

Efficient reduced basis methods for saddle point problems with applications in groundwater flow

Newsom, Craig J. and Powell, Catherine E.

2016

MIMS EPrint: **2016.60**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: <http://eprints.maths.manchester.ac.uk/>

And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

EFFICIENT REDUCED BASIS METHODS FOR SADDLE POINT PROBLEMS WITH APPLICATIONS IN GROUNDWATER FLOW

CRAIG J. NEWSUM* AND CATHERINE E. POWELL†

Abstract. Reduced basis methods (RBMs) are recommended to reduce the computational cost of solving parameter-dependent PDEs in scenarios where many choices of parameters need to be considered, for example in uncertainty quantification (UQ). A reduced basis is constructed during a computationally demanding offline (or set-up) stage that allows the user to obtain cheap approximations for parameters choices of interest, online. In this paper we consider RBMs for parameter-dependent saddle point problems, in particular the one that arises in the mixed formulation of the Darcy flow problem in groundwater flow modelling. We apply a discrete empirical interpolation method (DEIM) to approximate the inverse of the diffusion coefficient, which depends non-affinely on the system parameters. We develop an efficient RBM that exploits the DEIM approximation and combine it with a sparse grid stochastic collocation mixed finite element method (SCMFEM) to construct a surrogate solution, which then allows for efficient forward UQ. Through numerical experiments we demonstrate that significant computational savings can be made when we use the RB-DEIM-SCMFEM scheme over standard high fidelity methods. For groundwater flow problems, we provide a thorough cost assessment of the new method and show how the size of the reduced basis, and hence, the extent of the savings, depends on the statistical properties of the input parameters.

Key words. reduced basis methods, saddle point problems, stochastic collocation, uncertainty quantification

AMS subject classifications. 35R60 , 60H35, 65N22, 65N30

1. Introduction. We focus on the efficient numerical solution of parameter-dependent saddle point problems with the following abstract form. Let $D \subset \mathbb{R}^d$ be a given bounded spatial domain and let V and Q be Hilbert spaces on D , with inner products $\langle \cdot, \cdot \rangle_V$, $\langle \cdot, \cdot \rangle_Q$ and induced norms $\| \cdot \|_V$, $\| \cdot \|_Q$. In addition, let $\Gamma \subset \mathbb{R}^M$ be an M -dimensional parameter space. Given $\mathbf{y} \in \Gamma$, we want to find $(\bar{u}(\cdot, \mathbf{y}), p(\cdot, \mathbf{y})) \in V \times Q$ satisfying

$$(1.1) \quad \begin{aligned} a(\bar{u}(\cdot, \mathbf{y}), \vec{v}, \mathbf{y}) + b(\vec{v}, p(\cdot, \mathbf{y}), \mathbf{y}) &= l(\vec{v}), & \forall \vec{v} \in V, \\ b(\bar{u}(\cdot, \mathbf{y}), q, \mathbf{y}) &= m(q), & \forall q \in Q. \end{aligned}$$

Weak formulations of many well-known systems of PDEs give rise to saddle point problems of the form (1.1). For example, the Stokes equations, the steady-state Navier–Stokes equations, and mixed formulations of the Darcy flow problem, in situations where the coefficients and/or the spatial domain are represented as functions of parameters $\mathbf{y} \in \Gamma$. For a fixed $\mathbf{y} = (y_1, \dots, y_M) \in \Gamma$, the solution $(\bar{u}(\cdot, \mathbf{y}), p(\cdot, \mathbf{y}))$ can be approximated using standard techniques such as mixed finite element methods (so-called high fidelity methods).

In many complex applications (e.g., fluid flow modelling), obtaining a high fidelity approximation for even a single choice of \mathbf{y} may take several hours or days of computation time. If real-time simulations are required for many $\mathbf{y} \in \Gamma$, then using high fidelity methods is infeasible. To avoid this computational burden, reduced basis methods (RBMs), see [31, 39], have been developed. The idea is to project the high fidelity problem with dimension N_h (corresponding, say, to the number of degrees of freedom in a finite element mesh) onto a lower-dimensional subspace with dimension N_R . When $N_R \ll N_h$, we expect that the cost of solving the reduced problem will be significantly cheaper, for each $\mathbf{y} \in \Gamma$ of interest. Crucial to the success of RBMs, is that N_R is ‘not too large’, which requires that the underlying solution manifold (as a function of \mathbf{y}) is smooth. There is a large literature on RBMs for scalar elliptic PDEs. However, saddle point problems are more challenging as two compatible reduced spaces are required. RBMs for Stokes flow with

*School of Mathematics, The University of Manchester, Oxford Road, Manchester, M13 9PL, UK (craig.news@postgrad.manchester.ac.uk).

†School of Mathematics, The University of Manchester, Oxford Road, Manchester, M13 9PL, UK (c.powell@manchester.ac.uk).

parameter-dependent viscosity and spatial domains have been well studied, see [24, 41, 42], and RBMs for Stokes flow coupled with the elliptic formulation of Darcy flow are studied in [34].

RBM separates the computational work into two stages, an expensive offline stage where we carry out tasks whose costs depend on N_h , and a (hopefully) cheaper online stage where we only perform tasks whose costs depend on N_R . It is conventional to think of the offline stage as a set-up phase, carried out once ahead of the main numerical task. To facilitate this offline-online separation, it is essential that we can separate the dependence of the system inputs on $\mathbf{y} \in \Gamma$ from their dependence on $\mathbf{x} \in D$. Thus, in (1.1), we require that

$$(1.2) \quad a(\cdot, \cdot, \mathbf{y}) = \sum_{k=1}^K \theta_k(\mathbf{y}) a_k(\cdot, \cdot),$$

where $a_k : V \times V \rightarrow \mathbb{R}$ is a parameter-independent bilinear form and $\theta_k(\mathbf{y})$ is a function of \mathbf{y} , and similarly for $b(\cdot, \cdot, \mathbf{y})$. This allows us to precompute quantities that do not depend on \mathbf{y} offline. Provided that $K \ll N_h$, we can assemble and solve a reduced version of (1.1) online, at a cost that is independent of N_h . When the inputs have non-affine parameter dependence, we can use empirical interpolation methods [3, 12, 21, 28] to generate *approximations* of the form (1.2). This introduces an error, which must be controlled, and balanced against the reduced basis error.

In this paper we focus on the application of RBMs to the numerical solution of systems of PDEs with *uncertain* inputs, whose weak formulations give rise to saddle point problems of the form (1.1). If we represent the uncertain inputs as functions of M random variables (e.g. using a Karhunen-Loève expansion, see [30]), whose image is the parameter space $\Gamma \subset \mathbb{R}^M$, then choosing a particular $\mathbf{y} \in \Gamma$ corresponds to generating one sample of the inputs. Our goal is to propagate the uncertainty in the inputs through the system and approximate statistical quantities of interest (QoIs), for example the mean and variance of the solution. This is known as forward uncertainty quantification (UQ), see [45, 48]. The simplest way to perform forward UQ is to use sampling methods, which require repeated solution of (1.1) for many choices of \mathbf{y} . RBMs offer a promising way to speed up forward UQ assessments when sampling methods are used, see [10, 14, 15, 22]. They have also been applied to inverse problems [47, 17], and optimal control problems [16].

The two most commonly used sampling methods for forward UQ, see [29], are Monte Carlo (MC) and stochastic collocation methods (SCMs). The former are simple to implement but only allow for the approximation of moments. SCMs, on the other hand, provide an interpolant in both physical and parameter space. This can be used to approximate statistical moments but also acts as a *surrogate* that can be used to find approximations to $\vec{u}(\cdot, \mathbf{y})$ and $p(\cdot, \mathbf{y})$ for further choices of $\mathbf{y} \in \Gamma$ (in design or control experiments) and to approximate other QoIs. SCMs were first applied to scalar elliptic PDEs in [1, 35, 50] and have since been applied to many other problems, such as the Darcy flow model [23, 25], in a high-fidelity setting. SCMs often converge much faster than standard MC methods (whose error is $\mathcal{O}(N^{-1/2})$, where N is the number of samples). The rate of convergence of MC, however, is independent of the number M of parameters. Alternative methods have been proposed for dealing with high-dimensional problems and these can also be combined with RBMs to reduce costs. Two such examples are the ANOVA decomposition which decomposes the high-dimensional problem into low-dimensional problems, see [32], and adaptive sparse grid collocation [13]. Finally, we mention the stochastic Galerkin method [2], which is not a sampling method. This leads to a single large linear system that requires specialised solvers, see for example [37]. In this paper we develop an efficient RBM and combine it with a sparse grid stochastic collocation mixed finite element method (SCMFEM) for a class of parameter-dependent saddle point problems. A similar strategy has been employed in [5, 15, 22] for scalar elliptic PDEs.

In Section 2 we describe the reduced formulation of general parameter-dependent saddle point problems of the form (1.1). We also review sufficient conditions for the well-posedness of both high fidelity and reduced mixed finite element approximations. In Section 3 we introduce the Darcy

flow model as a special case, and then derive the linear systems that arise from the high fidelity and reduced formulations. In Section 4 we develop a reduced basis method which uses a non-standard training technique to construct a compatible pair of reduced approximation spaces in an offline stage, as well as a so-called DEIM approximation to the inverse of the diffusion coefficient. In Section 5 we describe a sparse grid SCMFEM which we then combine with the RB method and DEIM approximation. Finally, in Section 6 we present numerical experiments to demonstrate the superior performance of the new RB-DEIM-SCMFEM scheme, over the standard high fidelity SCMFEM scheme, for performing forward UQ in groundwater flow applications.

2. Parameter-dependent saddle point problems. We begin by considering the reduced formulation of general parameter-dependent saddle point problems of the form (1.1). The theoretical framework presented here is standard (e.g., see [9] and [24]).

We assume that for each $\mathbf{y} \in \Gamma$, $a(\cdot, \cdot, \mathbf{y}) : V \times V \rightarrow \mathbb{R}$ and $b(\cdot, \cdot, \mathbf{y}) : V \times Q \rightarrow \mathbb{R}$ are bounded bilinear forms and $l : V \rightarrow \mathbb{R}$ and $m : Q \rightarrow \mathbb{R}$ are also bounded. Now, for each $\mathbf{y} \in \Gamma$ we assume that $a(\cdot, \cdot, \mathbf{y})$ is coercive over

$$(2.1) \quad V_0 = \{\vec{v} \in V : b(\vec{v}, q, \mathbf{y}) = 0, \quad \forall q \in Q\} \subset V,$$

the nullspace of V . That is,

$$(2.2) \quad \alpha(\mathbf{y}) := \inf_{\vec{v} \in V_0} \frac{a(\vec{v}, \vec{v}, \mathbf{y})}{\|\vec{v}\|_V^2} > 0.$$

We also assume that for each $\mathbf{y} \in \Gamma$, $b(\cdot, \cdot, \mathbf{y})$ satisfies an inf-sup condition. Specifically,

$$(2.3) \quad \beta(\mathbf{y}) := \inf_{q \in Q} \sup_{\vec{v} \in V} \frac{b(\vec{v}, q, \mathbf{y})}{\|\vec{v}\|_V \|q\|_Q} > 0.$$

These four conditions are sufficient to guarantee that (1.1) is well-posed for a fixed $\mathbf{y} \in \Gamma$, see [9].

Next, we introduce finite-dimensional approximation spaces $V_h \subset V$ and $Q_h \subset Q$, and consider the high fidelity problem: given $\mathbf{y} \in \Gamma$, find $(\vec{u}_h(\cdot, \mathbf{y}), p_h(\cdot, \mathbf{y})) \in V_h \times Q_h$ satisfying

$$(2.4) \quad \begin{aligned} a(\vec{u}_h(\cdot, \mathbf{y}), \vec{v}_h, \mathbf{y}) + b(\vec{v}_h, p_h(\cdot, \mathbf{y}), \mathbf{y}) &= l(\vec{v}_h), & \forall \vec{v}_h \in V_h, \\ b(\vec{u}_h(\cdot, \mathbf{y}), q_h, \mathbf{y}) &= m(q_h), & \forall q_h \in Q_h. \end{aligned}$$

In this work, we assume that V_h and Q_h are finite element spaces, so that h denotes the characteristic mesh size. In the language of RBMs, we call $(\vec{u}_h(\cdot, \mathbf{y}), p_h(\cdot, \mathbf{y}))$ a *snapshot pair*. To ensure well-posedness of the high fidelity problem, we assume that

$$(2.5) \quad \alpha_h(\mathbf{y}) := \inf_{\vec{v}_h \in V_{0,h}} \frac{a(\vec{v}_h, \vec{v}_h, \mathbf{y})}{\|\vec{v}_h\|_V^2} > 0,$$

where

$$(2.6) \quad V_{0,h} = \{\vec{v}_h \in V_h : b(\vec{v}_h, q_h, \mathbf{y}) = 0, \quad \forall q_h \in Q_h\} \subset V_h,$$

is the nullspace of V_h , and secondly that

$$(2.7) \quad \beta_h(\mathbf{y}) := \inf_{q_h \in Q_h} \sup_{\vec{v}_h \in V_h} \frac{b(\vec{v}_h, q_h, \mathbf{y})}{\|\vec{v}_h\|_V \|q_h\|_Q} > 0.$$

In particular, this means that $\alpha_h(\mathbf{y})$ and $\beta_h(\mathbf{y})$ must be bounded away from zero, independently of h . Note that neither of these conditions is satisfied automatically. The spaces V_h and Q_h must be compatible for the problem at hand.

