Affine Decompositions of Parametric Stochastic Processes for Application within Reduced Basis Methods ⋆

Karsten Urban * **Bernhard Wieland** **

* *Ulm University, 89069 Ulm, Germany (e-mail: karsten.urban@uni-ulm.de).*
** *Ulm University, 89069 Ulm, Germany (e-mail: bernhard.wieland@uni-ulm.de)*

Abstract: We consider parameter dependent spatial stochastic processes in the context of partial differential equations (PDEs) and model order reduction. For a given parameter, a random sample of such a process specifies a sample coefficient function of a PDE, e.g. characteristics of porous media such as Li-ion batteries or random influences in biomechanical systems. To apply the Reduced Basis Method (RBM) to parametrized systems (with stochastic or deterministic parameter dependencies), it is necessary to obtain affine decompositions of the systems in parameter and space (cf. e.g. Patera and Rozza (2006); Haasdonk et al. (2012)). For deterministic problems, it is common to use the Empirical Interpolation Method (EIM) (cf. Barrault et al. (2004)). For stochastic coefficients, one can apply the Karhunen-Loève (KL) expansion (cf. Karhunen (1947), Loève (1978)) where the terms involving stochastic dependencies are assumed to satisfy certain distributions and are modeled using polynomial chaos (PC) expansions (cf. Ghanem and Spanos (1991)).
In this paper, we extend the EIM to parametrized spatial stochastic processes. The goal is to develop efficiently computable affine decompositions of not only parameter dependent but also stochastic systems that separate spatial dependencies from parametric and probabilistic influences without any assumptions on the distribution of non-spatial terms. We will use the basic concept of the EIM together with ideas from Proper Orthogonal Decomposition (POD) as well as from the Discrete Empirical Interpolation Method (DEIM) (cf. Chaturantabut and Sorensen (2010)). Furthermore, we introduce a combination of these methods with least-squares approximations.
We emphasize that the presented methods are not limited to stochastic functions but work analogously on noisy input data or on other hardly decomposable functions.

Keywords: empirical interpolation method, proper orthogonal decomposition, least-squares approximation, reduced-order models, reduced basis method, a posteriori error estimation

1. INTRODUCTION

Let $(\Omega, \mathcal{F}, P)$ be a probability space, $\mathcal{P} \subset \mathbb{R}^p$ be a set of deterministic parameters, and let $D \subset \mathbb{R}^d$ denote a spatial domain. Furthermore, let c denote a real-valued parameter dependent spatial stochastic process, i.e. $c : D \times (\mathcal{P} \times \Omega) \to \mathbb{R}$, $(x; \mu, \omega) \mapsto c(x; \mu, \omega)$. For each pair $(\mu, \omega) \in \mathcal{P} \times \Omega$, we obtain a trajectory $c(\mu, \omega) \in X$ for some appropriate Hilbert (function) space X on D.

We consider the following problem: For any trajectory $c(\mu, \omega) \in X$, evaluate some given linear functional $\ell(u(\mu, \omega))$, where $u(\mu, \omega) \in X$ is the solution of a PDE $\mathcal{L}(c(\mu, \omega))[u(\mu, \omega)] = 0$ on D. The spatial differential operator $\mathcal{L}$ is supposed to be linear w.r.t. the trajectories $c(\mu, \omega)$.

⋆ K.U. was supported by the Deutsche Forschungsgemeinschaft (DFG) under Ur-63/9 and GrK1100. B.W. was supported by the state of Baden-Württemberg within the Landesgraduierten-Programm. This paper was written while K.U. was Visiting Professor at MIT.

In the context of Reduced Basis Methods (RBM), it is essential that the operator $\mathcal{L}$ is affine w.r.t. the parameter pair (μ, ω), since this allows for an efficient offline-online decomposition. Due to the linearity of $\mathcal{L}$, this is satisfied provided that $c(\mu, \omega)$ is an affine function of the parameters and the spatial variables. In general, however, this requirement is not fulfilled, in particular in the presence of stochastic influences. The objective of this paper is thus (i) to find affine approximations of $c(\mu, \omega)$ of the form

$$c(\mu, \omega) \approx \sum_{m=1}^{M} \theta_m(\mu, \omega)\, q_m \tag{1}$$

with so-called collateral basis functions $q_m \in X$, $m = 1, ..., M$, (ii) to construct efficient evaluation procedures for the coefficients $\theta_m(\mu, \omega) \in \mathbb{R}$, $m = 1, ..., M$, and (iii) the derivation of effective a-posteriori error estimators to choose $M \in \mathbb{N}$ possibly small in order to guarantee a certain approximation error in (1).

