$
```

```
Algorithm 7: Online Phase of the Hyper-Reduction (HR)
  Input  : parameter vector $\boldsymbol{p} \in \mathcal{P}$; scatter/gather operator $\boldsymbol{P}_e$ of element $e$ and weight $v_i$ at the
           integration point $\boldsymbol{x}_i$ ($i = 1, \dots, n_{\mathrm{gp}}$)
  Output: reduced vector $\gamma(\boldsymbol{p})$ and nodal temperatures $\tilde{\boldsymbol{u}}$ (optional)
 1  set $\boldsymbol{u}^{(0)} = \bar{\boldsymbol{u}}_*(\boldsymbol{p}); \alpha = 0$ ;                                          // initialize
 2  set $\boldsymbol{r} = \boldsymbol{0}, \boldsymbol{J} = \boldsymbol{0}$ ;                                   // reset r.h.s. and Jacobian
 3  for $e \in \Omega_Z$ do                               // loop over reduced integration domain
 4      $\boldsymbol{r}_e = \boldsymbol{0}, \boldsymbol{J}_e = \boldsymbol{0}$;                         // initialize element residual and stiffness
 5      for $i = 1, \dots, n_e^{\mathrm{gp}}$ do                   // loop over the $n_e^{\mathrm{gp}}$ int. points of element $e$
 6          evaluate the FE matrices $\boldsymbol{N}, \boldsymbol{G}$ and $\boldsymbol{G}_e = \boldsymbol{G}\boldsymbol{P}_e$ at the current int. point;        // $c_{\mathrm{reloc}}$
 7          compute temperature $u_h^{(\alpha)} \leftarrow \boldsymbol{N}\boldsymbol{u}^{(\alpha)}$ and gradient $\boldsymbol{g}_h^{(\alpha)} \leftarrow \boldsymbol{G}\boldsymbol{u}^{(\alpha)}$ at int. point ; // $c_{\mathrm{reloc}}$
 8          evaluate constitutive model $\mu^{(\alpha)} \leftarrow \mu(u_h^{(\alpha)}; \boldsymbol{p})$ ;                         // $c_{\mathrm{const}}$
 9          $\boldsymbol{r}_e \leftarrow \boldsymbol{r}_e + v_i\,\mu^{(\alpha)}\,\boldsymbol{G}_e{}^T\boldsymbol{g}_h^{(\alpha)}$ ;                                                 // $c_{\mathrm{rhs}}$
10          $\boldsymbol{J}_e \leftarrow \boldsymbol{J}_e + v_i\,\mu^{(\alpha)}\,\boldsymbol{G}_e{}^T\boldsymbol{G}_e$ ;                                                 // $c_{\mathrm{Jac}}$
11      $\boldsymbol{r} \leftarrow \boldsymbol{r} + \boldsymbol{P}_e^T\boldsymbol{r}_e; \boldsymbol{J} \leftarrow \boldsymbol{J} + \boldsymbol{P}_e^T\boldsymbol{J}_e\boldsymbol{P}_e$ ;                                          // $c_{\mathrm{rhs}}, c_{\mathrm{Jac}}$
12  $\boldsymbol{r} \leftarrow \boldsymbol{V}^T\boldsymbol{Z}\boldsymbol{Z}^T\boldsymbol{r}$ ;                      // project residual at inner nodes of $\Omega_Z$; $c_{\mathrm{rhs}}$
13  $\boldsymbol{J} \leftarrow \boldsymbol{V}^T\boldsymbol{Z}\boldsymbol{Z}^T\boldsymbol{J}\boldsymbol{V}$ ;                     // project Jacobian at inner nodes of $\Omega_Z$; $c_{\mathrm{Jac}}$
14  solve $\boldsymbol{J}\delta\gamma^{(\alpha)} = -\boldsymbol{r}$ ;                                                // $c_{\mathrm{sol}}$
15  compute $\boldsymbol{u}^{(\alpha+1)} \leftarrow \boldsymbol{u}^{(\alpha)} + \boldsymbol{V}\delta\gamma^{(\alpha)}$ and set $\alpha \leftarrow \alpha + 1$ ; // update nodal temperatures; $c_{\mathrm{reloc}}$
16  converged ($\|\delta\tilde{u}^{(\alpha)}\|_{L^2(\Omega)} = \|\delta\gamma^{(\alpha)}\|_{l^2} < \epsilon_{\max}$)? → end;   else: goto 2
```

Furthermore, we can state an equivalence of the DEIM and the HR under certain conditions, more precisely, the reduced system of the HR is a special case of the DEIM reduced system. Let us assume that the sampling matrices coincide and the collateral basis is also chosen as this sampling matrix, i.e., $\boldsymbol{U} = \boldsymbol{P} = \boldsymbol{Z}$. Let us further assume that we have a Galerkin projection by choosing $\boldsymbol{W} = \boldsymbol{V}$ for the DEIM. Then, $\boldsymbol{P}^T\boldsymbol{U} = \boldsymbol{Z}^T\boldsymbol{Z} = \boldsymbol{I}_M$ is the M-dimensional identity matrix, hence we obtain

$$\boldsymbol{U}(\boldsymbol{P}^T\boldsymbol{U})^{-1}\boldsymbol{P}^T = \boldsymbol{Z}(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{Z}^T = \boldsymbol{Z}\boldsymbol{Z}^T. \tag{46}$$

Then, (38) yields

$$\boldsymbol{V}^T\boldsymbol{Z}\boldsymbol{Z}^T\boldsymbol{J}\boldsymbol{V}\delta\gamma = -\boldsymbol{V}^T\boldsymbol{Z}\boldsymbol{Z}^T\boldsymbol{r}, \tag{47}$$

which is exactly the HR reduced system (45).

A common aspect of HR, and (D)EIM obviously is the point selection by a sampling matrix. The difference, however, is the selection criterion of the internal points. In case of the DEIM, these points are used as interpolation points, while, for the HR, they are used to specify the reduced integration domain.

A main difference of (D)EIM to HR is the way an additional collateral reduced-space is introduced in the reduced setting of the equations. The HR is more simple by not using an additional basis related to the residuals, but the implicit assumption, that colspan$(\boldsymbol{V})$ (which approximates $\boldsymbol{u}$) also approximates $\boldsymbol{r}$ and $\boldsymbol{J}$ well. This is a very reasonable assumption in symmetric elliptic problems and—in a certain way—it mimics the idea of having the same ansatz and test space as in any Galerkin formulation. However, from a mathematical point of view, it may not be valid in some more general cases, as in principle $\boldsymbol{U}$ and $\boldsymbol{V}$ are completely independent. For example, we can multiply the vectorial residual Equation (11) by an arbitrary regular matrix, hence arbitrarily change $\boldsymbol{r}$ (and thus $\boldsymbol{U}$ for the DEIM), but not changing $\boldsymbol{u}$ at all (i.e., not changing the POD-basis $\boldsymbol{U}$). Hence, the collateral basis in the (D)EIM is first an additional technical ingredient and difficulty, which in turn allows for adopting the approximation space to the quantities that needs to be approximated well.