Now, let $V_R \subset V_h$ and $Q_R \subset Q_h$ denote reduced spaces, of smaller dimension than V_h and Q_h , respectively. The reduced problem is: given $\mathbf{y} \in \Gamma$, find $(\vec{u}_R(\cdot, \mathbf{y}), p_R(\cdot, \mathbf{y})) \in V_R \times Q_R$ satisfying

$$(2.8) \quad \begin{aligned} a(\vec{u}_R(\cdot, \mathbf{y}), \vec{v}_R, \mathbf{y}) + b(\vec{v}_R, p_R(\cdot, \mathbf{y}), \mathbf{y}) &= l(\vec{v}_R), & \forall \vec{v}_R \in V_R, \\ b(\vec{u}_R(\cdot, \mathbf{y}), q_R, \mathbf{y}) &= m(q_R), & \forall q_R \in Q_R. \end{aligned}$$

If the analogues of conditions (2.5) and (2.7) are satisfied, then (2.8) is well-posed. We need

$$(2.9) \quad \alpha_R(\mathbf{y}) := \inf_{\vec{v}_R \in V_{0,R}} \frac{a(\vec{v}_R, \vec{v}_R, \mathbf{y})}{\|\vec{v}_R\|_V^2} > 0,$$

where

$$(2.10) \quad V_{0,R} = \{\vec{v}_R \in V_R : b(\vec{v}_R, q_R, \mathbf{y}) = 0, \quad \forall q_R \in Q_R\} \subset V_R,$$

is the nullspace of V_R , and secondly that

$$(2.11) \quad \beta_R(\mathbf{y}) := \inf_{q_R \in Q_R} \sup_{\vec{v}_R \in V_R} \frac{b(\vec{v}_R, q_R, \mathbf{y})}{\|\vec{v}_R\|_V \|q_R\|_Q} > 0.$$

Reduced saddle point problems are challenging as (2.9) and (2.11) are not automatically satisfied for arbitrary choices of V_R and Q_R , even if $V_R \subset V_h$, $Q_R \subset Q_h$, and V_h and Q_h are chosen so that (2.4) is well-posed. It is an open question what is the ‘best’ choice of V_R and Q_R to ensure that the reduced basis errors $\|\vec{u}_h(\cdot, \mathbf{y}) - \vec{u}_R(\cdot, \mathbf{y})\|_V$ and $\|p_h(\cdot, \mathbf{y}) - p_R(\cdot, \mathbf{y})\|_Q$ decay to zero as quickly as possible, as N_R increases. Some possibilities for Stokes problems are discussed in [24].

If (2.9) and (2.11) are satisfied, we immediately obtain bounds for the reduced basis error, similar to standard high fidelity results (see [24]). Let $a_{\max}(\mathbf{y})$ and $b_{\max}(\mathbf{y})$ denote the boundedness constants for $a(\cdot, \cdot, \mathbf{y})$ and $b(\cdot, \cdot, \mathbf{y})$, respectively. Then, for a given $\mathbf{y} \in \Gamma$ we have

$$(2.12) \quad \begin{aligned} \|\vec{u}_h(\cdot, \mathbf{y}) - \vec{u}_R(\cdot, \mathbf{y})\|_V &\leq \left(1 + \frac{a_{\max}(\mathbf{y})}{\alpha_R(\mathbf{y})}\right) \inf_{\vec{v}_R \in V_{0,R}} \|\vec{u}_h(\cdot, \mathbf{y}) - \vec{v}_R(\cdot, \mathbf{y})\|_V \\ &\quad + \left(\frac{1}{\alpha_R(\mathbf{y})}\right) \inf_{q_R \in Q_R} \|p_h(\cdot, \mathbf{y}) - q_R(\cdot, \mathbf{y})\|_Q, \end{aligned}$$

c.f. [11, Theorem 12.3.7], and

$$(2.13) \quad \begin{aligned} \|p_h(\cdot, \mathbf{y}) - p_R(\cdot, \mathbf{y})\|_Q &\leq \left(\frac{a_{\max}(\mathbf{y})}{\alpha_R(\mathbf{y})}\right) \|\vec{u}_h(\cdot, \mathbf{y}) - \vec{u}_R(\cdot, \mathbf{y})\|_V \\ &\quad + \left(1 + \frac{1}{\beta_R(\mathbf{y})}\right) \inf_{q_R \in Q_R} \|p_h(\cdot, \mathbf{y}) - q_R(\cdot, \mathbf{y})\|_Q, \end{aligned}$$

c.f. [11, Theorem 12.5.12]. Thus, if $\alpha_R(\mathbf{y}) \rightarrow 0$ or $\beta_R(\mathbf{y}) \rightarrow 0$ as $N_R \rightarrow \infty$, the method is unstable.

A simple approach is to construct the reduced spaces from snapshots (solutions to the high fidelity problem). Suppose we take $\Theta = \{\mathbf{y}_1, \dots, \mathbf{y}_{N_R}\}$, a set of N_R points in the parameter domain, and the set of snapshot pairs $\{(\vec{u}_h(\cdot, \mathbf{y}_i), p_h(\cdot, \mathbf{y}_i)), i = 1, \dots, N_R\}$. Then, we could choose

$$(2.14) \quad V_R := \text{span} \{\vec{u}_h(\cdot, \mathbf{y}_i)\}_{i=1}^{N_R}, \quad Q_R := \text{span} \{p_h(\cdot, \mathbf{y}_i)\}_{i=1}^{N_R}.$$

As established in [24, 41, 42], however, this does not provide an inf-sup stable pair for (2.8). In particular, V_R is not rich enough for (2.11) to hold. Indeed, it well known from deterministic mixed finite element analysis, that the spaces should not have the same dimension; V_R must be augmented. Following the approach introduced in [42], we define an operator $T : Q_h \rightarrow V_h$ via

$$(2.15) \quad \langle Tq, \vec{v} \rangle_V = b(\vec{v}, q, \mathbf{y}), \quad \forall \vec{v} \in V_h.$$

The functions Tq are called *supremizer* functions since, given $q \in Q_h$, we have

$$(2.16) \quad Tq = \arg \sup_{\vec{v} \in V_h} \frac{b(\vec{v}, q, \mathbf{y})}{\|\vec{v}\|_V}.$$

Instead of using (2.14), we choose the reduced spaces as

$$(2.17) \quad V_R := \text{span} \{\vec{u}_h(\cdot, \mathbf{y}_i), Tp_h(\cdot, \mathbf{y}_i)\}_{i=1}^{N_R}, \quad Q_R := \text{span} \{p_h(\cdot, \mathbf{y}_i)\}_{i=1}^{N_R},$$

so that $\dim(V_R) = 2 \times \dim(Q_R) = 2 \times N_R$. With this choice, it can be shown that (2.11) is satisfied. Indeed, for all $\mathbf{y} \in \Gamma$, $\beta_R(\mathbf{y}) \geq \beta_h(\mathbf{y})$ (see [42]), so as long as $\beta_h(\mathbf{y})$ does not decay to zero as $h \rightarrow 0$ (V_h and Q_h are compatible), then neither will $\beta_R(\mathbf{y})$, as N_R increases.

Let us now briefly consider (2.9). If, for the high fidelity problem (2.4), $a(\cdot, \cdot, \mathbf{y})$ is coercive on the whole of V_h (or even V) rather than just on a strict subset $V_{0,h}$, then (2.9) follows immediately. That is, coercivity on V_h in $\|\cdot\|_V$ automatically gives coercivity on $V_{0,R}$ since $V_{0,R} \subset V_R \subset V_h \subset V$. For example, this is the case for Stokes equations, see [24, 41, 42]. In the problem we study below, the continuous and high-fidelity problems are only coercive with respect to $\|\cdot\|_V$ on V_0 and $V_{0,h}$, respectively. This does not cause a problem with well-posedness of (2.8), but the a priori error estimate (2.12) is not automatically obtained from the standard theory.

3. Darcy flow model. We now consider the Darcy flow problem, a system of PDEs that arises when modelling fluid flow through a porous medium. Typically, the permeability (or diffusion) coefficients are not known at every spatial location $\mathbf{x} \in D$ (epistemic uncertainty). Moreover, there may be measurement errors (aleatoric uncertainty). In such situations, it is common to model the unknown coefficient function as a random field $a(\mathbf{x}, \omega)$.

Let $\mathcal{D} \subset \mathbb{R}^2$ be a bounded physical domain whose boundary $\partial\mathcal{D}$ is decomposed into two disjoint subsets so that $\partial\mathcal{D} = \partial\mathcal{D}_D \cup \partial\mathcal{D}_N$. In addition, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, where Ω denotes a sample space, \mathcal{F} is a sigma algebra and \mathbb{P} is a probability measure. We want to find a velocity field $\vec{u} : \mathcal{D} \times \Omega \rightarrow \mathbb{R}^2$ and a pressure field $p : \mathcal{D} \times \Omega \rightarrow \mathbb{R}$ such that \mathbb{P} -a.s.,

$$(3.1) \quad \begin{aligned} a^{-1}(\mathbf{x}, \omega) \vec{u}(\mathbf{x}, \omega) + \nabla p(\mathbf{x}, \omega) &= 0, & \mathbf{x} \in \mathcal{D}, \\ \nabla \cdot \vec{u}(\mathbf{x}, \omega) &= f(\mathbf{x}), & \mathbf{x} \in \mathcal{D}, \\ p(\mathbf{x}, \omega) &= g(\mathbf{x}), & \mathbf{x} \in \partial\mathcal{D}_D, \\ \vec{u}(\mathbf{x}, \omega) \cdot \vec{n} &= 0, & \mathbf{x} \in \partial\mathcal{D}_N. \end{aligned}$$

We assume that $a(\mathbf{x}, \omega)$ is second-order and there exist constants a_{\min} and a_{\max} such that

$$(3.2) \quad 0 \leq a_{\min} \leq a^{-1}(\mathbf{x}, \omega) \leq a_{\max} < \infty, \quad \text{a.e. in } \mathcal{D} \times \Omega.$$

The first equation in (3.1) is Darcy's law $\vec{u} = -a\nabla p$, relating the velocity to the pressure gradient and the second equation relates the fluid mass to source and sink terms. Although both equations can be combined to obtain a standard scalar elliptic PDE for the pressure, our aim is to propagate the uncertainty through the PDE model and accurately approximate both $p(\mathbf{x}, \omega)$ and $\vec{u}(\mathbf{x}, \omega)$.

If the mean $\mu(\mathbf{x}) = \mathbb{E}[a(\mathbf{x}, \omega)]$ and covariance function $C(\mathbf{x}_1, \mathbf{x}_2)$ of $a(\mathbf{x}, \omega)$ are known, then $a(\mathbf{x}, \omega)$ can be approximated by a function of M random variables $\xi_k : \Omega \rightarrow \mathbb{R}$, by using a truncated Karhunen–Loève (KL) expansion [33],

$$(3.3) \quad a_M(\mathbf{x}, \xi(\omega)) = \mu(\mathbf{x}) + \sum_{k=1}^M \sqrt{\lambda_k} \phi_k(\mathbf{x}) \xi_k(\omega).$$

Here, λ_k and ϕ_k are eigenvalues and eigenfunctions of the integral operator associated with the covariance function, and we assume the ξ_k are independent. Alternatively, we may have another

underlying second-order random field $z(\mathbf{x}, \omega)$ with known mean and covariance, with $a(\mathbf{x}, \omega) = \exp(z(\mathbf{x}, \omega))$. This is common in groundwater flow modelling (e.g., see [23, 26]). We may then compute an approximation $a_M(\mathbf{x}, \boldsymbol{\xi}) = \exp(z_M(\mathbf{x}, \boldsymbol{\xi}))$, where $z_M(\mathbf{x}, \boldsymbol{\xi})$ is a truncated KL expansion. The important point is that the inverse of the diffusion coefficient is *some* function of a vector of independent random variables $\boldsymbol{\xi} = (\xi_1, \dots, \xi_M)$.