Known methods for the construction of affine approximations of non-affine functions include the Empirical Inter-

polation Method (EIM) (cf. Barrault et al. (2004); Tonn (2012)) for deterministic parametric functions and the Discrete EIM (DEIM) (cf. Chaturantabut and Sorensen (2010)) as well as the Operator EIM (cf. Drohmann et al. (2012)) for discrete operator approximations. For stochastic influences in terms of random variables, the Karhunen-Loève (KL) expansion is well-known (cf. Karhunen (1947); Ghanem and Spanos (1991)), which can also be seen as the stochastic counterpart of the Proper Orthogonal Decomposition (POD). The probability distributions of the KL expansion coefficients are modeled using Polynomial Chaos (PC) expansions (cf. Xiu and Karniadakis (2002)).

The main obstruction in our framework is that the EIM is based on the L_∞ approximation error of previously chosen snapshot trajectories. For non-smooth trajectories, the basis will not be smooth either, and the method may be inappropriate due to the presence of singularities. Especially for stochastic processes, one can usually at most guarantee smoothness P-a.s., and even though the essential supremum is taken to determine the EIM interpolation points (knots), it may be hard to distinguish between 'true' large deviations and singularities of measure zero.

Even when using an 'optimal' basis for the approximation of non-smooth functions, errors occur due to imprecise coefficients θ_m in (1). Since the EIM is based only upon few interpolation points, these coefficients may also be strongly influenced by singularities.

In Section 2, we provide necessary information about the POD, EIM, Operator EIM and DEIM that will be used to introduce two new approaches to construct affine decompositions of stochastic (non-smooth) processes. In Section 3, we introduce a Proper Orthogonal Interpolation Method (POIM) that is based on the EIM and POD and replaces the L_∞-based basis selection by an L_2-'optimal' basis. We show a connection to the DEIM and provide new error estimates for both methods. We then introduce a Least-Squares EIM (LSEIM) in Section 4 that uses more knots than basis functions. In Section 5 we provide a numerical example and show that we obtain close to optimal approximations.

2. PRELIMINARIES

In this section, we briefly review some of the basic known facts on POD and EIM that are needed in order to describe our approach.

2.1 Proper Orthogonal Decomposition (POD)

For some training set $\Xi_{\text{tr}} \subset \mathcal{P} \times \Omega$ of cardinality n_{tr} and corresponding trajectories $c(\mu,\omega), (\mu,\omega) \in \Xi_{\text{tr}}$, the POD space V_M^{POD} of dimension M is defined via the following optimization problem

$$V_M^{\text{POD}} := \arg \inf_{\substack{V_M \subset X_{\text{tr}} \\ \dim V_M = M}} n_{\text{tr}}^{-1} \cdot \sum_{(\mu,\omega)\in\Xi_{\text{tr}}} \inf_{w_M \in V_M} \|c(\mu,\omega) - w_M\|_2^2,$$

where $X_{\text{tr}} := \text{span}\{c(\mu,\omega) | (\mu,\omega) \in \Xi_{\text{tr}}\}$. It yields hierarchical spaces and is L_2-optimal in the sense of representing the trajectories of the training set in the mean.

A hierarchical basis of V_M^{POD} is given by the eigenfunctions v_m of decreasing eigenvalues λ_m, $m = 1, ..., M$, of the covariance operator $C : D \times D \to \mathbb{R}$ defined as

$$C(x_1, x_2) := n_{\text{tr}}^{-1} \cdot \sum_{(\mu,\omega)\in\Xi_{\text{tr}}} c(x_1;\mu,\omega)\, c(x_2;\mu,\omega), \;\; x_1, x_2 \in D.$$

The average approximation error of the trajectories in the training set is given by $\varepsilon_M^{\text{POD}} := \sqrt{\sum_{m>M} \lambda_m}$. For more details, see for example Kunisch and Volkwein (2002).

2.2 Empirical Interpolation Method (EIM)

We briefly review the EIM as introduced for example in Barrault et al. (2004) and Tonn (2012). We use the parametric stochastic specification that we consider in this work.

EIM: Offline-phase. A general form of the EIM offline procedure is described in Algorithm 1. It generates a so called *collateral basis* $Q_M = \{q_1, ..., q_M\}$ and corresponding interpolation points t_i, $i = 1, ..., M$, for $M = 1, ..., M_{\max}$. We will describe the main steps below.

Algorithm 1 Offline – Empirical Interpolation Method.

1: **for** $M = 1$ **to** $M_{\max}$ **do**
2: $\quad \xi \leftarrow$ *getNextBasisFunction*()
3: $\quad \xi_{M-1}^{\text{EIM}} \leftarrow$ *getApproximation*$(M-1, \xi)$
4: $\quad r_M \leftarrow \xi - \xi_{M-1}^{\text{EIM}}$
5: $\quad t_M \leftarrow \arg \operatorname{ess\,sup}_{x\in D} |r_M(x)|$
6: $\quad q_M \leftarrow r_M / r_M(t_M), \quad Q_M = \{Q_{M-1}, q_M\}$
7: **end for**