Theoretically, the EIM is well founded by analytical convergence results [32]. However, in addition, as a downside, the Lebesgue-constant, which essentially bounds the interpolation error to the best-approximation error, can grow exponentially. The DEIM is substantiated with a priori error estimates [10]. In particular, the error bounds depend on the conditioning of the small matrix $\boldsymbol{P}^T\boldsymbol{U}$. We are not aware of such a priori results for the HR, but also a posteriori error control has been presented in [31].

3.5. Computational Complexity

While the aim of model reduction is ultimately a reduction of the computing time, this quantity may heavily depend on the chosen implementation (see Section 4.6). Generally, the computational effort can be decomposed into the following contributions:

- the computation of the local unknowns and of their gradients c_{reloc} (gradient/temperature computation),
- the evaluations of the (nonlinear) constitutive model c_{const},
- the assembly of the residual c_{rhs} and of the Jacobian c_{Jac},
- the solution of the (dense) reduced linear system c_{sol}.

From a theoretical point of view, the presented methods differ with respect to c_{reloc}, c_{const}, c_{rhs} and c_{Jac}:

- **Finite Element Simulation** (n_{gp}: number of integration points; n_{el}: number of elements; $n_{\text{el}}^{\text{DOF}}$: degrees of freedom per element)

$$\begin{aligned} c_{\text{reloc}} &= 2n_{\text{gp}}n_{\text{el}}^{\text{DOF}} && \text{(gradient/temperature computation)} \\ c_{\text{const}} &\sim n_{\text{gp}} && \text{(constitutive model)} \\ c_{\text{rhs}} &= 2n_{\text{gp}}n_{\text{el}}^{\text{DOF}} && \text{(residual assembly)} \\ c_{\text{Jac}} &= n_{\text{gp}}\left[(n_{\text{el}}^{\text{DOF}})^2 + 4n_{\text{el}}^{\text{DOF}}\right] && \text{(Jacobian assembly)} \\ c_{\text{sol}} &\sim n^2. \end{aligned}$$

- **Galerkin–POD**

$$\begin{aligned} c_{\text{reloc}} &= 3n_{\text{gp}}m && \text{(gradient/temperature computation)} \\ c_{\text{const}} &\sim n_{\text{gp}} && \text{(constitutive model)} \\ c_{\text{rhs}} &= 2mn_{\text{gp}} && \text{(direct residual assembly)} \\ c_{\text{Jac}} &= (4m + m^2)n_{\text{gp}} && \text{(direct Jacobian assembly)} \\ c_{\text{sol}} &\sim m^3. \end{aligned}$$

- **Hyper-Reduction**

In the following, $n_{\text{gp}}^{\text{RID}}$ is the number of integration points in the RID. Furthermore, $c_{\text{FE}}^{N,B}$ denotes the cost for the evaluation of u and ∇u using the FE matrices $\boldsymbol{N}$ and $\boldsymbol{G}$ and c_{FE}^{r} is the related to the cost for the residual computation on element level (both at least linear in the number of nodes per element + scattered assembly + overhead) and c_{FE}^{K} is the cost related to the contribution to the element stiffness at one element (∼number of nodes per element squared + scattered assembly + overhead). Lastly, c_{A} is the cost for the Jacobian assembly (i.e., matrix scatter operations).

$$\begin{aligned} c_{\text{reloc}} &= \bar{l}m + n_{\text{gp}}^{\text{RID}}c_{\text{FE}}^{N,B} && (\text{get } u, \partial_x u, \partial_y u \text{ in } \Omega_Z) \\ c_{\text{const}} &\sim n_{\text{gp}}^{\text{RID}} && \text{(constitutive model)} \\ c_{\text{rhs}} &= n_{\text{gp}}^{\text{RID}}c_{\text{FE}}^{r} + ml && \text{(residual assembly and projection)} \\ c_{\text{Jac}} &= (m\omega + m^2)l + n_{\text{gp}}^{\text{RID}}c_{\text{FE}}^{K} + c_{\text{A}} && \text{(Jacobian assembly and projection)} \\ c_{\text{sol}} &\sim m^3. \end{aligned}$$

- **Discrete Empirical Interpolation Method**

The computational cost for the DEIM is closely related to the one of the HR by substituting M for l and $\bar{M}$ for $\bar{l}$ (denoting the number of nodes which are needed to evaluate the residual at the M magic points). Similar to the other methods, we denote $n_{\text{gp}}^{\text{DEIM}}$ as the number of quadrature points to evaluate the entries of the residual and Jacobian. In the cost notation of the Algorithm 5, we obtain

$$\begin{aligned} c_{\text{reloc}} &\sim m + nm \\ c_{\text{const}} &= n_{\text{gp}}^{\text{DEIM}} && \text{(constitutive model)} \\ c_{\text{rhs}} &\sim M\bar{M} && \text{(residual assembly and projection)} \\ c_{\text{Jac}} &\sim M\bar{M}m && \text{(Jacobian assembly and projection)} \\ c_{\text{sol}} &\sim m^3. \end{aligned}$$

From these considerations, the following conclusions can be drawn: First, the number of integration points (n_{gp}, $n_{\text{gp}}^{\text{RID}}$, $n_{\text{gp}}^{\text{DEIM}}$) required for the residual and Jacobian evaluation enter linearly into the effort. Second, the reduced basis dimension m enters linearly into the residual assembly and both linearly and quadratically into the Jacobian assembly. Third, for the HR and the (D)EIM, the ratio $\bar{l}/l$ and $\bar{M}/M$ have a significant impact on the efficiency: for the considered 2D problem with quadratic ansatz functions, these ratios can range from 1 up to 21, i.e., for the same number of magic points pronounced variations in the runtime are in fact possible. The ratio $\bar{l}/l$ is determined by the topology of the RID, i.e., for connected RIDs it is smaller than for a scatter RID (i.e., for many disconnected regions forming the RID). Similarly, the (D)EIM has much smaller computational complexity in the case of magic points belonging to connected elements. Fourth, the Galerkin-POD can be based on simplified algebraic operations as no nodal variables need to be computed. This is due to the fact that the reduced residual and Jacobian are directly assembled without recurse to nodal coordinates and to any standard FE routine.

4. Numerical Results

4.1. ONLINE/OFFLINE Decomposition and RB Identification

In the following, we investigate the behavior of the heat conduction problem (1) for parameters, which we recall from Equation (7)

$$\boldsymbol{p} = [g_x, g_y, c, \mu_0, \mu_1] \in [0,1] \times [0,1] \times [1,2] \times \{1\} \times \{0.5\}. \tag{48}$$

First, a regular parameter grid containing 125 equidistant snapshot points is generated and the high-fidelity FE model is solved for all those points yielding solutions $u_{h,i}, i = 1, \ldots, 125$. Here, the FEM discretization is based on a discretization into 800 biquadratic quadrilateral elements comprising a total of 2560 nodes (including 160 boundary nodes). The problem hence has $n = 2400$ unknowns. In order to exemplify the nonlinearity of the problem due to the temperature dependent conductivity, the conductivity (top row) and the temperature field (bottom row) are shown for three different parameters in Figure 2.