Instead of working with the random variable $\boldsymbol{\xi}$, with joint probability density function $\rho(\mathbf{y})$, we may work in terms of a vector of parameters $\mathbf{y} = (y_1, \dots, y_M)$, with $y_k = \xi_k(\Omega)$, which takes values in $\Gamma := \boldsymbol{\xi}(\Omega) \subset \mathbb{R}^M$. If we want to approximate *samples* of the velocity and pressure fields, we can replace the stochastic problem (3.1) with an equivalent parameter-dependent deterministic one. We assume that for each $\mathbf{y} \in \Gamma$ there exist constants $a_{\min}(\mathbf{y})$ and $a_{\max}(\mathbf{y})$ such that

$$(3.4) \quad 0 \leq a_{\min}(\mathbf{y}) \leq a_M^{-1}(\mathbf{x}, \mathbf{y}) \leq a_{\max}(\mathbf{y}) < \infty.$$

Given $\mathbf{y} \in \Gamma$, we then want to find $\vec{u}(\cdot, \mathbf{y}) : \mathcal{D} \rightarrow \mathbb{R}^2$ and $p(\cdot, \mathbf{y}) : \mathcal{D} \rightarrow \mathbb{R}$ satisfying

$$(3.5) \quad \begin{aligned} a_M^{-1}(\mathbf{x}, \mathbf{y}) \vec{u}(\mathbf{x}, \mathbf{y}) + \nabla p(\mathbf{x}, \mathbf{y}) &= 0, & \mathbf{x} \in \mathcal{D}, \\ \nabla \cdot \vec{u}(\mathbf{x}, \mathbf{y}) &= f(\mathbf{x}), & \mathbf{x} \in \mathcal{D}, \\ p(\mathbf{x}, \mathbf{y}) &= g(\mathbf{x}), & \mathbf{x} \in \partial \mathcal{D}_D, \\ \vec{u}(\mathbf{x}, \mathbf{y}) \cdot \vec{n} &= 0, & \mathbf{x} \in \partial \mathcal{D}_N. \end{aligned}$$

Now, let $Q = L^2(\mathcal{D})$ and $V = H_{0,N}(\text{div}, \mathcal{D})$, where

$$(3.6) \quad H_{0,N}(\text{div}, \mathcal{D}) = \{ \vec{v} : \mathcal{D} \rightarrow \mathbb{R}^2 : \vec{v} \in (L^2(\mathcal{D}))^2, \nabla \cdot \vec{v} \in L^2(\mathcal{D}), \vec{v} \cdot \vec{n} = 0 \text{ on } \partial \mathcal{D}_N \}$$

and define the norm

$$\| \vec{u} \|_{H(\text{div}, \mathcal{D})} = \left(\| \vec{u} \|_{L^2(\mathcal{D})}^2 + \| \nabla \cdot \vec{u} \|_{L^2(\mathcal{D})}^2 \right)^{1/2}.$$

For each $\mathbf{y} \in \Gamma$, the weak formulation of (3.5) is: find $(\vec{u}(\cdot, \mathbf{y}), p(\cdot, \mathbf{y})) \in V \times Q$ satisfying

$$(3.7) \quad \begin{aligned} \int_{\mathcal{D}} a_M^{-1}(\mathbf{x}, \mathbf{y}) \vec{u}(\mathbf{x}, \mathbf{y}) \cdot \vec{v}(\mathbf{x}) \, d\mathbf{x} - \int_{\mathcal{D}} p(\mathbf{x}, \mathbf{y}) \nabla \cdot \vec{v}(\mathbf{x}) \, d\mathbf{x} &= - \int_{\partial \mathcal{D}_D} g(\mathbf{x}) \vec{v}(\mathbf{x}) \cdot \vec{n} \, ds, \\ - \int_{\mathcal{D}} q(\mathbf{x}) \nabla \cdot \vec{u}(\mathbf{x}, \mathbf{y}) \, d\mathbf{x} &= - \int_{\mathcal{D}} f(\mathbf{x}) q(\mathbf{x}) \, d\mathbf{x}, \end{aligned}$$

for all $q(\mathbf{x}) \in Q$ and $\vec{v}(\mathbf{x}) \in V$. This is a saddle point problem of the form (1.1) with

$$(3.8) \quad a(\vec{u}(\cdot, \mathbf{y}), \vec{v}, \mathbf{y}) = \int_{\mathcal{D}} a_M^{-1}(\mathbf{x}, \mathbf{y}) \vec{u}(\mathbf{x}, \mathbf{y}) \cdot \vec{v}(\mathbf{x}) \, d\mathbf{x}, \quad b(\vec{u}(\cdot, \mathbf{y}), q) = - \int_{\mathcal{D}} q(\mathbf{x}) \nabla \cdot \vec{u}(\mathbf{x}, \mathbf{y}) \, d\mathbf{x}.$$

The above Darcy flow problem serves as a representative example from the class of parameter-dependent saddle point problems (1.1) where the \mathbf{y} dependence only appears in the bilinear form $a(\cdot, \cdot, \mathbf{y})$. Moreover, since the inverse of the diffusion coefficient appears, $a(\cdot, \cdot, \mathbf{y})$ has non-affine dependence on the parameters \mathbf{y} . For this problem, we note that $\nabla \cdot V \subset Q$ and so V_0 in (2.1) contains functions in V which are divergence-free. The coercivity condition (2.2) is satisfied on V_0 , with $\alpha(\mathbf{y}) = a_{\min}(\mathbf{y})$. The inf-sup condition (2.3) is also satisfied, see [9]. To perform forward UQ for (3.7) using sampling methods such as SCMs (see Section 5), we will need to be able to approximate $\vec{u}(\cdot, \mathbf{y})$ and $p(\cdot, \mathbf{y})$, for many different choices of \mathbf{y} , cheaply and efficiently, and to estimate statistical quantities such as

$$(3.9) \quad \mathbb{E}[\vec{u}(\mathbf{x}, \cdot)] = \int_{\Gamma} \rho(\mathbf{y}) \vec{u}(\mathbf{x}, \mathbf{y}) \, d\mathbf{y}, \quad \mathbb{E}[p(\mathbf{x}, \cdot)] = \int_{\Gamma} \rho(\mathbf{y}) p(\mathbf{x}, \mathbf{y}) \, d\mathbf{y},$$

as well as higher order moments. Using appropriate mixed finite element methods (MFEMs) allows us to approximate both the velocity and pressure to the same order of accuracy for each \mathbf{y} of interest and to ensure that mass is locally conserved on the elements, see [18, 23].

3.1. High fidelity problem. By introducing finite-dimensional subspaces $V_h \subset V$, $Q_h \subset Q$, the discrete analogue of (3.7) is a saddle point problem of the form (2.4). We use Raviart–Thomas mixed finite elements, see [9] for details, to ensure that (2.7) is satisfied. For these spaces, we have $\nabla \cdot V_h \subset Q_h$ so $V_{0,h}$ contains divergence-free functions and (2.5) is satisfied on $V_{0,h}$ with $\alpha_h(\mathbf{y}) = a_{\min}(\mathbf{y})$. In Section 6, we use lowest-order Raviart–Thomas elements, which guarantees that, if realisations of the diffusion coefficient are smooth enough, the high-fidelity discretisation errors satisfy $\|\vec{u}(\cdot, \mathbf{y}) - \vec{u}_h(\cdot, \mathbf{y})\|_{H(\text{div}, \mathcal{D})} = O(h)$ and $\|p(\cdot, \mathbf{y}) - p_h(\cdot, \mathbf{y})\|_{L^2(\mathcal{D})} = O(h)$.

Next, we derive the linear systems associated with the high fidelity problem. Given a basis $\{\phi_i(\mathbf{x}), i = 1, \dots, N_p\}$ for Q_h and a basis $\{\vec{\varphi}_i(\mathbf{x}), i = 1, \dots, N_u\}$ for V_h , where $N_u = \dim V_h$ and $N_p = \dim Q_h$, we define the matrices $A(\mathbf{y}) \in \mathbb{R}^{N_u \times N_u}$ and $B \in \mathbb{R}^{N_p \times N_u}$ with entries

$$\begin{aligned} [A(\mathbf{y})]_{ij} &= \int_{\mathcal{D}} a_M^{-1}(\mathbf{x}, \mathbf{y}) \vec{\varphi}_i(\mathbf{x}) \cdot \vec{\varphi}_j(\mathbf{x}) \, d\mathbf{x}, \quad i, j = 1, \dots, N_u, \\ B_{kj} &= - \int_{\mathcal{D}} \phi_k(\mathbf{x}) \nabla \cdot \vec{\varphi}_j(\mathbf{x}) \, d\mathbf{x}, \quad j = 1, \dots, N_u, k = 1, \dots, N_p, \end{aligned}$$

and vectors $\mathbf{g} \in \mathbb{R}^{N_u}$ and $\mathbf{f} \in \mathbb{R}^{N_p}$ with entries

$$g_i = - \int_{\partial \mathcal{D}_D} g(\mathbf{x}) \vec{\varphi}_i(\mathbf{x}) \cdot \vec{n} \, ds, \quad i = 1, \dots, N_u, \quad f_j = - \int_{\mathcal{D}} f(\mathbf{x}) \phi_j(\mathbf{x}) \, d\mathbf{x}, \quad j = 1, \dots, N_p.$$

Then, for a fixed $\mathbf{y} \in \Gamma$, the parameter-dependent high fidelity problem can be written as

$$(3.10) \quad \begin{bmatrix} A(\mathbf{y}) & B^\top \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}(\mathbf{y}) \\ \mathbf{p}(\mathbf{y}) \end{bmatrix} = \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix},$$

where $\mathbf{u}(\mathbf{y})$ and $\mathbf{p}(\mathbf{y})$ are the coefficients that represent the snapshot pair $(\vec{u}_h(\cdot, \mathbf{y}), p_h(\cdot, \mathbf{y}))$. The coefficient matrix is of size $N_h \times N_h$, where $N_h = N_u + N_p$, and is sparse. Solution methods for indefinite systems of the form (3.10) are discussed in [6]. Iterative methods such as MINRES [27, 36], with preconditioning [38, 43], which exploit sparsity, can be implemented at $\mathcal{O}(N_h)$ cost.

To construct reduced spaces V_R and Q_R that satisfy (2.11) we need to compute N_R supremizer functions Tq , defined as in (2.16), for the pressure snapshots $q = p_h(\cdot, \mathbf{y}_i) \in Q_h$, $i = 1, \dots, N_R$. Given any $q \in Q_h$, we can expand $Tq \in V_h$ as

$$(3.11) \quad Tq = \sum_{i=1}^{N_u} t_i \vec{\varphi}_i.$$

Recall that there is no \mathbf{y} dependence in the bilinear form $b(\cdot, \cdot)$. Substituting (3.11) into (2.15) and setting $\vec{v} = \vec{\varphi}_j$ and $q = \sum_{i=1}^{N_p} q_i \phi_i$ yields

$$(3.12) \quad \sum_{i=1}^{N_u} t_i \langle \vec{\varphi}_i(\mathbf{x}), \vec{\varphi}_j(\mathbf{x}) \rangle_V = - \sum_{i=1}^{N_p} \left(\int_{\mathcal{D}} \nabla \cdot \vec{\varphi}_j(\mathbf{x}) \phi_i(\mathbf{x}) \, d\mathbf{x} \right) q_i, \quad j = 1, \dots, N_u.$$

Hence, the vector of coefficients $\mathbf{t}(\mathbf{y}_i)$ associated with the supremizer function $Tp_h(\cdot, \mathbf{y}_i)$, $i = 1, \dots, N_R$, can be computed by solving the linear system

$$(3.13) \quad (M + D)\mathbf{t}(\mathbf{y}_i) = B^\top \mathbf{q}_i,$$

where the matrices M and D are defined as

$$(3.14) \quad M_{ij} = \int_{\mathcal{D}} \vec{\varphi}_i(\mathbf{x}) \cdot \vec{\varphi}_j(\mathbf{x}) \, d\mathbf{x}, \quad D_{ij} = \int_{\mathcal{D}} (\nabla \cdot \vec{\varphi}_i(\mathbf{x})) (\nabla \cdot \vec{\varphi}_j(\mathbf{x})) \, d\mathbf{x}, \quad i, j = 1, \dots, N_u.$$