For an empty basis, the procedure *getApproximation*$(0, \xi)$ in line 3 returns zero. Otherwise, *getApproximation*(M, ξ) computes the coefficients $\boldsymbol{\theta}_M = (\theta_j(\xi))_{j=1}^M$ by solving the linear system

$$\sum_{j=1}^{M} \theta_j(\xi) q_j(t_i) = \xi(t_i),\ i = 1, ..., M, \tag{2}$$

and returns some $\xi_M^{\text{EIM}} = \sum_{j=1}^M \theta_j(\xi) q_j$. By construction, this approximation is exact at the knots $t_i, i = 1, ..., M$. Denoting $B_M := (q_j(t_i))_{i,j=1}^M$ and $\boldsymbol{\xi}_M := (t_i)_{i=1}^M$ allows to rewrite (2) as $B_M \boldsymbol{\theta}_M = \boldsymbol{\xi}_M$ and $\xi_M^{\text{EIM}} = Q_M \boldsymbol{\theta}_M$.

The procedure *getNextBasisFunction*() in line 2 uses a training set $\Xi_{\text{tr}} \subset \mathcal{P} \times \Omega$, evaluates EIM approximations $c_{M-1}^{\text{EIM}}(\mu,\omega)$ of all trajectories $c(\mu,\omega), (\mu,\omega) \in \Xi_{\text{tr}}$, and returns the trajectory that is so far worst approximated in the L_∞-sense. In line 4, the residual is evaluated. The next knot t_M is defined in line 5 in order to supremize the residual, i.e. as that point where ξ is so far worst approximated. Hence, the interpolation point selection procedure is based upon the L_∞-error. The next collateral basis function q_M is added in line 6, defined as the L_∞-normalized residual. We denote the approximation space at step M by $W_M^{\text{EIM}} := \text{span}\{q_1, ..., q_M\}$.

As mentioned before, the approximation is exact at the knots, i.e. the residual r_M is zero at $t_1, ..., t_{M-1}$. This implies that the linear system (2) is lower triangular with diagonal unity, i.e., $q_j(t_j) = 1$ and $q_j(t_i) = 0$ for $i < j$. The computational complexity of the evaluation of the EIM coefficients $\boldsymbol{\theta}_M$ is thus $\mathcal{O}(M^2)$.

EIM: Online-phase. In the online phase, sketched in Algorithm 2, we choose an $M < M_{\max}$ that is assumed

Algorithm 2 Online – Empirical Interpolation Method.

1: choose M and M^+ such that $M < M^+ \leq M_{\max}$
2: select a trajectory $c(\mu,\omega)$ for some $(\mu,\omega) \in \mathcal{P} \times \Omega$
3: $\boldsymbol{\theta}_{M^+}(\mu,\omega) \leftarrow$ *getCoefficients*$(M^+, c(\mu,\omega))$
4: evaluate approximation

$$c_M^{\mathrm{EIM}}(\mu,\omega) = \sum_{j=1}^{M} \theta_j(\mu,\omega) q_j \tag{3}$$

5: evaluate the L_∞-error estimator

$$\Delta_{M,M^+}^{\mathrm{EIM}}(\mu,\omega) = \sum_{j=M+1}^{M^+} |\theta_j(\mu,\omega)| \tag{4}$$

to be sufficiently large for a good approximation quality. Additionally, we define M^+ with $M < M^+ \leq M_{\max}$.

We then call *getCoefficients*$(M^+, c(\mu,\omega))$ that evaluates the trajectory at the knots $(t_i)_{i=1}^{M^+}$ and returns the solution $\boldsymbol{\theta}_{M^+}(\mu,\omega)$ of the lower triangular linear system

$$\sum_{j=1}^{M^+} \theta_j(\mu,\omega) q_j(t_i) = c(t_i;\mu,\omega),\ i = 1,...,M^+. \tag{5}$$

For an efficient application of the EIM, we require that evaluations of trajectories at the knots $(t_i)_{i=1}^{M^+}$ are fast, ideally of complexity $\mathcal{O}(M^+)$. Due to the lower triangular form of the linear system (5), the solutions are independent of the size of the system, i.e. $\boldsymbol{\theta}_{M+1} = (\boldsymbol{\theta}_M, \theta_{M+1})$.

We use only the first M coefficients to evaluate the approximation $c_M^{\mathrm{EIM}}(\mu,\omega)$ of the given trajectory, see line 4 of Algorithm 2. In the RBM context, it is not even necessary to evaluate $c_M^{\mathrm{EIM}}(\mu,\omega)$ in the online phase at all, only the coefficients $\boldsymbol{\theta}_M$ are used.

One often uses the additional coefficients to get error estimators. Under the assumption that the trajectory $c(\mu,\omega)$ is in $W_{M^+}^{\mathrm{EIM}}$, the quantity $\Delta_{M,M^+}^{\mathrm{EIM}}(\mu,\omega)$ from (4) provides a rigorous upper bound of the L_∞-error. The respective bound for the L_2-error would be given by $\sum_{j=M+1}^{M^+} \|q_j\|_2 |\theta_j(\mu,\omega)|$. However, the assumption $c(\mu,\omega) \in W_{M^+}^{\mathrm{EIM}}$ usually does not hold and $\Delta_{M,M^+}^{\mathrm{EIM}}$ just provides a non-rigorous (but in practice very good) estimate. For more details on EIM error estimators and more accurate bounds, see Tonn (2012).