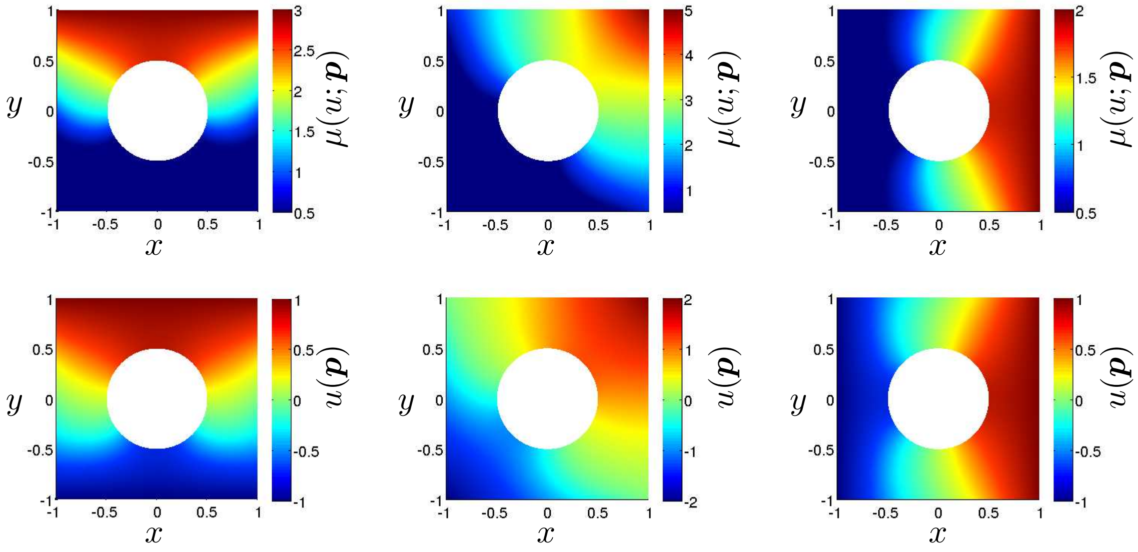


Figure 2. Parameter dependent conductivity $\mu(u;\boldsymbol{p})$ (top row) and solution $u(\boldsymbol{x};\boldsymbol{p})$ (bottom row) for three different snapshot parameters.

Then, a snapshot POD is performed in order to obtain an RB from the snapshots, and the normalized spectrum of the snapshot correlation matrix is shown in Figure 3. Different dimensions of the RB are considered with the projection error

$$E_m = \sqrt{\frac{\sum_{i=1}^{125} \|u_h(\boldsymbol{p}_i) - \tilde{P}_m u_h(\boldsymbol{p}_i)\|^2_{L^2(\Omega)}}{\sum_{i=1}^{125} \|u_h(\boldsymbol{p}_i)\|^2_{L^2(\Omega)}}} \tag{49}$$

given in Table 1 and the decay is visualized in Figure 3 (right). Here, $\tilde{P}_m$ denotes the orthogonal projection operator onto the m-dimensional RB with respect to the standard inner product $\langle \cdot, \cdot \rangle_{L^2(\Omega)}$ in the $L^2(\Omega)$ function space

$$\tilde{P}_m u_h(\boldsymbol{p}) = \sum_{i=1}^{m} \sum_{k=1}^{m} \psi_i \, (\tilde{\boldsymbol{M}}^{-1})_{ik} \, \langle u_h(\boldsymbol{p}), \psi_k \rangle_{L^2(\Omega)}, \qquad \tilde{M}_{ik} = \langle \psi_i, \psi_k \rangle_{L^2(\Omega)} \quad i, j = 1, \dots, m. \tag{50}$$

Table 1. Dimension of the RB vs. projection error for the snapshots.

Dimension of RB	**16**	**24**	**32**	**48**	**60**
E_m	6.629×10^{-3}	4.120×10^{-3}	3.180×10^{-3}	2.017×10^{-3}	1.486×10^{-3}

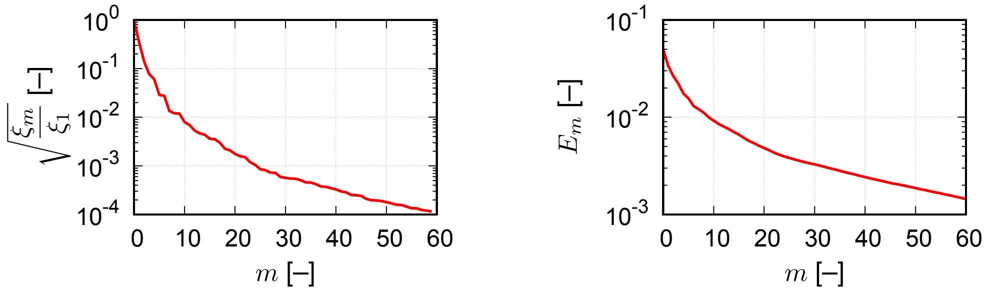


Figure 3. Decay of the spectrum of $\boldsymbol{C}$ (normalized to largest eigenvalue ξ_1) and the relative projection error E_m of the snapshot data defined by (49).