The cost of solving (3.13) for each $\mathbf{y} \in \Gamma$ can be $\mathcal{O}(N_u)$ if we use iterative methods, but care must be taken. The matrix $M + D$ is a discrete representation of the $H(\text{div}, \mathcal{D})$ norm since

$$(3.15) \quad \|\vec{v}\|_{H(\text{div}, \mathcal{D})}^2 = \mathbf{v}^\top (M + D) \mathbf{v}, \quad \vec{v} \in V.$$

It does not represent an elliptic operator. In the reduced formulation of parameter-dependent Stokes and steady-state Navier–Stokes problems, computing the supremizer functions involves solving elliptic (Poisson) problems, see [24, 41, 42]. Standard multigrid methods, for example, will work well in that case, but not for (3.13). It should also be noted that the matrix $M + D$ can also be used to construct an h -optimal block-diagonal preconditioner

$$(3.16) \quad P = \begin{pmatrix} M + D & 0 \\ 0 & N \end{pmatrix},$$

for the high fidelity system (3.10) see [38], where N is the pressure mass matrix, with entries

$$(3.17) \quad N_{ks} = \int_{\mathcal{D}} \phi_k(\mathbf{x}) \cdot \phi_s(\mathbf{x}), \quad k, s = 1, \dots, N_p.$$

3.2. Reduced problem. Given reduced spaces $V_R \subset V_h$ and $Q_R \subset Q_h$, we can derive a parameter-dependent reduced problem of the form (2.8) with $a(\cdot, \cdot, \mathbf{y}) : V_R \times V_R \rightarrow \mathbb{R}$ and $b(\cdot, \cdot) : V_R \times Q_R \rightarrow \mathbb{R}$ defined as in (3.8). We construct V_R and Q_R as in (2.17) so that the inf-sup condition (2.11) is satisfied. However, it is not clear whether (2.9) is satisfied in the $H(\text{div}, \mathcal{D})$ norm, unless $\nabla \cdot V_R \subset Q_R$. Note, however, that we do have coercivity in the $L^2(D)$ norm since

$$(3.18) \quad a(\vec{v}_R, \vec{v}_R, \mathbf{y}) \geq a_{\min}(\mathbf{y}) \|\vec{v}_R\|_{L^2(D)}, \quad \forall \vec{v}_R \in V_R.$$

We assume that we have constructed two reduced basis matrices $Q_u \in \mathbb{R}^{N_u \times 2N_R}$ and $Q_p \in \mathbb{R}^{N_p \times N_R}$ that represent discrete versions of the reduced spaces V_R and Q_R , respectively, and define

$$Q = \begin{bmatrix} Q_u & 0 \\ 0 & Q_p \end{bmatrix}.$$

We describe how to construct these in Section 4.1. Then, for each $\mathbf{y} \in \Gamma$, the linear system associated with the reduced Darcy flow problem is,

$$(3.19) \quad Q^\top \begin{bmatrix} A(\mathbf{y}) & B^\top \\ B & 0 \end{bmatrix} Q \begin{bmatrix} \mathbf{u}_R(\mathbf{y}) \\ \mathbf{p}_R(\mathbf{y}) \end{bmatrix} = Q^\top \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix}.$$

The coefficient matrix in (3.19) is of size $3N_R \times 3N_R$, but is dense rather than sparse as in (3.10). Using direct methods, the cost of solving (3.19) for each $\mathbf{y} \in \Gamma$ is $\mathcal{O}(N_R^3)$. Hence, when $N_R \ll N_h$ we expect that the cost of solving (3.19) will be significantly cheaper than solving (3.10). It may be, however, that while $N_R \ll N_h$ we have $N_R^3 > N_h$, and so solving (3.19) may not actually be cheaper than solving (3.10). The size of the reduced basis N_R is key, and this is of course, problem-dependent. In [20] the authors consider using preconditioned iterative methods for solving the dense reduced systems that arise from the reduced formulation of a linear elliptic PDE. For each $\mathbf{y} \in \Gamma$ the cost is $\mathcal{O}(kN_R^2)$, where k is the number of iterations. This can be an effective strategy in some problems, as long as the number of iterations k is small.

We can assemble the system (3.19) from the high fidelity matrices and vectors via

$$A_R(\mathbf{y}) = Q_u^\top A(\mathbf{y}) Q_u, \quad B_R = Q_p^\top B Q_u, \quad \mathbf{g}_R = Q_u^\top \mathbf{g}, \quad \mathbf{f}_R = Q_p^\top \mathbf{f}.$$

However this will incur a cost that depends on N_h and so we cannot construct (3.19) directly in the online stage, for each $\mathbf{y} \in \Gamma$ of interest. To construct (3.19) in a way that the cost is independent of N_h , we need to be able to decompose the matrix $A(\mathbf{y})$ as

$$(3.20) \quad A(\mathbf{y}) = \sum_{k=1}^K A_k \theta_k(\mathbf{y}).$$

This means we need an affine expansion of $a(\cdot, \cdot, \mathbf{y})$ as in (1.2). For the Darcy flow problem, we require that the inverse of the diffusion coefficient $a_M(\mathbf{x}, \mathbf{y})$ is of the form

$$a_M^{-1}(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^K \theta_k(\mathbf{y}) a_k(\mathbf{x}).$$

If this is the case, we can define K parameter independent matrices $A_k \in \mathbb{R}^{N_u \times N_u}$ with entries

$$[A_k]_{ij} = \int_{\mathcal{D}} a_k(\mathbf{x}) \vec{\varphi}_i(\mathbf{x}) \cdot \vec{\varphi}_j(\mathbf{x}) \, d\mathbf{x}, \quad i, j = 1, \dots, N_u,$$

for $k = 1, \dots, K$. Exploiting this we have

$$A_R(\mathbf{y}) = Q_u^\top A(\mathbf{y}) Q_u = Q_u^\top \sum_{k=1}^K A_k \theta_k(\mathbf{y}) Q_u = \sum_{k=1}^K \theta_k(\mathbf{y}) Q_u^\top A_k Q_u.$$

Hence, we can precompute and store the matrices $\{Q_u^\top A_k Q_u\}_{k=1}^K$ offline, allowing for the system (3.19) to be assembled for the $\mathbf{y} \in \Gamma$ of interest online, at a cost independent of N_h .

For the Darcy flow problem, however, the coefficient $a_M(\mathbf{x}, \mathbf{y})$ appears as $a_M^{-1}(\mathbf{x}, \mathbf{y})$ in the bilinear form $a(\cdot, \cdot, \mathbf{y})$ and so we do not have a decomposition of the form (3.20). To resolve this, we apply a discrete empirical interpolation method (DEIM) to obtain an approximation $a_M^{-1}(\mathbf{x}, \mathbf{y}) \approx \sum_{k=1}^K \theta_k(\mathbf{y}) a_k(\mathbf{x})$. We describe how to do this in Section 4.2. This yields a matrix approximation $\tilde{A}(\mathbf{y}) \approx A(\mathbf{y})$ and we solve the RB-DEIM system

$$(3.21) \quad Q^\top \begin{bmatrix} \tilde{A}(\mathbf{y}) & B^\top \\ B & 0 \end{bmatrix} Q \begin{bmatrix} \tilde{\mathbf{u}}_R(\mathbf{y}) \\ \tilde{\mathbf{p}}_R(\mathbf{y}) \end{bmatrix} = Q^\top \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix}, \quad \tilde{A}(\mathbf{y}) := \sum_{k=1}^K A_k \theta_k(\mathbf{y}).$$

We use \mathbf{u}_R and \mathbf{p}_R to denote the solution to the RB system (3.19) and $\tilde{\mathbf{u}}_R$ and $\tilde{\mathbf{p}}_R$ to denote the solution to the RB-DEIM system (3.21).

To end this section, we summarise the key features of the parameter-dependent Darcy flow saddle point problem. First, the bilinear form $a(\cdot, \cdot, \mathbf{y})$ depends non-affinely on the vector of parameters \mathbf{y} . We deal with this by using DEIM, taking care to balance the DEIM error with the reduced basis error associated with V_R and Q_R (see Section 6). Second, the reduced problem does not automatically inherit coercivity from the high fidelity problem in the $H(\text{div}, \mathcal{D})$ norm, so the standard reduced basis error estimates do not immediately apply. Finally, computing the supremizer functions $Tp_h(\cdot, \mathbf{y}_i)$ that are needed to construct a pair of inf-sup stable reduced spaces requires the solution of N_R discrete $H(\text{div}, \mathcal{D})$ problems.

4. Reduced basis method offline. In this section we describe the offline procedure for constructing the reduced spaces V_R and Q_R . In particular we describe how to construct the matrices Q_u and Q_p and a DEIM approximation to the inverse of the diffusion coefficient.

4.1. Reduced basis construction. The reduced basis matrices Q_u and Q_p are given by

$$(4.1) \quad Q_u := [\mathbf{u}(\mathbf{y}_1), \dots, \mathbf{u}(\mathbf{y}_{N_R}), \mathbf{t}(\mathbf{y}_1), \dots, \mathbf{t}(\mathbf{y}_{N_R})] \in \mathbb{R}^{N_u \times 2N_R},$$

and

$$(4.2) \quad Q_p := [\mathbf{p}(\mathbf{y}_1), \dots, \mathbf{p}(\mathbf{y}_{N_R})] \in \mathbb{R}^{N_p \times N_R},$$

where $(\mathbf{u}(\mathbf{y}_i), \mathbf{p}(\mathbf{y}_i))$ are the coefficients associated with the snapshot pair $(\vec{u}_h(\cdot, \mathbf{y}_i), p_h(\cdot, \mathbf{y}_i))$ and $\mathbf{t}(\mathbf{y}_i)$ is the vector of coefficients representing the supremizer function $Th(\cdot, \mathbf{y}_i)$, $i = 1, \dots, N_R$. To construct the matrices, we have to solve N_R high fidelity mixed finite element problems and N_R discrete $H(\text{div}, \mathcal{D})$ problems. In this paper, we select the points $\{\mathbf{y}_i\}_{i=1}^{N_R}$ using a multilevel approach which is based on the algorithm presented in [22] for scalar elliptic PDEs. In our experience, it generally gives the quickest offline times (see Section 6), compared to more standard methods like greedy algorithms [39, Chapter 7], [31, Section 3.2.2] and the proper orthogonal decomposition (POD) [39, Chapter 6], [31, Section 3.2.1].

First, we must choose a training set, a finite set of points $\Theta \subset \Gamma$. These points could be chosen randomly, or deterministically. For example, Θ could be a set of sparse grid points (see Section 5). The main requirement is that Θ should be rich enough so that the resulting reduced bases yield accurate enough approximations to the high fidelity problem for any $\mathbf{y} \in \Gamma$ that may be encountered online. If the user has a priori knowledge about the set of points to be encountered online, then this information can be used to design a suitable training set. In this work, we choose nested sets of points Θ_l , indexed by a level number l , so that $\Theta_{l-1} \subset \Theta_l$.

Next, we must select an error estimator to assess the error between the high fidelity approximation and a reduced approximation for a given $\mathbf{y} \in \Gamma$. We use the error indicator

$$(4.3) \quad \Delta_R(\mathbf{y}) = \left\| \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix} - \begin{bmatrix} A(\mathbf{y}) & B^\top \\ B & 0 \end{bmatrix} \begin{bmatrix} Q_u \mathbf{u}_R(\mathbf{y}) \\ Q_p \mathbf{p}_R(\mathbf{y}) \end{bmatrix} \right\|_2 \bigg/ \left\| \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix} \right\|_2,$$

(the discrete relative residual error). This is a simple to compute approximation that, unlike a posteriori error estimators for the reduced basis error $\|\vec{u}_h(\cdot, \mathbf{y}) - \vec{u}_R(\cdot, \mathbf{y})\|_V + \|p_h(\cdot, \mathbf{y}) - p_R(\cdot, \mathbf{y})\|_Q$, does not require the computation of coercivity or inf-sup constants. However, there is no guarantee that $\Delta_R(\mathbf{y})$ is an upper bound for the true error, so our RB method is not ‘certified’ in the usual sense. To control the accuracy, we choose a tolerance $\epsilon_1 > 0$ and select a point \mathbf{y} to help construct the reduced bases only if $\Delta_R(\mathbf{y}) > \epsilon_1$.