2.3 Empirical Interpolation of Differential Operators

The DEIM (cf. Chaturantabut and Sorensen (2010)) and the empirical operator interpolation proposed in Drohmann et al. (2012) work in a similar context. Both methods generate affine decompositions of discretized differential operators. As opposed to the EIM, the basis function selection is based upon operator evaluations and the knots represent indices of the discrete operator or 'degrees of freedom'. In the online phase, the discrete operator evaluations are approximated instead of trajectories $c(\mu,\omega)$. Hence, Algorithms 1 and 2 can directly be used for the empirical operator interpolation, considering t_i to be degrees of freedom and c to be operator evaluations.

The DEIM implies further modifications of the presented algorithms. At the start of the method, one applies a POD on the discrete operator — usually using the method of snapshots (cf. Handler et al. (2006)) — generating eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots$ and corresponding orthonormal eigenfunctions $v_1, v_2, \cdots$. The further steps are sketched in Algorithm 3.

Algorithm 3 Offline – DEIM.

1: **for** $M = 1$ **to** $M_{\max}$ **do**
2: $\quad \xi_{M-1}^{\mathrm{DEIM}} \leftarrow$ *getApproximation*$(M-1, v_M)$
3: $\quad r_M \leftarrow v_M - \xi_{M-1}^{\mathrm{DEIM}}$
4: $\quad t_M \leftarrow \arg\operatorname{ess\,sup}_{x \in D} |r_M(x)|$
5: $\quad V_M = \{V_{M-1}, v_M\}$
6: **end for**

Instead of *getNextBasisFunction*() in Algorithm 1, we select the M-th POD eigenfunction in iteration M. Furthermore, in the DEIM context, we do not add residuals to the collateral basis, but the eigenfunctions. Hence, line 6 of Algorithm 1 changes to $V_M = \{V_{M-1}, v_M\}$ and the approximation space reads $W_M^{\mathrm{DEIM}} := \mathrm{span}\{v_1, ..., v_M\}$. Lines 3 to 5 remain necessary to determine the knots.

Due to the different selection method, the linear systems (2), solved in line 2 of Algorithm 3, and (5), solved online, become full, and the complexity increases to $\mathcal{O}(M^3)$ and $\mathcal{O}((M^+)^3)$, respectively.

Furthermore, the error estimator introduced in line 5 of Algorithm 2 is not valid anymore. We are not aware of any other adequate a-posteriori error bound. There are some non-rigorous a-priori average-error estimates, see Chaturantabut and Sorensen (2010).

3. A PROPER ORTHOGONAL INTERPOLATION METHOD (POIM)

In this section, we propose a Proper Orthogonal Interpolation Method (POIM) that is based on the EIM and POD. The main idea is to replace the L_∞-error based basis selection by some L_2-'optimal' procedure. Even though the method is motivated by stochastic problems, it can be applied to deterministic formulations as well and may lead to improved approximations there, too.

The method has some similarities to the DEIM. In fact, we show that we can modify the DEIM according to the POIM methodology, making it faster but still producing the same approximations. Furthermore, we show that the provided a-posteriori error estimates for the POIM can also be applied to the DEIM.

3.1 Outline of the Method

We adopt the concept of the DEIM and apply the POD to our problem in a first step. I.e., we define a training set $\Xi_{\mathrm{train}} \subset \mathcal{P} \times \Omega$, evaluate trajectories $c(\mu,\omega), (\mu,\omega) \in \Xi_{\mathrm{train}}$, and use the method of snapshots to compute POD eigenvalues $\lambda_1, ..., \lambda_{M_{\max}}$ and eigenfunctions $v_1, ..., v_{M_{\max}}$.

As for the DEIM, we select in each iteration the respective POD eigenfunction and evaluate its approximation to define the residual and the knot. However, in contrast to the DEIM, we do not directly add the POD eigenfunction to the collateral basis, we use the L_∞-normalized residual q_M, as described in Algorithm 4, line 5. This part of the

Algorithm 4 Offline – POIM.

1: **for** $M = 1$ **to** $M_{\max}$ **do**
2: $\quad \xi_{M-1}^{\mathrm{POIM}} \leftarrow$ *getApproximation*$(M-1, v_M)$
3: $\quad r_M \leftarrow v_M - \xi_{M-1}^{\mathrm{POIM}}$
4: $\quad t_M \leftarrow \arg\operatorname{ess\,sup}_{x\in D} |r_M(x)|$
5: $\quad q_M \leftarrow r_M / r_M(t_M), \quad Q_M = \{Q_{M-1}, q_M\}$
6: **end for**

algorithm has been adopted from the EIM and ensures that the linear systems (2) and (5) are still lower triangular. Therefore, the procedure *getApproximation*$(M, \cdot)$ is identical to the one used in Algorithm 1 and the online phase of the POIM is identical to the online phase of the EIM provided in Algorithm 2.