The low approximation errors E_m given in Table 1 and in Figure 3 (right) indicate that the training data can essentially be represented rather accurately by the RB using a projection. Additionally, the projection error naturally decreases with increasing dimension. However, the solution of the reduced problem does not necessarily follow the same monotonicity. Since μ is bounded away from 0 due to $\mu_0 > 0$, the problem under consideration is coercive. Similar to the linear case, we expect that the approximation error $e(\boldsymbol{p}) = \tilde{w}(\gamma(\boldsymbol{p}); \boldsymbol{p}) - w_h(\boldsymbol{p})$ and the projection error are comparable in the sense that

$$\eta(\boldsymbol{p}) := \frac{\|e(\boldsymbol{p})\|_{L^2(\Omega)}}{\|u_h - \tilde{P}_m u_h\|_{L^2(\Omega)}} \tag{51}$$

is small. Due to the best-approximation by the orthogonal projection, we certainly have $\eta(\boldsymbol{p}) \geq 1$. Numerical Values are provided in the following. The constant $\eta(\boldsymbol{p})$ is generally not available in closed form for the considered nonlinear problem. Assuming a constant conductivity μ evaluated at the solution $u(\boldsymbol{p})$, an upper bound $\eta_{\mathrm{UB}}(\boldsymbol{p})$ for the numerically determined $\eta(\boldsymbol{p})$ is provided via Cea's Lemma (for symmetric problems):

$$\eta_{\mathrm{UB}}^{\mathrm{h}} := \sqrt{\frac{\hat{c}_{\mathrm{h}}(\boldsymbol{p})}{\check{c}_{\mathrm{h}}(\boldsymbol{p})}}, \qquad \hat{c}_{\mathrm{h}} := \sup_{v_{\mathrm{h}}, w_{\mathrm{h}} \in \mathcal{V}_0^{\mathrm{h}} \setminus \{0\}} \frac{a_{\mathrm{c}}(v_{\mathrm{h}}, w_{\mathrm{h}}; \boldsymbol{p})}{\|v_{\mathrm{h}}\|_{L^2(\Omega)} \|w_{\mathrm{h}}\|_{L^2(\Omega)}}, \qquad \check{c}_{\mathrm{h}} := \inf_{v_{\mathrm{h}} \in \mathcal{V}_0^{\mathrm{h}} \setminus \{0\}} \frac{a_{\mathrm{c}}(v_{\mathrm{h}}, v_{\mathrm{h}}; \boldsymbol{p})}{\|v_{\mathrm{h}}\|_{L^2(\Omega)}^2}. \tag{52}$$

Here, $\hat{c}_{\mathrm{h}}$ and $\check{c}_{\mathrm{h}}$ are the continuity constant and the coercivity constant of the condensed bilinear form

$$a_{\mathrm{c}}(w_{\mathrm{h}}, v_{\mathrm{h}}; \boldsymbol{p}) := \int_{\Omega} \nabla w_{\mathrm{h}} \cdot \mu(u_{\mathrm{h}}(\boldsymbol{p}); \boldsymbol{p}) \nabla v_h \, \mathrm{d}\Omega \quad (\forall v_{\mathrm{h}}, w_{\mathrm{h}} \in \mathcal{V}_0^{\mathrm{h}}), \tag{53}$$

with $u_{\mathrm{h}}(\boldsymbol{p})$ denoting the Finite Element solution of the nonlinear problem. The estimate $\eta_{\mathrm{UB}}^{\mathrm{h}}$ is not a tight bound and, at the same time, its numerical evaluation is computationally expensive, so it is typically considered to be of limited practical use.

4.2. Test Cases

In order to investigate the accuracy of the reduced models, additional parameter vectors need to be considered that are not matching the training data. Two test cases are considered in the sequel:

[A] A diagonal in the parameter space is considered with

$$\boldsymbol{p}^{(j)} := \boldsymbol{p}_0 + \beta^{(j)}(\hat{\boldsymbol{p}} - \boldsymbol{p}_0), \qquad \boldsymbol{p}_0 := [0, 0, 1, 1, 1/2], \qquad \hat{\boldsymbol{p}} := [1, 1, 2, 1, 1/2]. \tag{54}$$

A total of 101 equally spaced values of $\beta^{(j)}$ was chosen, i.e., $\beta^{(j)} = \frac{j}{100}$ for $j = 0, 1, \ldots, 100$.

[B] A set of 1000 random parameters $\boldsymbol{p}^{(j)}$ was generated using a uniform distribution in parameter space, i.e., a uniform distribution $\mathcal{U}([0, 1])$ was chosen for g_{x}, g_{y} and the parameter c was assumed to be distributed via $\mathcal{U}([1, 2])$.

4.3. Certification of the Galerkin RB Method

First, the ability of the Galerkin RB solution to approximate the optimal orthogonal projection and, thereby, the high-fidelity solution, was verified. Therefore, the constant $\eta(\boldsymbol{p})$ was evaluated for all snapshots of case **[A]** and **[B]**. The minimum, the mean and the maximum of $\eta(\boldsymbol{p})$ were determined for the 101 and 1000 test of case **[A]** and **[B]**, respectively. The results shown in Table 2 state the POD approximation error is found close to the projection error. This confirms the quality of the chosen RB.

Table 2. Computed values of $\eta(\boldsymbol{p})$ for different modes sets and for test cases **[A]**, **[B]**; the last row represents the upper bound $\eta_{\mathrm{UB}}^{\mathrm{h}} \geq \eta(\boldsymbol{p})$

Test Case	**[A]**			**[B]**		
	min.	**mean**	**max.**	**min.**	**mean**	**max.**
$m = 16$	1.000	1.4708	1.8248	1.000	1.3185	2.0239
$m = 24$	1.000	1.9088	2.8926	1.000	1.3187	2.6559
$m = 32$	1.000	1.7273	2.7441	1.000	1.3679	2.4393
$m = 48$	1.000	1.5386	2.0232	1.000	1.3051	1.9371
$m = 60$	1.000	1.5096	1.9333	1.000	1.3447	1.8922
$\eta_{\mathrm{UB}}^{\mathrm{h}}$ cf. (52)	62.12	64.693	70.581	62.51	74.728	82.504

Note that in test case **[B]** only a finite number of random parameter vectors was chosen which does not necessarily contain the extreme values of $\eta(\boldsymbol{p})$. The numerical data in Table 2 for test case **[A]** shows that indeed, **[A]** contains parameters leading to larger values of $\eta(\boldsymbol{p})$. When increasing the size of the random parameter set for **[B]**, the maximum values of $\eta(\boldsymbol{p})$ in case **[B]** should be equal or larger than the maximum values of case **[A]**.

In addition to the error magnification parameter $\eta(\boldsymbol{p})$, the minimum, average and maximum of the relative error

$$\delta(\tilde{u}(\boldsymbol{p}), \boldsymbol{p}) = \frac{\|\tilde{u}(\boldsymbol{p}) - u_h(\boldsymbol{p})\|_{L^2(\Omega)}}{\|u_h(\boldsymbol{p})\|_{L^2(\Omega)}} \tag{55}$$

are also computed for all samples. The results are provided in Table 3. Note that for test case **[A]** the minimum error is truly zero for $g_x = g_y = 0$, which implies a homogeneous zero temperature. For an RB of dimension 32, the mean error is well below 10^{-3} for all tests and the maximum error over all tests is 3.23×10^{-3}. This basis provides a compromise between accuracy and computational cost and is therefore used for the comparison of the methods in the sequel.

Table 3. Relative error of the Galerkin RB approximation

Test Case	**[A]**			**[B]**		
	min.	**mean**	**max.**	**min.**	**mean**	**max.**
$m = 16$	0.00 10^{-3}	2.54 10^{-3}	7.28 10^{-3}	0.67 10^{-3}	1.75 10^{-3}	7.49 10^{-3}
$m = 24$	0.00 10^{-3}	1.07 10^{-3}	3.80 10^{-3}	0.16 10^{-3}	0.88 10^{-3}	5.29 10^{-3}
$m = 32$	0.00 10^{-3}	0.82 10^{-3}	3.23 10^{-3}	0.18 10^{-3}	0.63 10^{-3}	3.13 10^{-3}
$m = 48$	0.00 10^{-3}	0.31 10^{-3}	1.14 10^{-3}	0.06 10^{-3}	0.33 10^{-3}	2.07 10^{-3}
$m = 60$	0.00 10^{-3}	0.20 10^{-3}	0.70 10^{-3}	0.05 10^{-3}	0.26 10^{-3}	1.55 10^{-3}

Note that the slow decay of the accuracy of the Galerkin approximation indicated by the data provided in Table 3 indicates that the information content captured in the training snapshots is not sufficient to provide better accuracies. Therefore, we decided on a dimension $m = 32$ for the reduced basis in the subsequent experiments.