To initialise Q_u and Q_p , we first solve the systems (3.10) and (3.13) for a single point \mathbf{y}_0 , for example the zero vector. Then, for each level $l = 1, \dots, L$, we loop through each point $\mathbf{y}_j \in \Theta_l \setminus \Theta_{l-1}$, for $j = 1, \dots, M_l$, where $M_l = |\Theta_l \setminus \Theta_{l-1}|$, solve the reduced system (3.19), and compute $\Delta_R(\mathbf{y}_j)$. If $\Delta_R(\mathbf{y}_j) > \epsilon_1$ then we solve the high fidelity problem and update the current bases immediately with a new snapshot pair corresponding to the point \mathbf{y}_j , as well as a supremizer function. Otherwise, we accept that the current reduced approximation for \mathbf{y}_j is good enough. The detailed procedure is presented in Algorithm 1. The matrices Q_u and Q_p are constructed one column at a time. For numerical stability, we orthonormalise the columns using a QR factorisation. Once the iteration has terminated, we obtain the matrices $Q_p \in \mathbb{R}^{N_p \times N_R}$ and $Q_u \in \mathbb{R}^{N_u \times 2N_R}$. Note that Algorithm 1 differs from the standard greedy algorithm as we do not update our reduced bases with the $\mathbf{y} \in \Theta_l$ that *maximizes* $\Delta_R(\mathbf{y})$ at each step. This means we only need to solve (3.19) once for each $\mathbf{y} \in \Theta_l$. In Section 6 we present numerical experiments showing that Algorithm 1 is significantly cheaper than the standard greedy algorithm and achieves similar values of N_R in the construction of the matrices Q_u and Q_p .

The costs involved in Algorithm 1 are summarised in Table 4.1. Let $N_{R_l}^j$ denote the number of high fidelity systems that have been solved after l outer iterations and j inner iterations, so

Algorithm 1 Multilevel sparse grid sampling to construct reduced basis matrices Q_u and Q_p .

- 1: Input: Maximum level number L and tolerance ϵ_1
 - 2: Set $\mathbf{y}_0 = \Theta_0$.
 - 3: Solve (3.10) to obtain $\mathbf{u}(\mathbf{y}_0)$ and $\mathbf{p}(\mathbf{y}_0)$.
 - 4: Compute supremizer function $\mathbf{t}(\mathbf{y}_0)$.
 - 5: Initialise $Q_u = [\mathbf{u}(\mathbf{y}_0), \mathbf{t}(\mathbf{y}_0)]$ and $Q_p = [\mathbf{p}(\mathbf{y}_0)]$.
 - 6: Orthogonalise Q_u and Q_p with a QR factorisation.
 - 7: **for** $l = 1 : L$ **do**
 - 8: Construct $\Theta_l \setminus \Theta_{l-1} = \{\mathbf{y}_1, \dots, \mathbf{y}_{M_l}\}$.
 - 9: **for** $j = 1 : M_l$ **do**
 - 10: Solve (3.19) with $\mathbf{y} = \mathbf{y}_j$ and compute $\Delta_R(\mathbf{y}_j)$.
 - 11: **if** $\Delta_R(\mathbf{y}_j) > \epsilon$ **then**
 - 12: Solve (3.10) to obtain $\mathbf{u}(\mathbf{y}_j)$ and $\mathbf{p}(\mathbf{y}_j)$.
 - 13: Compute supremizer function $\mathbf{t}(\mathbf{y}_j)$.
 - 14: Update $Q_u = [Q_u, \mathbf{u}(\mathbf{y}_j), \mathbf{t}(\mathbf{y}_j)]$ and $Q_p = [Q_p, \mathbf{p}(\mathbf{y}_j)]$.
 - 15: Orthogonalise Q_u and Q_p with a QR factorisation.
 - 16: **end if**
 - 17: **end for**
 - 18: **end for**
 - 19: Output: Reduced basis matrices Q_u and Q_p .
-

TABLE 4.1
Summary of costs associated with *Algorithm 1*.

Task	Cost
Assemble reduced system matrices $A_R(\mathbf{y})$	$\sum_{l=1}^L \sum_{j=1}^{M_l} \mathcal{O} \left(N_h^2 N_{R_{l-1}}^{j-1} + N_h \left(N_{R_{l-1}}^{j-1} \right)^2 \right)$
Solve reduced systems (3.19)	$\sum_{l=1}^L \sum_{j=1}^{M_l} \mathcal{O} \left(\left(N_{R_l}^j \right)^3 \right)$
Evaluate error indicator (4.3)	$\sum_{l=1}^L \sum_{j=1}^{M_l} \mathcal{O} \left(N_h N_{R_{l-1}}^{j-1} + N_h^2 \right)$
Solve (3.10) & (3.13)	$N_R \times \mathcal{O}(N_h)$
Orthonormalise columns of Q_u and Q_p	$\sum_{k=1}^{N_R} \mathcal{O}(k^2 N_h)$

that the sizes of the reduced bases at that stage are $2N_{R_l}^j$ and $N_{R_l}^j$ for the velocity and pressure, respectively. At the l^{th} outer iteration, and j^{th} inner iteration, we have to assemble and solve (3.19) with the current reduced spaces, and evaluate (4.3) for $\mathbf{y}_j \in \Theta_l \setminus \Theta_{l-1}$. Over the course of the iteration, we also need to solve N_R high fidelity and supremiser systems to augment the reduced bases, where $N_R = N_{R_L}^{M_L}$. Finally, the columns of Q_u and Q_p are orthonormalised in each iteration. The total cost therefore depends on N_h^2 and N_R^3 .

4.2. DEIM. The discrete empirical interpolation method (DEIM) [12], see also the empirical interpolation method [3],¹ allows us to generate an approximation to a parameter-dependent function $f(\mathbf{x}, \mathbf{y})$, of the form

$$(4.4) \quad f(\mathbf{x}, \mathbf{y}) \approx \sum_{k=1}^K z_k(\mathbf{x}) \theta_k(\mathbf{y}).$$

¹Contrary to what the names suggest, both of these methods are in fact ‘discrete’.

Algorithm 2 Proper orthogonal decomposition (POD), [39, Algorithm 6.1].

- 1: Input: snapshot matrix S and tolerance ϵ_2
- 2: Compute the SVD

$$S = U \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix} V^\top.$$

- 3: Find minimum K such that

$$\frac{\sum_{i=1}^K \sigma_i^2}{\sum_{i=1}^r \sigma_i^2} \geq 1 - \epsilon_2^2.$$

- 4: Output: $[\mathbf{z}_1, \dots, \mathbf{z}_K] = U(:, 1 : K)$.
-

Algorithm 3 DEIM algorithm, [39, Algorithm 10.3].

- 1: Input: snapshot matrix S and tolerance ϵ_2 .
 - 2: Do $[\mathbf{z}_1, \dots, \mathbf{z}_K] = \text{POD}(S, \epsilon_2)$.
 - 3: Compute $i = \arg \max_{i=1, \dots, m} (|\mathbf{z}_1|)$.
 - 4: Initialise $Z = [\mathbf{z}_1], P = [\mathbf{e}_i]$.
 - 5: **for** $k = 2 : K$ **do**
 - 6: Set $\mathbf{r} = \mathbf{z}_k - Z(P^\top Z)^{-1} P^\top \mathbf{z}_k$.
 - 7: Solve $i = \arg \max_{i=1, \dots, m} (|\mathbf{r}|)$.
 - 8: Update $Z = [Z, \mathbf{z}_k], P = [P, \mathbf{e}_i]$.
 - 9: **end for**
 - 10: Output: Matrix of basis functions Z and index matrix P .
-

We employ the DEIM to approximate the coefficient $a_M^{-1}(\mathbf{x}, \mathbf{y})$ in the Darcy flow model. In the following description we summarise the procedure for a general parameter-dependent function $f(\mathbf{x}, \mathbf{y})$ to simplify notation. In practice, we generally have access to a vector $\mathbf{f}(\mathbf{y}) \in \mathbb{R}^m$, corresponding to the function $f(\mathbf{x}, \mathbf{y})$ evaluated at m points \mathbf{x} in the spatial domain \mathcal{D} (for example, at points in a finite element mesh). The DEIM projects $\mathbf{f}(\mathbf{y})$ onto the subspace spanned by the K columns of a chosen basis matrix $Z \in \mathbb{R}^{m \times K}$, with $K \ll m$.

Given a basis matrix $Z \in \mathbb{R}^{m \times K}$, the DEIM approximation to $\mathbf{f}(\mathbf{y}) \in \mathbb{R}^m$ can be written as

$$\mathbf{f}_K(\mathbf{y}) = Z\mathbf{c}(\mathbf{y}) \in \mathbb{R}^m,$$

where the K columns of Z represent the functions $z_k(\mathbf{x})$ in (4.4) evaluated at m points $\mathbf{x} \in \mathcal{D}$, and the entries of $\mathbf{c}(\mathbf{y}) \in \mathbb{R}^K$ are the scalar coefficients $\theta_k(\mathbf{y})$ in (4.4) corresponding to a particular choice of \mathbf{y} . To construct Z we must first select a set of points $\{\mathbf{y}_i\}_{i=1}^n \subset \Gamma$ to act as a training set. Given the training set, we then construct the snapshot matrix

$$(4.5) \quad S = [\mathbf{f}(\mathbf{y}_1), \dots, \mathbf{f}(\mathbf{y}_n)] \in \mathbb{R}^{m \times n},$$

and compute its SVD, see [49], to obtain $S = U\Sigma V^\top$, where Σ is the diagonal matrix of singular values in descending order and U and V are orthogonal matrices. The matrix Z is chosen to be the first K columns of U , where K is the smallest number such that

$$\frac{\sum_{i=1}^K \sigma_i^2}{\sum_{i=1}^r \sigma_i^2} \geq 1 - \epsilon_2^2,$$

where ϵ_2 is a chosen tolerance, and r is the rank of S . This procedure is summarized in Algorithm 2.

Once we have a basis matrix Z we could, for a particular \mathbf{y} , determine the vector $\mathbf{c}(\mathbf{y})$ by solving $\mathbf{f}(\mathbf{y}) = Z\mathbf{c}(\mathbf{y})$. However, this is an overdetermined system, so we instead demand that the equations are satisfied for only K points $\mathbf{x} \in D$. The procedure for choosing these is called empirical interpolation and details are given in [Algorithm 3](#). Given a set of K indices, the algorithm constructs a matrix $P \in \mathbb{R}^{m \times K}$ whose columns are the columns of the $m \times m$ identity matrix associated with the chosen indices. Note that applying P^\top to a vector $\mathbf{v} \in \mathbb{R}^m$ simply extracts K components of \mathbf{v} . P does not need to be constructed. Given Z and P (or equivalently, the set of K indices), for each \mathbf{y} of interest, $\mathbf{c}(\mathbf{y})$ is found by solving the projected dense system

$$(4.6) \quad (P^\top Z) \mathbf{c}(\mathbf{y}) = P^\top \mathbf{f}(\mathbf{y}).$$

We can then write the final DEIM approximation as

$$\mathbf{f}_K(\mathbf{y}) = Z\mathbf{c}(\mathbf{y}) = Z(P^\top Z)^{-1}P^\top \mathbf{f}(\mathbf{y}).$$

The main cost associated with [Algorithms 2](#) and [3](#) is the SVD, though this is only done once offline. Given n training points $\mathbf{y} \in \Gamma$ and evaluating $f(\mathbf{x}, \mathbf{y})$ at m points $\mathbf{x} \in D$, the cost is $\mathcal{O}(mn^2)$. The number of points n in the training set should be large enough to incorporate potential variability in the system. Once the DEIM iteration has terminated, we precompute and store the parameter independent matrices and vectors

$$(4.7) \quad \{Q_u^\top A_k Q_u\}_{k=1}^K, \quad Q_p^\top B Q_u, \quad Q_p^\top \mathbf{g}, \quad Q_u \mathbf{f}.$$

The cost of this assembly is $K \times \mathcal{O}(N_h^2 N_R + N_h N_R^2)$. Online, for any given $\mathbf{y} \in \Gamma$ of interest, we additionally need to solve the dense $K \times K$ linear system (4.6) to obtain the coefficients $\mathbf{c}(\mathbf{y})$ needed to compute the matrix $\tilde{A}(\mathbf{y})$ in (3.21). The cost of this is $\mathcal{O}(K^3)$. Therefore, as long as K does not depend on N_h , the DEIM can be used online. If we also wish to evaluate the error indicator (4.3) online, then in order to make the cost independent of N_h , additional parameter independent matrices and vectors will need to be computed offline.