It is clear that the approximation space W_M^{POIM} is still L_2-optimal. In other words, we have

$$W_M^{\mathrm{POIM}} = \operatorname{span}\{v_1, ..., v_M\} = \operatorname{span}\{q_1, ..., q_M\}, \qquad (6)$$

which can be easily shown by induction. However, the basis Q_M is not orthonormal. The knots depend on the L_∞-error of the residual r_M, however, since r_M is a linear combination of the first M POD eigenfunctions, it is typically smooth and the knot should be adequately chosen.

3.2 Error Estimators

We can directly apply the error estimator defined in Algorithm 2, line 5, i.e., we solve the lower triangular system (5) in $\mathcal{O}((M^+)^2)$ for some $M^+ > M$ and use the additional coefficients $\theta_{M+1}, ..., \theta_{M^+}$ to evaluate $\Delta_{M,M^+}^{\mathrm{EIM}}(\mu, \omega)$.

3.3 Application within the DEIM context

As indicated in Section 2.3, the concepts of EIM and DEIM differ only slightly, using operator evaluations instead of trajectories and degrees of freedom instead of interpolation points. Hence, the POIM can directly be used to approximate operators as well. In view of (6), the approximation spaces of the DEIM and the POIM coincide. Now, we show that both methods also produce the same approximations.

Lemma 1. Let c be an arbitrary function and let $\tilde{c}_M$, $\hat{c}_M$ be approximations using M basis functions generated by the POIM and the DEIM, respectively, using the same interpolation points. Then, $\tilde{c}_M = \hat{c}_M$.

Proof. Let $Q_M = (q_1, ..., q_M)$ denote the POIM-basis and $V_M = (v_1, ..., v_M)$ the DEIM-basis. Since both span the same space, there exists a matrix $\Psi_M \in \mathbb{R}^{M\times M}$ such that $Q_M = V_M \cdot \Psi_M$. Due to the construction of Q_M in Algorithm 4, Ψ_M is upper triangular. Define $\tilde{B}_M := (q_j(t_i))_{i,j=1}^M$, $\hat{B}_M := (v_j(t_i))_{i,j=1}^M$ and $\mathbf{c}_M := (c(t_i))_{i=1}^M$. Then, $\hat{c}_M = V_M \cdot \hat{B}_M^{-1}\mathbf{c}_M$ and $\tilde{c}_M = Q_M \cdot \tilde{B}_M^{-1}\mathbf{c}_M = V_M\Psi_M \cdot \tilde{B}_M^{-1}\mathbf{c}_M$. Since $Q_M = V_M \cdot \Psi_M$, we have $\tilde{B}_M = \hat{B}_M \cdot \Psi_M$ which leads to $\tilde{c}_M = V_M\Psi_M \cdot (\hat{B}_M\Psi_M)^{-1}\mathbf{c}_M = \hat{c}_M$, proving the claim. □

It remains to show that the knots coincide.

Lemma 2. The DEIM in Algorithm 3 and the POIM in Algorithm 4 generate the same set of interpolation points.

Proof. Let $(\tilde{t}_i)_{i=1}^M$ denote the POIM-knots and $(\hat{t}_i)_{i=1}^M$ the DEIM-knots. Since for both methods, the procedure *getApproximation*$(M, \cdot)$ returns zero for $M = 0$, we have that $r_1 = v_1$ for both methods and $\tilde{t}_1 = \hat{t}_1$. Let the assertion be true for $M - 1$. Then, Lemma 1 provides that both methods return the same approximation, i.e. $\xi_{M-1}^{\mathrm{DEIM}} = \xi_{M-1}^{\mathrm{POIM}}$. Hence, both methods use the same residual to evaluate the next knot so that $\tilde{t}_M = \hat{t}_M$. □

As a consequence of the two results above, we can use the POIM instead of the DEIM, generating the same approximations, but solving only a triangular system. Hence, the complexity is only $\mathcal{O}(M^2)$. Furthermore, we can now use the EIM a-posteriori error estimates for the DEIM as well. At the same time, the DEIM a-priori error estimates are still valid since neither the approximation space is changed nor the actual approximations.

Even if an orthonormal basis would be needed and the DEIM is directly applied, we can efficiently evaluate a-posteriori error estimates. We first solve the triangular system (5) for coefficients $\boldsymbol{\theta}_{M^+}^{\mathrm{POIM}}$. Using the definitions and notations introduced in the proof of Lemma 1, it holds that

$$\boldsymbol{\theta}_{M^+}^{\mathrm{DEIM}} = \Psi_{M^+}\boldsymbol{\theta}_{M^+}^{\mathrm{POIM}} \qquad (7)$$

and we can still apply error estimator (4) with the POIM coefficients $\theta_{M+1}^{\mathrm{POIM}}, ..., \theta_{M^+}^{\mathrm{POIM}}$. The computational complexity of (7) is $\mathcal{O}((M^+)^2)$. We do not need to store two sets of basis functions but only the two triangular matrices Ψ_{M^+} and $\tilde{B}_{M^+}$ of the POIM.