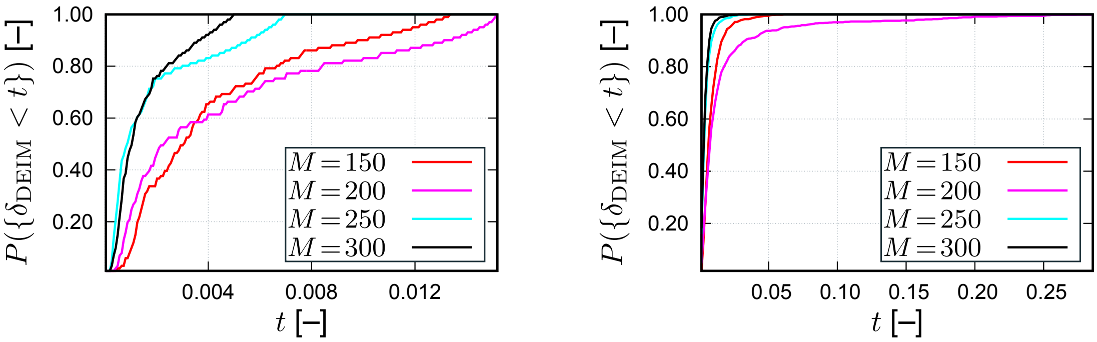
In order to further reduce the computational cost, the use of additional reduction techniques such as the Hyper-Reduction (HR) and Discrete Empirical Interpolation Method (DEIM) is required. Note also that the reduction techniques using a POD basis are only approximations of the Galerkin RB. Hence, the HR and DEIM cannot be better than the Galerkin RB solution except in few cases where $\eta_{\mathrm{HR}}(\boldsymbol{p}) < \eta(\boldsymbol{p})$ or $\eta_{\mathrm{DEIM}}(\boldsymbol{p}) < \eta(\boldsymbol{p})$, where $\eta_{\mathrm{HR}}(\boldsymbol{p})$ and $\eta_{\mathrm{DEIM}}(\boldsymbol{p})$ denote the constant η from (51) for the HR and DEIM method. In our numerical tests, this occurred only exceptionally, which can be explained by the slightly different considered residual in comparison to the Galerkin RB. Let us now turn to a more realistic multi-query situation where RB is crucial for non-prohibitive runtimes.

4.4. Application to Uncertainty Quantification

In real world simulation scenarios, material coefficients and boundary conditions are often not exactly known and one is interested in the impact of this uncertainty on the quantities of interest. To this end, uncertainty quantification (UQ) has been proposed and has become an active research field on its own. In classical forward UQ, the critical parameters are modeled as random variables; the distributions and correlation are derived from measurements as for example shown for nonlinear material curves in [33]. Finally, the forward model is evaluated at collocation points $\boldsymbol{p}_i$ in the parameter space according to a quadrature method as e.g., Monte Carlo, [34]. Typically, many collocations (or 'sampling') points are needed and therefore model order reduction has been shown to significantly reduce the computational costs, e.g., [35].

In the case of our thermal benchmark problem, the parameter vector $\boldsymbol{p} = \boldsymbol{P}(\omega)$ is considered as a realization of the random vector, where $\omega \in \Omega_{\mathrm{p}}$ and $(\Omega_{\mathrm{p}}, \mathcal{F}, \mathcal{P})$ is the usual probability space. We refer to this as test case **[C]** and assume that the random variables are independent and uniformly distributed as already introduced in test case **[B]**

$$\boldsymbol{P}(\omega) = [G_{\mathrm{x}}(\omega), G_{\mathrm{y}}(\omega), C(\omega)] \qquad \text{with} \quad G_{\mathrm{x}}, G_{\mathrm{y}} \sim \mathcal{U}([0,1]) \quad \text{and} \quad C \sim \mathcal{U}([1,2]).$$



test case **[A]**

test case **[B]**

Figure 4. Statistical distribution function of the relative error of the DEIM for different numbers of magic points $M \in \{150, 200, 250, 300\}$ (dimension of POD basis: $m = 32$).

Finally, the statistical moments of the solution $u_\star(\boldsymbol{x}; \boldsymbol{P}(\omega))$ are approximated by the Monte Carlo method, e.g., [34]

$$\mathbb{E}\big(u_\star(\boldsymbol{x}; \boldsymbol{P})\big) \approx \frac{1}{n_{\text{p}}} \sum_{j=1}^{n_{\text{p}}} u_\star(\boldsymbol{x}; \boldsymbol{p}^{(j)}) =: \bar{u}_\star(\boldsymbol{x}), \tag{56}$$

$$\mathbb{M}^k\big(u_\star(\boldsymbol{x}; \boldsymbol{P})\big) \approx \frac{1}{n_{\text{p}}} \sum_{j=1}^{n_{\text{p}}} \Big(u_\star(\boldsymbol{x}; \boldsymbol{p}^{(j)}) - \bar{u}_\star(\boldsymbol{x})\Big)^k =: m_\star^k(\boldsymbol{x}), \tag{57}$$

where $\boldsymbol{p}^{(j)}$ are the same $n_{\text{p}} = 1000$ random sample points generated for case **[B]** and $u_\star$ and $m_\star^k$ are the approximations of the solution and its k-th centered moment ($k > 1$) obtained from Finite Elements, Galerkin RB, DEIM and HR, i.e., $\star \in \{\text{h}, \text{RB}, \text{DEIM}, \text{HR}\}$, respectively (e.g., m_{h}^3 is the third statistical moment obtained from Finite Element computations). For the DEIM, we selected a fixed number of $M = 400$ magic points, which is a conservative choice (see Figure 4 for a discussion). In practice, one may choose an adaptive selection strategy.

An estimation of the normalized root mean square error of the finite element solution due to Monte Carlo sampling can be obtained by

$$E_{\text{MC}} = \frac{1}{\sqrt{n_{\text{p}}}\, \|\bar{u}_{\text{FE}}\|_{L^2(\Omega)}} \left\| \sqrt{m_h^2} \right\|_{L^2(\Omega)} \approx 1.78 \cdot 10^{-2} \equiv 1.78\%.$$

This implies that the accuracy of the reduced models in the prediction of $\bar{u}_\star$ should be around 2% or better, in order to render the reduced models capable of providing meaningful quantitative statistical information. Figure 5 shows the error in the moments computed for the various reduction techniques with respect to the finite element reference solution

$$E_\star^k = \frac{\|m_\star^k - m_h^k\|_{L^2(\Omega)}}{\|m_h^k\|_{L^2(\Omega)}}.$$

The figure indicates that the approximations of the expected value $\bar{u}_\star$ are at least accurate up to 10^{-3} and thus below (i.e., better than) the sampling accuracy E_{MC}. Generally, the DEIM tends to perform better than the HR; the largest errors occur for $E_{\text{HR}}^7 = 1.09 \times 10^{-1}$ and $E_{\text{DEIM}}^7 = 3.46 \times 10^{-2}$, respectively.