When we combine [Algorithms 1](#) and [3](#) there are two error tolerances, ϵ_1 and ϵ_2 . We find that choosing $\epsilon_1 = \epsilon_2$ provides a less accurate RB-DEIM approximation when compared to the RB approximation without DEIM. Note that the SVD is performed on the full snapshot matrix S , so the offline cost is not actually affected by the choice of ϵ_2 . To balance the contributions of the reduced basis and DEIM errors our recommendation is that $\epsilon_2 < \epsilon_1$ (see Section 6).

Remark 4.1. When $a_M^{-1}(\mathbf{x}, \mathbf{y})$ takes values close to zero, realisations of the DEIM approximation may be negative, making (3.21) ill-posed. We do not encounter this issue in the numerical experiments presented in [Section 6](#). However, it may arise when, for example, the standard deviation is large. In that case, the DEIM approximation should be improved by increasing K .

The conventional wisdom is that large offline costs are to be expected and that the computational savings made online are worth the price paid offline. The offline costs are often not reported. Our view is that this is problem dependent. In an online experiment where real time information is required for many choices of $\mathbf{y} \in \Gamma$, the offline cost will not necessarily be important. However, if the main task is, say, to compute moments of the solution, by combining a RB method with a sampling method such as a stochastic collocation method (SCM), then the offline/online splitting is artificial. The user knows a priori for which choices of parameters the high-fidelity problem needs to be solved. In such scenarios, the cost of constructing the reduced bases must be optimized.

5. Reduced basis collocation. In this section we outline sparse grid stochastic collocation methods for forward UQ. To begin, we describe how to construct sparse grids on the parameter domain Γ from combinations of sets of interpolation points in one dimension. The idea was first introduced by Smolyak in 1963 [46], but we follow [4]. Assuming that the M underlying random

variables in the problem are independent, we have a parameter domain of the form $\Gamma = \prod_{i=1}^M \Gamma_i$, where $\Gamma_i \subset \mathbb{R}$. Now, for $i = 1, \dots, M$, let $\Theta_i^1 = \{y_i^1, \dots, y_i^{n_i+1}\} \subset \Gamma_i$ denote a one-dimensional set of $n_i + 1$ interpolation points such that

$$(5.1) \quad n_i = \begin{cases} 0, & i = 1, \\ 2^{i-1}, & i > 1. \end{cases}$$

Given an approximation level l , the *sparse grid* on Γ is the set of points

$$(5.2) \quad \Theta_l := H(l, M) = \bigcup_{l \leq \|\mathbf{i}\|_1 < l+M} \Theta_{i_1}^1 \times \dots \times \Theta_{i_M}^1,$$

where $\mathbf{i} = (i_1, \dots, i_M)^\top \in \mathbb{N}^M$ is a multi-index. If the one-dimensional points are nested, in the sense that $\Theta_i^1 \subset \Theta_{i+1}^1$, then the formula simplifies to

$$\Theta_l := H(l, M) = \bigcup_{\|\mathbf{i}\|_1 = l+M-1} \Theta_{i_1}^1 \times \dots \times \Theta_{i_M}^1.$$

Now, given a level number l and the corresponding sparse grid $\Theta_l = \{\mathbf{y}_1, \dots, \mathbf{y}_{N_l}\}$, where $N_l = |\Theta_l|$, the high fidelity stochastic collocation mixed FEM (SCMFEM) approximation to the solution $\vec{u} : D \times \Gamma \rightarrow \mathbb{R}$ and $p : D \times \Gamma \rightarrow \mathbb{R}$ of (3.1), where $a^{-1}(\mathbf{x}, \omega)$ is replaced with $a_M^{-1}(\mathbf{x}, \mathbf{y})$ and $\mathbf{y} = \boldsymbol{\xi}(\omega)$, is given by

$$(5.3) \quad \vec{u}_{h,\Theta}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N_l} \vec{u}_h(\mathbf{x}, \mathbf{y}_i) L_i(\mathbf{y}), \quad p_{h,\Theta}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N_l} p_h(\mathbf{x}, \mathbf{y}_i) L_i(\mathbf{y}),$$

where $L_i : \Gamma \rightarrow \mathbb{R}$ is the M -dimensional Lagrange polynomial satisfying $L_i(\mathbf{y}_j) = \delta_{ij}$, for $j = 1, \dots, N_l$. We can use (5.3) as a surrogate in any experiment where it is desired to approximate the solution to (3.5) at additional points $\mathbf{y} \in \Gamma$ that are not collocation points. Computing the high fidelity SCMFEM approximation requires the solution of (3.10) for all $\mathbf{y} \in \Theta_l$. The total cost is $N_l \times \mathcal{O}(N_h)$, which is infeasible in complex applications. Approximating statistical moments, such as (3.9), can be done on-the-fly, but if we wish to use (5.3) as a surrogate then we will need to store N_l vectors of length N_h for future use.

Reduced basis methods can be used in conjunction with stochastic collocation methods, such as in [15, 22] for scalar elliptic PDEs. If we replace the high fidelity approximations in (5.3) with RB-DEIM approximations then the new RB-DEIM-SCMFEM approximations are given by

$$(5.4) \quad \vec{u}_{R,\Theta}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N_l} \vec{u}_R(\mathbf{x}, \mathbf{y}_i) L_i(\mathbf{y}), \quad p_{R,\Theta}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N_l} p_R(\mathbf{x}, \mathbf{y}_i) L_i(\mathbf{y}).$$

Using Algorithms 1 and 3, we can compute (5.4) by solving (3.21) for each $\mathbf{y} \in \Theta_l$. In Algorithm 1, we use Θ_l as the training set and the cost is $N_l \times \mathcal{O}(N_R^3 + K^3)$. As long as $N_R \ll N_h$ and $K \ll N_h$ it will be cheaper to compute (5.4) than (5.3). Again, (3.9) can be approximated on-the-fly, but if we wish to make further use of (5.4) then we need only store N_l vectors of length $3N_R$.

In the framework of stochastic collocation we can think of the construction of either (5.3) or (5.4) as an offline computation, while the evaluation of either (5.3) or (5.4) for parameters $\mathbf{y} \in \Gamma$ that are not collocation points, or some other quantity of interest, as an online computation. As long as $N_R \ll N_h$ and $N_R \ll N_l$ we expect that reduced basis collocation (with DEIM approximation if necessary) will outperform high fidelity collocation.

6. Numerical results. In this section we present some numerical results to illustrate the efficiency of using the RB-DEIM-SCMFEM scheme for the Darcy flow problem (3.1) with random coefficients. All computations are carried out using MATLAB on an Intel Xeon CPU with 2.50GHz. All linear systems, including (3.13), are solved using the MATLAB backslash function.

6.1. Non-affine test problems. Let \mathcal{D} be a square domain and set $f(\mathbf{x}) = 0$. We consider (3.1) with $a(\mathbf{x}, \omega) = \exp(z(\mathbf{x}, \omega))$, where $z(\mathbf{x}, \omega)$ is a random field whose approximation, as a function of M random variables, is $z_M(\mathbf{x}, \boldsymbol{\xi}(\omega))$, where $\boldsymbol{\xi}(\omega) = [\xi_1(\omega), \dots, \xi_M(\omega)]^\top$. Choosing independent random variables $\xi_k \sim U(-\sqrt{3}, \sqrt{3})$ leads to the parameter domain $\Gamma = [-\sqrt{3}, \sqrt{3}]^M$. Now, working in terms of $\mathbf{y} \in \Gamma$, we assume that

$$(6.1) \quad z_M(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^M \sqrt{\lambda_k} \phi_k(\mathbf{x}) y_k,$$

which has the structure of a truncated Karhunen–Loève expansion (3.3) with mean $\mu(\mathbf{x}) = 0$. We consider three examples. In the first two cases, $\lambda_k \rightarrow 0$ algebraically as $k \rightarrow \infty$, and in the third case, $\lambda_k \rightarrow 0$ exponentially as $k \rightarrow \infty$.

Example 6.1. Let $\mathcal{D} = [-1, 1]^2$, set $p = 1$ on $\{-1\} \times [-1, 1]$, $p = 0$ on $\{1\} \times [-1, 1]$, and choose homogeneous Neumann conditions for the velocity on $(-1, 1) \times \{-1, 1\}$. Here, we choose λ_k and $\phi_k(\mathbf{x})$ in (6.1) to be eigenvalues and eigenfunctions, respectively, of the covariance function

$$(6.2) \quad C(\mathbf{x}_1, \mathbf{x}_2) = \sigma^2 \exp\left(\frac{-\|\mathbf{x}_1 - \mathbf{x}_2\|_1}{c}\right), \quad \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{D},$$

where c is the correlation length, which we fix to be $c = 2$, and σ is the standard deviation of the random field $z(\mathbf{x}, \omega)$. These eigenpairs can be explicitly computed, see [33, Example 7.55].

Example 6.2. Let $\mathcal{D} = [0, 1]^2$, set $p = 1$ on $\{0\} \times [0, 1]$, $p = 0$ on $\{1\} \times [0, 1]$, and choose homogeneous Neumann conditions for the velocity on $(0, 1) \times \{0, 1\}$. This time, following an example in [19], we choose $\sqrt{\lambda_k} = \bar{\alpha} k^{-\tilde{\sigma}}$, and

$$\phi_k(\mathbf{x}) := \cos(2\pi\beta_1(k)x_1) \cos(2\pi\beta_2(k)x_2), \quad \mathbf{x} = (x_1, x_2)^T \in \mathcal{D},$$

where $k \in \mathbb{N}$, $\beta_1(k) = (k - l(k))(l(k) + 1)/2$, $\beta_2(k) = l(k) - \beta_1(k)$, $l(k) = \lfloor -1/2 + \sqrt{1/4 + 2k} \rfloor$. The parameter $\tilde{\sigma}$ controls the rate of decay of the terms $\sqrt{\lambda_k}$. We consider two cases: $\tilde{\sigma} = 2$ with $\bar{\alpha} = 0.547$, and $\tilde{\sigma} = 4$ with $\bar{\alpha} = 0.832$.

Example 6.3. We choose the spatial domain D and the boundary conditions as in Example 6.2. In addition, following [33, Example 9.37], we choose $\lambda_0 = 1/4$, $\phi_0(\mathbf{x}) = 1$, and

$$\lambda_{ij} := \frac{1}{4} \exp(-\pi(i^2 + j^2)), \quad \phi_{ij}(\mathbf{x}) := 2 \cos(i\pi x_1) \cos(j\pi x_2), \quad \mathbf{x} = (x_1, x_2)^T \in \mathcal{D},$$

for $i, j \geq 1$. The indices i and j are associated with two one-dimensional problems. The values λ_{ij} , and corresponding functions ϕ_{ij} , are then ordered in terms of a single index k as in (6.1).

In each case, choosing $a_M(\mathbf{x}, \mathbf{y}) = \exp(z_M(\mathbf{x}, \mathbf{y}))$ yields a problem of the form (3.5). For the spatial discretisation we use lowest-order square Raviart–Thomas elements associated with two meshes: the first gives $N_p = 4,096$ and $N_u = 8,192$ and the second gives $N_p = 65,536$ and $N_u = 131,072$. We construct the reduced basis matrices Q_u and Q_p using Algorithm 1 where the training set Θ_l is chosen as a set of Clenshaw–Curtis sparse grid points. Unless stated otherwise, we set $l = 4$. Since $a_M^{-1}(\mathbf{x}, \mathbf{y})$ does not depend affinely on \mathbf{y} we also construct a DEIM approximation using Algorithm 3 with tolerance $\epsilon_2 = \epsilon_1/10$ and a training set of 2000 randomly selected points $\mathbf{y} \in \Gamma$. We use the DEIM approximation to compute the matrices and vectors in (4.7) offline. This allows us to assemble and evaluate (3.21) online, at a cost that is independent of $N_h = N_u + N_p$.

TABLE 6.1

Values of N_R and K required for [Example 6.1](#) on two spatial meshes, as we vary the tolerance ϵ_1 , with fixed training set Θ_4 , and maximum relative residual errors δ_1 and δ_2 over 100 points $\mathbf{y} \in \Gamma$.

M	ϵ_1	N_R	K	δ_1	δ_2
8	1e-3	10	38	6.71e-4	6.70e-4
	1e-4	26	60	6.15e-5	6.20e-5
	1e-5	44	84	7.44e-6	7.48e-6
	1e-6	67	114	1.01e-6	1.01e-6
12	1e-3	15	55	6.08e-4	6.14e-4
	1e-4	37	87	6.79e-5	6.80e-5
	1e-5	59	124	8.80e-6	8.98e-6
	1e-6	96	166	8.52e-7	8.63e-7

(a) $N_h = 12,288$, $N_p = 4096$, $N_u = 8192$.