4. A LEAST-SQUARES EMPIRICAL INTERPOLATION METHOD (LSEIM)

In this section, we introduce a Least-Squares Empirical Interpolation Method (LSEIM) that uses more knots than basis functions and solves a least-squares problem to evaluate $\boldsymbol{\theta}_M$. This can be combined with both EIM and POIM.

4.1 Outline of the Method

The general concept of the LSEIM offline procedure is described in Algorithm 5. We will describe the main steps below.

Algorithm 5 Offline – LSEIM.

1: **for** $M = 1$ **to** $M_{\max}$ **do**
2: $\quad \xi \leftarrow$ *getNextBasisFunction*()
3: $\quad \xi_{M-1}^{\mathrm{LSEIM}} \leftarrow$ *getApproximation*$(M-1, \xi)$
4: $\quad r_M \leftarrow \xi - \xi_{M-1}^{\mathrm{LSEIM}}$
5: $\quad (t_i)_{i=I_{M-1}+1}^{I_M} \leftarrow$ *getNextKnots*(r_M)
6: $\quad q_M \leftarrow$ *getL*$_2$*orthonormal*(r_M), $Q_M = \{Q_{M-1}, q_M\}$
7: **end for**

The procedure *getNextBasisFunction*() in line 2 returns either the so far worst approximated snapshot, as described for the EIM in Section 2.2, or the M-th POD eigenfunction, if the LSEIM is combined with the POIM.

For the LSEIM-approximation in line 3, we solve the least-squares problem

$$\sum_{i=1}^{I_M}\left(\sum_{j=1}^{M}\theta_j(\xi)q_j(t_i) - \xi(t_i)\right)^2 \to \min \qquad (8)$$

for the coefficients $\boldsymbol{\theta}_M \in \mathbb{R}^M$ and evaluate $\xi_M^{\text{LSEIM}} = Q_M \boldsymbol{\theta}_M$, where I_M denotes the number of used knots. Since the approximation and thus the residual r_M as well are no longer exact at the knots, the system is full and the complexity of solving (8) increases to $\mathcal{O}(I_M M^2)$.

There is no unique way to determine the number and location of the new knots $(t_i)_{i=I_{M-1}+1}^{I_M}$. In our examples, we used a constant number of 2 new knots per basis functions, defined by the essential infimum and the essential supremum of the residual, respectively: $t_{I_M-1} := \arg\operatorname{ess\,inf}_{x\in D} r_M(x)$ and $t_{I_M} := \arg\operatorname{ess\,sup}_{x\in D} r_M(x)$ with $I_M = 2M$. It is also possible to use iterative and adaptive selection methods (e.g., iteratively add knots until coefficients are close to 'optimal'). We will see in Section 5 that the method works very well in practice.

In line 6, we add the L_2-projection of the residual on $W_{M-1}^{\text{LSEIM}} := \operatorname{span}\{q_1, ..., q_{M-1}\}$ to the basis, i.e., we obtain an L_2-orthonormal basis. Analogously to Lemmas 1 and 2, we can show that this is equivalent to add L_∞-normalized residuals. We just replace the solutions $B_M \boldsymbol{\theta}_M = \boldsymbol{c}_M$ in the proof of Lemma 1 by the solution of a minimization problem of the form (8). However, since the system is full anyway, we prefer the L_2-orthonormal basis.

Once M is fixed in the online phase, one can compute and store a QR-decomposition and solve (8) in $\mathcal{O}(I_M M)$ for any new right-hand side. Under the assumption that the number of selected knots per iteration is $\mathcal{O}(1)$, i.e. $I_M \in \mathcal{O}(M)$, the cost increases only moderately. A drawback in the online application is the necessity to evaluate trajectories $c(\mu, \omega)$ at additional knots to get new right-hand sides, which can be expensive. However, we hope to reduce the number M of affine terms such that the overall cost decreases. Furthermore, within the RBM context, the total online complexity to assemble the system and to compute solution and error bounds is $\mathcal{O}(I_M M + MN^2 + N^3 + M^2 N^2)$, where N is the dimension of the reduced space (c.f. Patera and Rozza (2006)) and thus a small M becomes more important than a decrease of the number of knots.