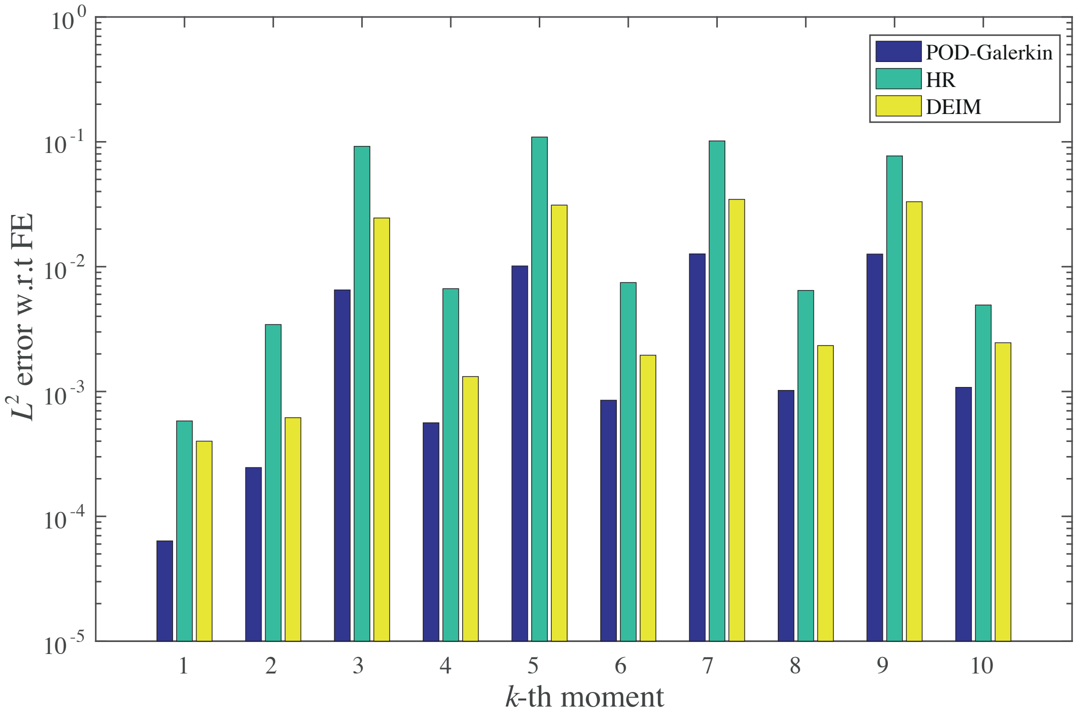


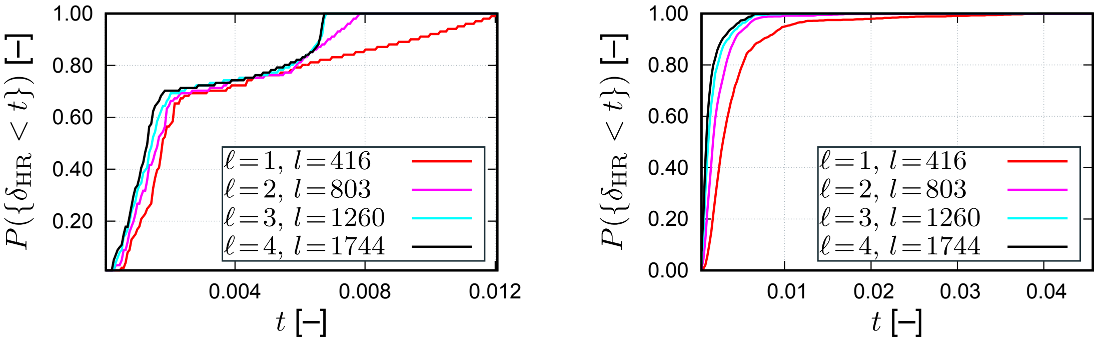
Figure 5. $L^2(\Omega)$ error of the centered moments w.r.t. the finite element solution. Please note that we have set $m_\star^1 := \bar{u}_\star$ for simplicity of notation.

4.5. Accuracy of the HR and DEIM vs. Number of Interpolation Points

Both the HR and the DEIM sample the nonlinearity only at entries on the right-hand side: the interpolation points. The HR has one additional parameter ℓ describing how many element layers around a certain degree of freedom (DOF) should be considered in order to generate Ω^+, which add additional interpolation points to the RID. The location of the DOF around which the interpolation points are located are selected based on the criterion described in Section 3.3. In contrast to that, the DEIM selects the points using only the right-hand side information of the system. For the DEIM, the number of sampling points M is an input parameter describing the dimension of the collateral basis. The effect of the number of points is investigated in the following. Based on the considerations in Section 3.4, both the HR and the DEIM should reproduce the POD solution for a large number of sampling sites, while, for a lower number of points, the accuracy is a trade-in for computational efficiency.

For the DEIM, different interpolation point numbers are considered for both test cases **[A]** and **[B]**. The resulting relative errors are compared in Figure 4 in terms of the statistical distribution function $P(t)$ of the relative error, i.e., the probability of finding a relative error δ that is smaller than or equal to t. Obviously the number of interpolation points has a pronounced impact on the distribution. Generally, the error function for a low number of points states a significant increase of the computational error due to DEIM in comparison with the POD. With an increasing number of points, the distribution function approaches the one of the POD. In our test, the use of more than 300 sampling points can only improve the accuracy in a minor way. We must note that, in general, the accuracy of the DEIM must not be a monotonic function of interpolation points number.