M	ϵ_1	N_R	K	δ_1	δ_2
8	1e-3	10	38	5.25e-4	5.17e-4
	1e-4	26	60	6.97e-5	6.99e-5
	1e-5	37	84	5.83e-6	5.82e-6
	1e-6	59	114	8.65e-7	8.66e-7
12	1e-3	14	55	4.89e-4	4.89e-4
	1e-4	30	88	8.13e-5	8.12e-5
	1e-5	53	124	6.56e-6	6.61e-6
	1e-6	85	168	9.19e-7	9.23e-7

(b) $N_h = 196,608$, $N_p = 65,536$, $N_u = 131,072$.

Comparison with high fidelity approximation. First, we consider [Example 6.1](#). We fix $\sigma = 0.25$ in [\(6.2\)](#) and choose two values for the number of terms M in [\(6.1\)](#): $M = 8$ and $M = 12$, which corresponds to retaining 87% and 89% of the integral of the variance of the underlying random field $z(\mathbf{x}, \omega)$, respectively. After constructing the reduced bases and DEIM approximation, we look at the relative residual error in the high fidelity linear system, and compare the accuracy of solutions to the RB-DEIM model [\(3.21\)](#) and the RB model [\(3.19\)](#). Specifically, we compute

$$\delta_1 = \max_{\mathbf{y} \in \Theta} \left(\left\| \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix} - \begin{bmatrix} A(\mathbf{y}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} Q_u \mathbf{u}_R(\mathbf{y}) \\ Q_p \mathbf{p}_R(\mathbf{y}) \end{bmatrix} \right\|_2 / \left\| \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix} \right\|_2 \right),$$

$$\delta_2 = \max_{\mathbf{y} \in \Theta} \left(\left\| \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix} - \begin{bmatrix} A(\mathbf{y}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} Q_u \tilde{\mathbf{u}}_R(\mathbf{y}) \\ Q_p \tilde{\mathbf{p}}_R(\mathbf{y}) \end{bmatrix} \right\|_2 / \left\| \begin{bmatrix} \mathbf{g} \\ \mathbf{f} \end{bmatrix} \right\|_2 \right),$$

as we decrease ϵ_1 , where $\Theta \subset \Gamma$ is a set of 100 randomly selected vectors of parameters. Results are presented in [Table 6.1](#). We observe that the accuracies of the RB and the RB-DEIM approximations are similar in all cases, so that the reduced basis and DEIM contributions to the error are balanced. Hence $\epsilon_2 = \epsilon_1/10$ is a good choice for this problem. Furthermore we observe that both N_R , the required dimension of Q_R , and K , the required number of terms in the DEIM approximation, are effectively independent of N_h but do increase as we decrease the tolerance ϵ_1 and increase the number of parameters M .

Next, we look at the time taken to assemble and solve the high fidelity systems [\(3.10\)](#), the reduced systems [\(3.19\)](#) associated with non-affine coefficients, and the reduced systems [\(3.21\)](#) which incorporate DEIM approximation, denoted T , t and τ respectively, while decreasing the tolerance ϵ_1 . Once the reduced bases have been constructed using the training set Θ_4 , we compute the average time taken to assemble and solve [\(3.10\)](#), [\(3.19\)](#), and [\(3.21\)](#), over an additional 100 points $\mathbf{y} \in \Gamma$. The results are presented in [Table 6.2](#). We observe that the average high fidelity time T depends on N_h but does not, of course, depend on ϵ_1 . The average RB time t depends on both N_h and ϵ_1 since we have to assemble $A_R(\mathbf{y})$ directly. The average RB-DEIM time τ does not depend on N_h but does depend on ϵ_1 . In all cases the average cost of solving the reduced system with integrated DEIM approximation is the cheapest option. When $N_h = 196,608$, the largest value of N_R is 85 and so $N_R \ll N_h$. Of course, increasing M and decreasing ϵ_1 further will cause τ to increase and at some stage, τ will become larger than T .

Now, we compare the time taken to estimate $\mathbb{E}[\tilde{u}]$ and $\mathbb{E}[p]$ using the RB-DEIM-SCMFEM approximation [\(5.4\)](#) and the high fidelity SCMFEM approximation [\(5.3\)](#) as we increase the number of collocation points N_l . The offline stage of the RB-DEIM-SCMFEM scheme involves constructing

TABLE 6.2

Average time in seconds to solve the high fidelity systems (3.10) (T), RB systems (3.19) (t) and RB-DEIM systems (3.21) (τ), over 100 points $\mathbf{y} \in \Gamma$ for Example 6.1 on two spatial meshes with training set Θ_4 .

M	ϵ_1	N_R	K	T	t	τ
8	1e-3	10	38	5.01e-2	1.65e-2	2.30e-4
	1e-4	26	60	5.02e-2	2.19e-2	6.62e-4
	1e-5	44	84	5.01e-2	2.67e-2	1.43e-3
	1e-6	67	114	5.01e-2	3.95e-2	5.04e-3
12	1e-3	15	55	5.00e-2	1.84e-2	3.41e-4
	1e-4	37	87	5.03e-2	2.55e-2	1.93e-3
	1e-5	59	124	5.00e-2	3.36e-2	3.97e-3
	1e-6	96	166	5.00e-2	7.28e-2	8.34e-3

(a) $N_h = 12,288$, $N_p = 4096$, $N_u = 8192$.

M	ϵ_1	N_R	K	T	t	τ
8	1e-3	10	38	1.23e0	2.91e-1	7.74e-4
	1e-4	26	60	1.23e0	5.30e-1	1.10e-3
	1e-5	37	84	1.22e0	8.11e-1	1.68e-3
	1e-6	59	114	1.23e0	1.18e0	4.10e-3
12	1e-3	14	55	1.23e0	3.63e-1	9.71e-4
	1e-4	30	88	1.21e0	6.35e-1	1.62e-3
	1e-5	53	124	1.22e0	1.07e0	3.59e-3
	1e-6	85	168	1.23e0	1.63e0	1.20e-2

(b) $N_h = 196,608$, $N_p = 65,536$, $N_u = 131,072$.

the reduced bases using Algorithm 1 and constructing a DEIM approximation using Algorithm 3. The online stage involves computing the approximation to $\mathbb{E}[\bar{u}]$ and $\mathbb{E}[p]$. Results are presented in Table 6.3. The times for the online stage of the RB-DEIM-SCMFEM scheme are significantly quicker than the times for the high fidelity SCMFEM scheme, since the online cost does not depend on N_h . We also see that the total time (offline + online) for the RB-DEIM-SCMFEM scheme is quicker than for the high fidelity SCMFEM scheme since for a fixed N_h , N_R does not increase significantly as N_l increases.

Cost comparison with greedy RBM. Next, we compare the performance of our RBM, which uses Algorithm 1 to construct Q_u and Q_p , with the standard greedy method. Algorithm 1 differs both in terms of the training set used and the method for selecting the points with which to form snapshots. In the standard greedy algorithm, randomly generated points are used, and in each iteration, the reduced bases are updated with the snapshot pair corresponding to the point $\mathbf{y} \in \Theta$ that maximises the estimated error $\Delta_R(\mathbf{y})$, continuing until $\Delta_R(\mathbf{y}) < \epsilon_1$ for all $\mathbf{y} \in \Theta$. First, we select the same training set used in Algorithm 1 and perform the greedy algorithm on each sparse grid level. Next, we perform the greedy algorithm with N_l randomly selected points. We use the DEIM approximation as before. In Table 6.4 we present the values of N_R , K , and the total time in seconds taken to perform the offline stage using the three approaches. We observe no more than a 10% difference in the values of N_R . However, the times recorded for Algorithm 1 are significantly lower than for the greedy approaches. The gains are substantial. This is because we only consider each point in the training set *once*, whereas the greedy approaches consider every point in each iteration. The results highlight the advantage of using a multilevel sparse grid approach, as more updates occur on the lower levels where there are fewer points.

TABLE 6.3

Time in seconds for the RB-DEIM-SCMFEM scheme with $\epsilon_1 = 10^{-5}$, and the high fidelity SCMFEM scheme, to compute the expected solution for [Example 6.1](#) on two spatial meshes as we increase N_l .

			RB-DEIM-SCMFEM				SCMFEM
M	l	N_l	N_R	K	offline	online	high fidelity
8	3	849	43	84	28.8	1.7	43.0
	4	3937	44	83	111.0	7.3	198.7
	5	15,713	45	83	430.5	28.0	781.6
	6	56,737	46	83	1,553.6	96.1	2873.6
12	3	2,649	55	123	88.6	10.9	134.8
	4	17,265	59	124	538.9	76.8	875.8
	5	93,489	61	124	2,970.3	447.3	4,777.2

(a) $N_h = 12,288$, $N_p = 4096$, $N_u = 8192$.

			RB-DEIM-SCMFEM				SCMFEM
M	l	N_l	N_R	K	offline	online	high fidelity
8	3	849	35	84	497.6	1.2	1,013.9
	4	3,937	37	84	1895.0	5.8	4,685.2
	5	15,713	40	84	7,352.9	22.2	18,835.2
	6	56,737	41	84	27,089.6	80.5	68,305.7
12	3	2,649	52	124	1548.9	8.8	3,182.5
	4	17,265	53	124	9,327.1	57.6	20,746.1
	5	93,489	53	124	53,685.8	300.0	116,049.5

(b) $N_h = 196,608$, $N_p = 65,536$, $N_u = 131,072$.

TABLE 6.4

Values of N_R and K and time in seconds for the offline stage where Q_u and Q_p are constructed using either [Algorithm 1](#), the greedy algorithm with sparse grid points, or the greedy algorithm with random points for [Example 6.1](#) with $M = 8$ on two spatial meshes.

	Algorithm 1			Greedy (sparse grid)			Greedy (random)		
ϵ_1	N_R	K	time (seconds)	N_R	K	time (seconds)	N_R	K	time (seconds)
1e-3	10	38	77.0	12	38	80.6	14	38	959.4
1e-4	26	60	93.1	27	60	203.5	25	60	1,891.9
1e-5	44	84	106.9	40	83	243.5	40	84	3,466.9
1e-6	67	114	157.7	66	114	986.1	63	113	6,414.7

(a) $N_h = 12,288$, $N_p = 4096$, $N_u = 8192$.

	Algorithm 1			Greedy (sparse grid)			Greedy (random)		
ϵ_1	N_R	K	time (seconds)	N_R	K	time (seconds)	N_R	K	time (seconds)
1e-3	10	38	1,171.7	12	38	1,251.1	10	38	11,485.3
1e-4	23	60	1,538.6	23	60	2,131.9	23	60	25,822.1
1e-5	37	83	1,887.1	37	83	4,720.9	36	84	51,806.8
1e-6	59	114	2,542.9	55	113	7,109.5	57	114	97,549.5

(b) $N_h = 196,608$, $N_p = 65,536$, $N_u = 131,072$.