4.2 Error Estimators

It is not possible to directly adopt the error estimators used for the EIM and POIM since $\boldsymbol{\theta}_{M+1} \neq (\boldsymbol{\theta}_M, \theta_{M+1})$. Instead, we separately solve (8) for M and M^+ and denote the solutions by $(\theta_j^M)_{j=1}^M$ and $(\theta_j^{M^+})_{j=1}^{M^+}$, respectively. Since Q_M is L_2-orthonormal, the L_2-error estimator is given by

$$\Delta_{M,M^+}^{\text{LSEIM}} := \sum_{j=1}^{M} \left| \theta_j^{M^+} - \theta_j^M \right| + \sum_{j=M+1}^{M^+} \left| \theta^{M_j^+} \right| \qquad (9)$$

whereas the respective L_∞-error estimator is given by $\sum_{j=1}^M \|q_j\|_\infty |\theta_j^{M^+} - \theta_j^M| + \sum_{j=M+1}^{M^+} \|q_j\|_\infty |\theta^{M_j^+}|$. The computational complexity increases compared to EIM and DEIM, even though it still is $\mathcal{O}((M^+)^2)$ for given QR-decompositions and $I_M \in \mathcal{O}(M)$.

5. NUMERICAL EXAMPLE

We consider a Wiener process $W : [0,1] \times \Omega \to \mathbb{R}$ with probability space $(\Omega, \mathcal{F}, P)$ such that $W(x;\omega) - W(y;\omega)$ is normally distributed with zero mean and variance $|x-y|$. Furthermore, we apply a parameter dependent smoothing filter $F(x, y; \mu) = \frac{1}{\sqrt{2\pi}\mu} \exp(-\frac{1}{2}\frac{(x-y)^2}{\mu^2})$ with deterministic parameters $\mu \in \mathcal{P} = [10^{-3}, 10^{-1}]$. We evaluate affine approximations of processes of the form

$$c(x; \mu, \omega) = \int_{x-1/2}^{x+1/2} F(x, y; \mu)\, W(y; \omega) dy. \qquad (10)$$

Thus, the trajectories $c(\mu, \omega) : [0, T] \to \mathbb{R}$ are continuous with increasing smoothness for larger μ. Hence, we will approximate a set of functions with different smoothness properties. Figure 1 shows random trajectories for four values of μ, logarithmically equally spaced on $\mathcal{P}$.

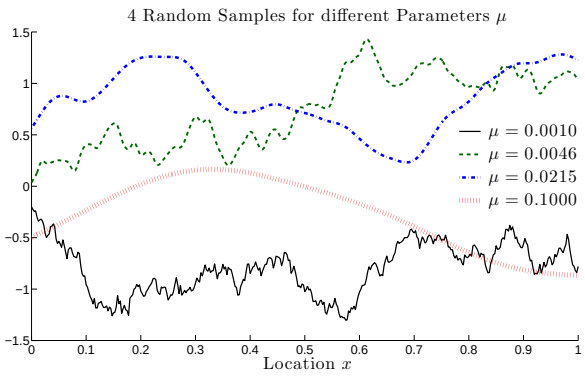

Fig. 1. Four random trajectories $c(\mu, \omega)$ as defined in (10) for different smoothing parameter configurations.

In the RBM context, we use $c(\mu, \omega)$ as a stochastic coefficient of some PDE, e.g. $\nabla \cdot (c(\mu, \omega) \nabla u(\mu, \omega)) = f$. Here, $c(\mu, \omega)$ is constructed to exemplarily represent both the case of random functions and the case of noisy input data.

We used a discretization of $\mathcal{N} = 400$ equidistant subintervals of the domain $D = [0, 1]$. For the generation of trajectories $c(\mu, \omega)$, we generated samples of the Wiener process W on the interval $[-1/2, 3/2]$ and evaluated (10). We used a training set $\Xi_{\text{tr}} \subset \mathcal{P} \times \Omega$ with a total of 3000 samples, divided on 30 logarithmically spaced parameters $\mu \in \mathcal{P}$. This training set has been used to perform the POD, EIM, DEIM, POIM and LSEIM. We used POD eigenfunctions for the generation of the LSEIM basis.

Figure 2 shows the average L_2-error of all training trajectories $c(\mu, \omega)$, $(\mu, \omega) \in \Xi_{\text{tr}}$. In this context, the POD provides the minimal error that can not be improved. We can see that the average EIM-error convergence is far from optimal whereas the LSEIM almost reaches the minimum. Even though the POIM uses the same basis as the LSEIM, the error is noticeably larger. Thus, the coefficients are not adequately evaluated. For an error tolerance of 10^{-2}, 105 basis functions and 210 knots are needed for the LSEIM whereas the POIM needs 240 knots and basis functions and the EIM more than 350. In this case, the LSEIM needs even less knots than the POIM and would considerably save online time within an RBM. As shown in Section 3.3, the POIM and DEIM produce the same results.

Figure 3 shows the maximal L_∞-error convergence of the methods. Here, the EIM and the POIM show a similar behavior. The errors decrease very slow and significant variations can be observed. For the POIM, it is clear that

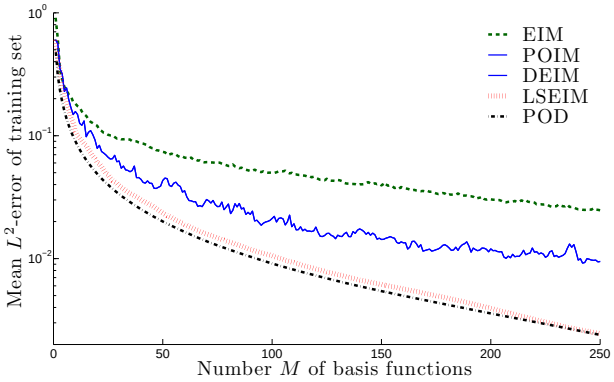

Fig. 2. Average L_2-error of training trajectories.