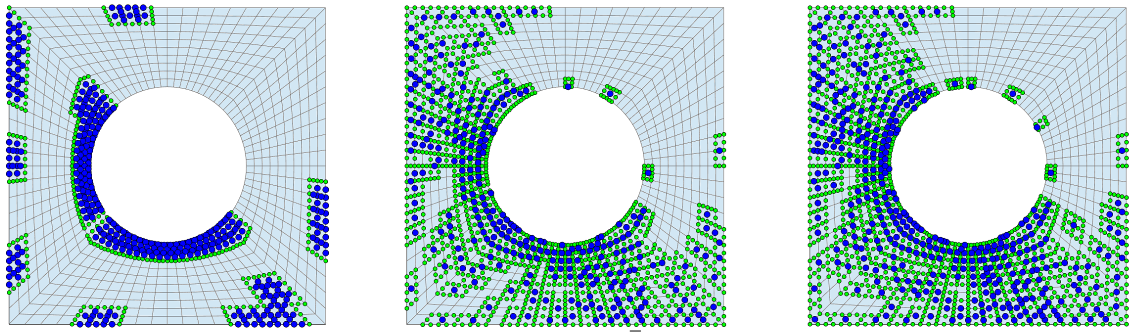
For the hyper-reduced predictions, different layers of elements are added in Ω^+ in order to extend the RID. We have considered here $\ell = 1, 2, 3$ and 4 layers of elements connected to Ω^u, for both test cases **[A]** and **[B]**. The resulting relative errors are compared in Figure 6 in terms of the statistical distribution function $P(t)$ of the relative error. With increasing number of layers, the distribution function approaches the one of the POD. The number of internal points, l, increases rapidly when increasing the number of layers. In the case of the DEIM, the growth of interpolation points is much more progressive. More than two layers of elements do not improve the accuracy significantly.



test case **[A]** test case **[B]**

Figure 6. Statistical distribution function of the relative error of the hyper-reduction for different layers of elements added to the RID $\ell \in \{1, 2, 3, 4\}$ (dimension of POD basis: $m = 32$).

One layer of additional elements gives predictions less accurate than the DEIM for approximately the same number of internal points/interpolation points (here: $l = 416$ for the HR and $M = 300$ for the DEIM). However, the number of additional points required for the residual evaluation differs considerably: $\bar{l} = 657$ vs. $\bar{M} = 1490$. This can be explained by direct comparison of the reduced domains in Figure 7. For the Hyper-reduction, the RID is rather compact (leftmost plot) while the magic points of the DEIM are rather scattered, thus requiring the temperature evaluation at many additional points.



HR $(\ell = 1; l = 416, \bar{l} = 657)$ DEIM $(M = 300, \bar{M} = 1490)$ DEIM $(M = 400, \bar{M} = 1651)$

Figure 7. Position of the magic points (blue points) and the additional points required for the evaluation of the residual (green points); Hyper-reduction (left) vs. DEIM (middle, right) for $m = 32$.

The predictions of the Hyper-reduction are particularly less accurate, which can especially be seen by comparing Figure 4 (left, test case **[A]**, black line) and Figure 6 (left, test case **[A]**, red line), where 80% of the samples lead to errors below ≈ 0.002 for the DEIM and ≈ 0.01 for the HR. Nevertheless, the accuracy of the hyper-reduced predictions is generally of the same order of magnitude as the accuracy of the DEIM.

4.6. Computing Times

Although the main objective of model reduction is the reduction of computing time, this paper does not aim at benchmarking HR and DEIM. If not optimized properly, runtime benchmarks compare rather the performance of the implementations than that of the algorithms. Nonetheless, the numbers in Table 4 indicate the effect of varying parameters in the respective techniques using our Matlab implementation. In the case of a few magic points, divergence of the fixed point iteration was observed for a small number of configurations in the parametric space. The number is stated in the table as failure. HR and DEIM have similar performance, although HR seems to be slightly more robust for the problem at hand in the sense that it exhibits slightly less failures of the iterative solver. Finally, as expected, the overall runtime scales linearly with the number of solves, i.e., computing test case **[B]** is approx. 10 times slower than **[A]**.

Table 4. Elapsed time and number of failures of the nonlinear solver for the solution of all computations of test case **[A]** (101 solves) and **[B]** (1000 solves) for Finite Element (FE), Hyper-Reduction (HR) with varying number of additional element layers $\ell \in \{1,2,3,4\}$, DEIM for $M \in \{200,250,300,400\}$ magic points; all computations are carried out using $N = 32$ POD modes.

	Test Case [A]		**Test Case [B]**	
	Time	**Fail**	**Time**	**Fail**
FE	59.9 s	-	660.5	-
HR, $\ell = 1$	10.6 s	0	118.9 s	5
HR, $\ell = 2$	17.4 s	0	180.6 s	2
HR, $\ell = 3$	24.5 s	0	247.2 s	1
HR, $\ell = 4$	30.9 s	0	303.5 s	0
DEIM, $M = 200$	20.6 s	0	292.3 s	108
DEIM, $M = 250$	21.4 s	0	240.6 s	7
DEIM, $M = 300$	24.4 s	0	249.0 s	3
DEIM, $M = 400$	27.6 s	0	272.7 s	2

5. Conclusions

The presented study revisits an—at first sight—rather simple nonlinear heat conduction problem. In order to accelerate the nonlinear computations, a Galerkin reduced basis ansatz is proposed (see Section 3.1) using preliminary offline computations in the established framework of the snapshot POD.

Both the HR and the (D)EIM can achieve significant accelerations of the computing time. These scale approximately with the number of magic points (here: l or M) and/or with the number of points connected to the magic points ($\bar{l}$ and $\bar{M}$, respectively). In the presented examples, less than 25% of the original mesh were considered in both the (D)EIM and the HR. By virtue of the considerations presented in Section 3.5, the computing times are reduced accordingly.

The selection of the magic points in the HR and the (D)EIM requires the computation of the solution at an increased number of nodes, i.e., $\bar{l} \geq l$ and $\bar{M} \geq M$, respectively. The higher the space dimension (2D, 3D, . . .), the more scattered the magic points and the higher the number of degrees of freedom per element, the more are $\bar{l}$ and $\bar{M}$ increased in comparison to l and M.

With respect to the implementation, it shall be noted that the HR is less intrusive than the (D)EIM as it uses standard simulation outputs to generate the modes, while the (D)EIM requires additional outputs for the construction of the collateral basis. Other than that, both techniques can be implemented using mostly the same implementation, which is also confirmed by the similarity of both techniques presented in Section 3.4.

Acknowledgments: The authors acknowledge the generous funding of the CoSiMOR scientific network by the German Research Foundation/Deutsche Forschungsgemeinschaft (DFG) under grants DFG-FR-2702/4-1, DFG-FR-2702/7-1. Felix Fritzen is thankful for financial support in the framework of the DFG Emmy-Noether group EMMA under grant DFG-FR2702/6. Bernard Haasdonk and Felix Fritzen acknowledge the support via the cluster of excellence SimTech (EXC310). Sebastian Schöps acknowledges support from the Excellence Initiative of the German Federal and State Governments and the Graduate School of CE at TU Darmstadt.

Author Contributions: All authors performed experiments, contributed to the software and the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gubisch, M.; Volkwein, S. POD for Linear-Quadratic Optimal Control. In *Model Reduction and Approximation: Theory and Algorithms*; Benner, P., Cohen, A., Ohlberger, M., Willcox, K., Eds.; SIAM: Philadelphia, PA, USA, 2017.
2. Volkwein, S. Optimal control of a phase-field model using proper orthogonal decomposition. *ZAMM* **2001**, *81*, 83–97.
3. Antoulas, A. *Approximation of Large–Scale Dynamical Systems*; SIAM Publications: Philadelphia, PA, USA, 2005.
4. Patera, A.; Rozza, G. *Reduced Basis Approximation and a Posteriori Error Estimation for Parametrized Partial Differential Equations*; Version 1.0, Copyright MIT 2006-2007, to Appear in (Tentative Rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering; Massachusetts Institute of Technology (MIT): Cambridge, MA, USA, 2007.
5. Haasdonk, B. Reduced Basis Methods for Parametrized PDEs—A Tutorial Introduction for Stationary and Instationary Problems. In *Model Reduction and Approximation: Theory and Algorithms*; Benner, P., Cohen, A., Ohlberger, M., Willcox, K., Eds.; SIAM: Philadelphia, PA, USA, 2017.
6. Barrault, M.; Maday, Y.; Nguyen, N.; Patera, A. An 'empirical interpolation' method: Application to efficient reduced-basis discretization of partial differential equations. *C. R. Math. Acad. Sci. Paris Ser. I* **2004**, *339*, 667–672.
7. Haasdonk, B.; Ohlberger, M. Reduced basis method for explicit finite volume approximations of nonlinear conservation laws. In Proceedings of the HYP 2008, International Conference on Hyperbolic Problems: Theory, Numerics and Applications, Providence, RI, USA, 9–13 June 2008; American Mathematical Society: Providence, RI, USA, 2009; Volume 67, pp. 605–614.
8. Drohmann, M.; Haasdonk, B.; Ohlberger, M. Reduced Basis Approximation for Nonlinear Parametrized Evolution Equations based on Empirical Operator Interpolation. *SIAM J. Sci. Comput.* **2012**, *34*, A937–A969.
9. Chaturantabut, S.; Sorensen, D.C. Discrete empirical interpolation for nonlinear model reduction. In Proceedings of the 48th IEEE Conference on Decision and Control and the 28th Chinese Control Conference (CDC/CCC 2009), Shanghai, China, 15–18 December 2009; pp. 4316–4321.
10. Chaturantabut, S.; Sorensen, D.C. A State Space Error Estimate for POD-DEIM Nonlinear Model Reduction. *SIAM J. Numer. Anal.* **2012**, *50*, 46–63.
11. Ryckelynck, D. A priori hypereduction method: An adaptive approach. *J. Comput. Phys.* **2005**, *202*, 346–366.
12. Ryckelynck, D. Hyper reduction of mechanical models involving internal variables. *Int. J. Numer. Methods Eng.* **2009**, *77*, 75–89.
13. Everson, R.; Sirovich, L. Karhunen-Loève procedure for gappy data. *J. Opt. Soc. Am. A* **1995**, *12*, 1657–1664.
14. Astrid, P.; Weiland, S.; Willcox, K.; Backx, T. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Trans. Autom. Control* **2008**, *53*, 2237–2251.
15. Carlberg, K.; Bou-Mosleh, C.; Farhat, C. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *Int. J. Numer. Methods Eng.* **2011**, *86*, 155–181.
16. Farhat, C.; Avery, P.; Chapman, T.; Cortial, J. Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency. *Int. J. Numer. Methods Eng.* **2014**, *98*, 625–662.
17. Fritzen, F.; Hodapp, M. The Finite Element Square Reduced (FE^{2R}) method with GPU acceleration: Towards three-dimensional two-scale simulations. *Int. J. Numer. Methods Eng.* **2016**, *107*, 853–881.
18. Fritzen, F.; Xia, L.; Leuschner, M.; Breitkopf, P. Topology optimization of multiscale elastoviscoplastic structures. *Int. J. Numer. Methods Eng.* **2016**, *106*, 430–453.
19. Dimitriu, G.; Ştefănescu, R.; Navon, I.M. Comparative numerical analysis using reduced-order modeling strategies for nonlinear large-scale systems. *J. Comput. Appl. Math.* **2017**, *310*, 32–43.
20. Bathe, K.I.; *Finite-Elemente-Methoden*; Springer: Berlin/Heidelberg, Germany, 2002.
21. Zienkiewicz, O.; Taylor, R.; Zhu, J. *Finite Element Method*; Butterworth-Heinemann: Oxford, UK, 2006.
22. Kelley, C.T. *Iterative Methods for Linear and Nonlinear Equations*; SIAM: Philadelphia, PA, USA, 1995.
23. Sirovich, L. Turbulence and the dynamics of coherent structures part I: Coherent structures. *Q. Appl. Math.* **1987**, *65*, 561–571.
24. Joliffe, I. *Principal Component Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2002.

25. Ştefănescu, R.; Sandu, A.; Navon, I.M. Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations. *Int. J. Numer. Methods Fluids* **2014**, *76*, 497–521.
26. Fritzen, F.; Hodapp, M.; Leuschner, M. GPU accelerated computational homogenization based on a variational approach in a reduced basis framework. *Comput. Methods Appl. Mech. Eng.* **2014**, *278*, 186–217.
27. Haasdonk, B.; Ohlberger, M.; Rozza, G. A Reduced Basis Method for Evolution Schemes with Parameter-Dependent Explicit Operators. *ETNA Electron. Trans. Numer. Anal.* **2008**, *32*, 145–161.
28. Wirtz, D.; Sorensen, D.; Haasdonk, B. A Posteriori Error Estimation for DEIM Reduced Nonlinear Dynamical Systems. *SIAM J. Sci. Comput.* **2014**, *36*, A311–A338.
29. Ryckelynck, D.; Missoum Benziane, D. Multi-level a priori hyper reduction of mechanical models involving internal variables. *Comput. Methods Appl. Mech. Eng.* **2010**, *199*, 1134–1142.
30. Ryckelynck, D.; Vincent, F.; Cantournet, S. Multidimensional a priori hyper-reduction of mechanical models involving internal variables. *Comput. Methods Appl. Mech. Eng.* **2012**, *225*, 28–43.
31. Ryckelynck, D.; Gallimard, L.; Jules, S. Estimation of the validity domain of hyper-reduction approximations in generalized standard elastoviscoplasticity. *Adv. Model. Simul. Eng. Sci.* **2015**, *2*, 6.
32. Maday, Y.; Nguyen, N.; Patera, A.; Pau, G. *A General, Multi-Purpose Interpolation Procedure: The Magic Points*; Technical Report RO7037; Laboratoire Jaques-Louis-Lions, Université Piere et Marie Curie: Paris, France, 2007.
33. Römer, U.; Schöps, S.; Weiland, T. Stochastic Modeling and Regularity of the Nonlinear Elliptic curl-curl Equation. *SIAM/ASA J. Uncertain. Quantif.* **2016**, *4*, 952–979.
34. Xiu, D. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*; Princeton University Press: Princeton, NI, USA, 2010.
35. Haasdonk, B.; Urban, K.; Wieland, B. Reduced Basis Methods for parameterized partial differential equations with stochastic influences using the Karhunen-Loeve expansion. *SIAM/ASA J. Uncertain. Quantif.* **2013**, *1*, 79–105.