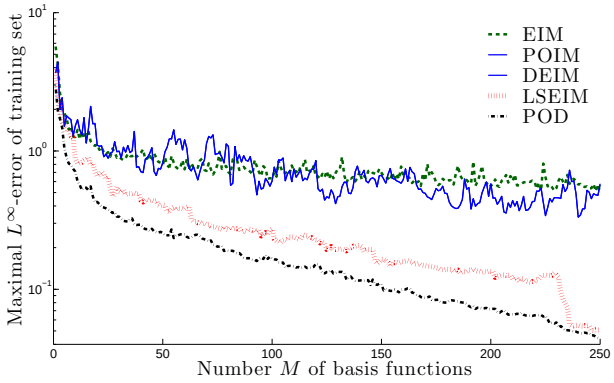

Fig. 3. Maximal L_∞-error of training trajectories.

the low convergence is caused by imprecise coefficients since the LSEIM still produces better results using the same basis. Even though the construction of the EIM is based on maximum L_∞-error minimization, the convergence is not monotonic either, since inappropriate basis functions may be selected.

In Table 1, we provide the effectivities of the introduced L_∞-error estimators, i.e. the ratio of error estimator and real error. We used a test set $\Xi_{\text{test}} \subset \mathcal{P} \times \Omega$ with a total of 3200 samples, divided on 32 logarithmically spaced parameters $\mu \in \mathcal{P}$. For all error estimators, we used 8 additional coefficients, i.e. $M^+ = M + 8$, and the table shows the minimal, average, and maximal effectivities of all test trajectories and all $M \leq \mathcal{N} - 8$. We can see that the error is not rigorous since effectivities less than one occur. However, the percentage of ineffective estimators, given in the last column, is very low, and for higher accuracy, we could increase M^+. In most cases, the estimators denote error bounds and the effectivities are rather small, where the LSEIM yields slightly better results than the EIM and the POIM, respectively.

Table 1. Effectivities of the L_∞-error estimators for 3200 test trajectories, $1 \leq M \leq \mathcal{N}-8$, and $M^+ = M+8$

	Minimal	Average	Maximal	$\% < 1$
EIM	0.373	3.025	9.148	0.022 %
POIM	0.320	3.411	14.849	0.014 %
LSEIM	0.446	2.431	6.992	0.024 %

6. CONCLUSION

We demonstrated that it is useful to add POD eigenfunctions instead of snapshots to generate the EIM basis if these may be non-smooth. We proved that the described method produces the same approximation as the DEIM with less computational cost and provided error estimators for both methods. Furthermore, we showed that using more knots than basis functions improves the approximation quality and arrives at close to optimal results.

REFERENCES

Barrault, M., Maday, Y., Nguyen, N.C., and Patera, A.T. (2004). An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math. Acad. Sci. Paris*, 339(9), 667–672.

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

Drohmann, M., Haasdonk, B., and Ohlberger, M. (2012). Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM Journal on Scientific Computing*, 34(2), A937–A969.

Ghanem, R.G. and Spanos, P.D. (1991). *Stochastic finite elements: A spectral approach.* Springer-Verlag, New York.

Haasdonk, B., Urban, K., and Wieland, B. (2012). Reduced basis methods for parametrized partial differential equations with stochastic influences using the Karhunen-Loève expansion. Preprint, Ulm University.

Handler, R.A., Housiadas, K.D., and Beris, A.N. (2006). Karhunen-Loeve representations of turbulent channel flows using the method of snapshots. *Internat. J. Numer. Methods Fluids*, 52(12), 1339–1360.

Karhunen, K. (1947). Über lineare Methoden in der Wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fennicae. Ser. A. I. Math.-Phys.*, 1947(37), 79.

Kunisch, K. and Volkwein, S. (2002). Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numer. Anal.*, 40(2), 492–515.

Loève, M. (1978). *Probability theory. II.* Springer-Verlag, New York, fourth edition. Graduate Texts in Mathematics, Vol. 46.

Patera, A.T. and Rozza, G. (2006). Reduced basis approximation and a-posteriori error estimation for parametrized partial differential equations. Version 1.0, Copyright MIT 2006.

Tonn, T. (2012). *Reduced-Basis Method (RBM) for Non-Affine Elliptic Parametrized PDEs (Motivated by Optimization in Hydromechnanics).* Ph.D. thesis, Ulm University, Ulm, Germany.

Xiu, D. and Karniadakis, G.E. (2002). The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24(2), 619–644.