Issues affecting the size of the reduced bases. We have seen that the RB-DEIM-SCMFEM scheme is efficient if $N_R \ll N_h$. Hence, a natural question is: what features of the

problem affect N_R ? Let $Z = V \times Q$ with $\|z\|_Z := \|\vec{u}\|_V + \|p\|_Q$. For a fixed $\mathbf{y} \in \Gamma$, the high fidelity solution $z_h(\cdot, \mathbf{y}) = (\vec{u}_h(\cdot, \mathbf{y}), p_h(\cdot, \mathbf{y})) \in Z_h = V_h \times Q_h$ satisfies (2.4) and the reduced solution $z_R(\cdot, \mathbf{y}) = (\vec{u}_R(\cdot, \mathbf{y}), p_R(\cdot, \mathbf{y})) \in Z_R = V_R \times Q_R$ satisfies (2.8). The quicker $\|z_h(\cdot, \mathbf{y}) - z_R(\cdot, \mathbf{y})\|_Z$ decays to zero as N_R increases, for all $\mathbf{y} \in \Gamma$ of interest, the smaller the value of N_R we will need to ensure that the reduced basis error is smaller than a prescribed tolerance ϵ . One way to obtain bounds for reduced basis errors is to seek bounds for the Kolmogorov width, see [39, Chapter 5]. For saddle point problems, see [16], this is defined as

$$(6.3) \quad d_N(\Gamma) = \inf_{Z_N \subset Z_h, \dim Z_N = N} \left(\sup_{\mathbf{y} \in \Gamma} \inf_{z_N \in Z_N} \|z_h(\cdot, \mathbf{y}) - z_N(\cdot, \mathbf{y})\|_{Z_h} \right).$$

For a fixed approximation space Z_N of dimension N , the term in parentheses measures the worst case (in a point-wise sense on Γ) best approximation error. The Kolmogorov width characterises the smallest such error over all possible choices of N -dimensional spaces Z_N . The rate at which the Kolmogorov width decays with respect to N then tells us the rate at which the reduced basis error decays, see [8, 15]. It can be shown, see [16], that a bound for the Kolmogorov width is

$$(6.4) \quad d_N(\Gamma) \leq C \sum_{k=1}^M \exp(-r_k N_k), \quad r_k = \log \left(\frac{2\tau_k}{|\Gamma_k|} \left(1 + \sqrt{1 + \frac{|\Gamma_k|^2}{4\tau_k^2}} \right) \right),$$

where τ_k is the size of the region in the complex plane into which both $\vec{u}(\cdot, \mathbf{y})$ and $p(\cdot, \mathbf{y})$ admit an analytic extension with respect to y_k . The bound (6.4) exploits a result in [1, Theorem 4.1] for the error associated with a tensor product approximation to a scalar-valued function constructed on a grid with $N_k + 1$ points in each direction so that $N = \prod_{k=1}^M (N_k + 1)$. The authors in [7] consider the saddle point problem (3.5) posed on $\mathcal{D} \times \Gamma$, where $\Gamma = [-1, 1]^M$ and $a_M^{-1}(\mathbf{x}, \mathbf{y})$ is a truncated Karhunen–Loève expansion. Assuming the mean and covariance of $a^{-1}(\mathbf{x}, \omega)$ are given, it is shown in [7, Lemma 4.1] that

$$(6.5) \quad \tau_k := \frac{a_{\min}}{\sqrt{\lambda_k} \|\phi_k(\mathbf{x})\|_{L^\infty(\mathcal{D})}}, \quad k = 1, \dots, M,$$

where $\lambda_k, \phi_k(\mathbf{x})$ are eigenpairs of the covariance associated with $a^{-1}(\mathbf{x}, \omega)$. Combined with the bound in (6.4) this suggests that the quicker the terms λ_k decay as $k \rightarrow \infty$, the smaller the space Z_N needs to be to obtain an adequate reduced basis solution. This is what we observe for our parameter-dependent saddle point problem where we construct a reduced space Z_R with dimension $N = 3N_R$. We now illustrate this.

Consider Example 6.1 with $M = 8$. Again, we build the reduced bases using the training set Θ_4 , for which $|\Theta_4| = 3,937$ (when $M = 8$). In Table 6.5, we record the values of N_R and K returned by Algorithm 1 and Algorithm 3, respectively, as we increase the standard deviation σ in (6.2) and vary the tolerance ϵ_1 . Note that σ in (6.2) controls the magnitude of the λ_k in (3.3). We observe that increasing σ , which increases λ_k , does indeed cause N_R and K to increase.

Next, we look at the effect that the rate of decay of the coefficients $\sqrt{\lambda_k}$ in (6.1) has on the values of N_R returned by Algorithm 1 and the values of K returned by Algorithm 3. We refer to $z_M(\mathbf{x}, \mathbf{y})$ from Example 6.1 as KL1, $z_M(\mathbf{x}, \mathbf{y})$ from Example 6.2 with $\tilde{\sigma} = 2$ as KL2, $z_M(\mathbf{x}, \mathbf{y})$ from Example 6.2 with $\tilde{\sigma} = 4$ as KL3, and $z_M(\mathbf{x}, \mathbf{y})$ from Example 6.3 as KL4. Note that in the KL1 case, $\sqrt{\lambda_k} = \mathcal{O}(k^{-1})$, see [33, Example 7.58], and so these are the slowest decaying terms, while the values λ_k in the KL4 case decay exponentially. These are the fastest decaying terms. Results are presented in Table 6.6 for several values of ϵ_1 . We observe that the KL1 case generally requires larger values of N_R and K , while the KL4 case requires smaller values of N_R and K .

We now examine the effect that the different rates of decay of the coefficients $\sqrt{\lambda_k}$ have on the convergence with respect to N_R of the reduced basis error. First, we compute the maximum

TABLE 6.5

Values of N_R and K required for [Example 6.1](#) with $M = 8$ on two spatial meshes, as we increase the standard deviation σ , with fixed training set Θ_4 .

σ	0.1		0.25		0.5		1	
ϵ_1	N_R	K	N_R	K	N_R	K	N_R	K
1e-3	9	23	10	38	21	58	36	90
1e-4	15	34	26	60	39	86	67	130
1e-5	27	56	44	84	66	120	108	177
1e-6	41	73	67	114	101	161	163	229

(a) $N_h = 12,288$, $N_p = 4096$, $N_u = 8192$.

σ	0.1		0.25		0.5		1	
ϵ	N_R	K	N_R	K	N_R	K	N_R	K
1e-3	8	23	10	38	19	58	29	90
1e-4	11	34	23	60	34	86	55	130
1e-5	25	56	37	84	56	120	92	178
1e-6	37	73	59	114	87	161	146	230

(b) $N_h = 196,608$, $N_p = 65,536$, $N_u = 131,072$.

TABLE 6.6

Values of N_R and K required for [Example 6.1](#) (KL1), [Example 6.2](#) with $\tilde{\sigma} = 2$ (KL2), [Example 6.2](#) with $\tilde{\sigma} = 4$ (KL3) and [Example 6.3](#) (KL4) with $M = 8$, on two spatial meshes with fixed training set Θ_4 .

	KL1		KL2		KL3		KL4	
ϵ_1	N_R	K	N_R	K	N_R	K	N_R	K
1e-3	10	38	20	22	14	12	10	11
1e-4	26	60	34	29	18	18	17	14
1e-5	44	84	42	39	22	21	20	19
1e-6	67	114	53	51	29	27	25	24

(a) $N_h = 12,288$, $N_p = 4096$, $N_u = 8192$.

	KL1		KL2		KL3		KL4	
ϵ_1	N_R	K	N_R	K	N_R	K	N_R	K
1e-3	10	38	24	22	13	12	11	11
1e-4	26	60	33	29	19	18	12	14
1e-5	37	84	44	39	22	21	16	19
1e-6	59	114	67	50	30	27	21	24

(b) $N_h = 196,608$, $N_p = 65,536$, $N_u = 131,072$.

relative pointwise error (over 100 randomly selected points $\mathbf{y} \in \Gamma$) between the reduced and high fidelity pressure and velocity approximations, which we denote by

$$(6.6) \quad \delta_R^p = \max_{\mathbf{y} \in \Theta} \left(\frac{\|p_h(\cdot, \mathbf{y}) - p_R(\cdot, \mathbf{y})\|_Q}{\|p_h(\cdot, \mathbf{y})\|_Q} \right), \quad \delta_R^u = \max_{\mathbf{y} \in \Theta} \left(\frac{\|\vec{u}_h(\cdot, \mathbf{y}) - \vec{u}_R(\cdot, \mathbf{y})\|_V}{\|\vec{u}_h(\cdot, \mathbf{y})\|_V} \right).$$

In addition, we also compute the error between the high fidelity SCMFEM and RB-DEIM-SCMFEM approximations of the mean of the pressure and the mean of the velocity, which we

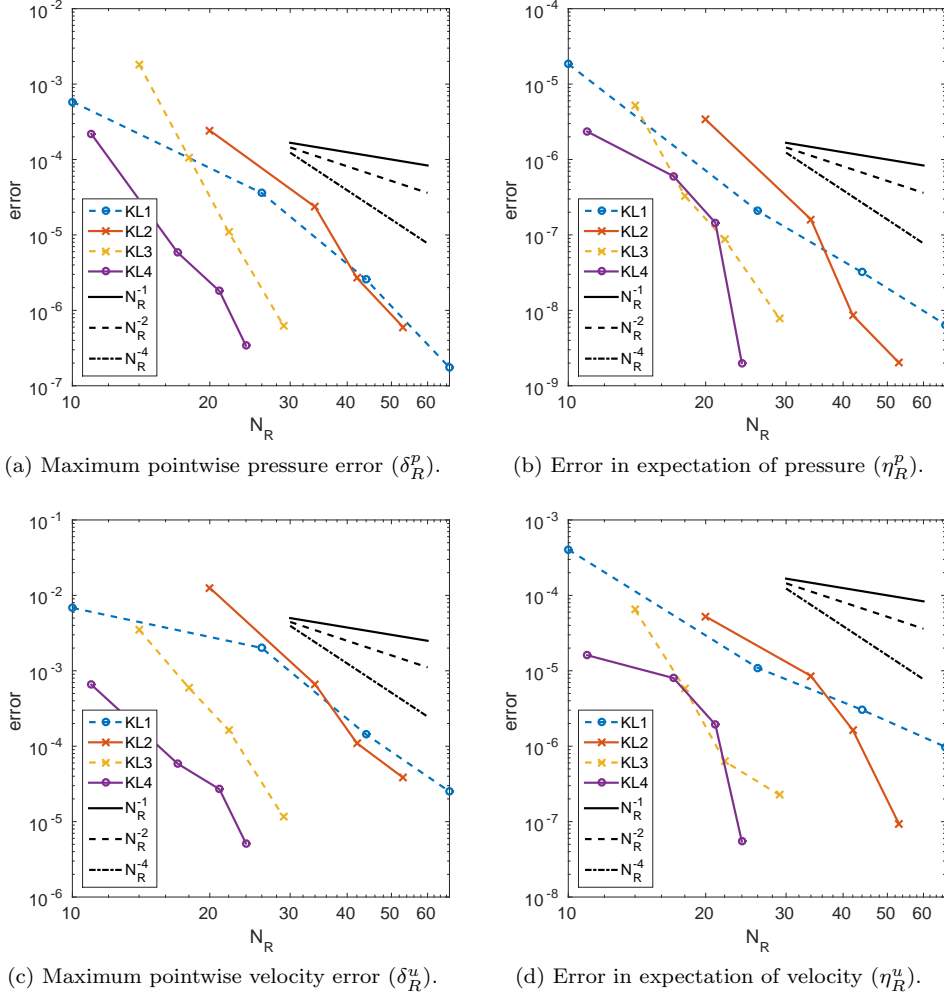


FIG. 6.1. Errors δ_R^p , δ_R^u , η_R^p , η_R^u for *Example 6.1* (KL1), *Example 6.2* with $\tilde{\sigma} = 2$ (KL2), with $\tilde{\sigma} = 4$ (KL3) and *Example 6.3* (KL4) with $M = 8$, $N_h = 12, 288$, $N_p = 4096$, $N_u = 8192$ and training set Θ_4 as we decrease ϵ_1 .

denote by

$$(6.7) \quad \eta_R^p = \frac{\|\mathbb{E}[p_{R,\Theta}] - \mathbb{E}[p_{h,\Theta}]\|_Q}{\|\mathbb{E}[p_{h,\Theta}]\|_Q}, \quad \eta_R^u = \frac{\|\mathbb{E}[\vec{u}_{R,\Theta}] - \mathbb{E}[\vec{u}_{h,\Theta}]\|_V}{\|\mathbb{E}[\vec{u}_{h,\Theta}]\|_V}.$$

We fix $N_h = 12, 288$ and use the training set Θ_4 to construct the reduced bases. We present the errors δ_R^p , δ_R^u , η_R^p , η_R^u in Figure 6.1 as we decrease ϵ_1 . For the KL1, KL2 and KL3 cases, we also indicate the decay rates of the $\sqrt{\lambda_k}$ terms for comparison. Each point on the curves in Figure 6.1 corresponds to $\epsilon_1 = 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$, respectively. We observe that for the KL1 case, the errors δ_R^p , δ_R^u , η_R^p , η_R^u decay the slowest, while for the KL4 case, the errors decay the fastest. We also observe that as N_R increases, both the pointwise approximations and the RB-DEIM-SCMFEM approximations of the mean converge. However, the velocity error δ_R^u is usually observed to be larger than ϵ_1 while the pressure error δ_R^p is smaller. It appears that the reduced velocity approximation is converging slower than the reduced pressure approximation. This was also observed in [24] for a Stokes flow problem.

Convergence comparison with greedy RBM. We now examine the decay of the reduced basis error incurred when Q_u and Q_p are constructed using [Algorithm 1](#) as opposed to a standard greedy method. Note that theoretical bounds on the Kolmogorov width assume we use the latter. In [Figure 6.2](#) we plot δ_R^u and δ_R^p as we increase N_R for both [Algorithm 1](#) and the greedy strategies described earlier. Similar rates of decay are observed for each approach.

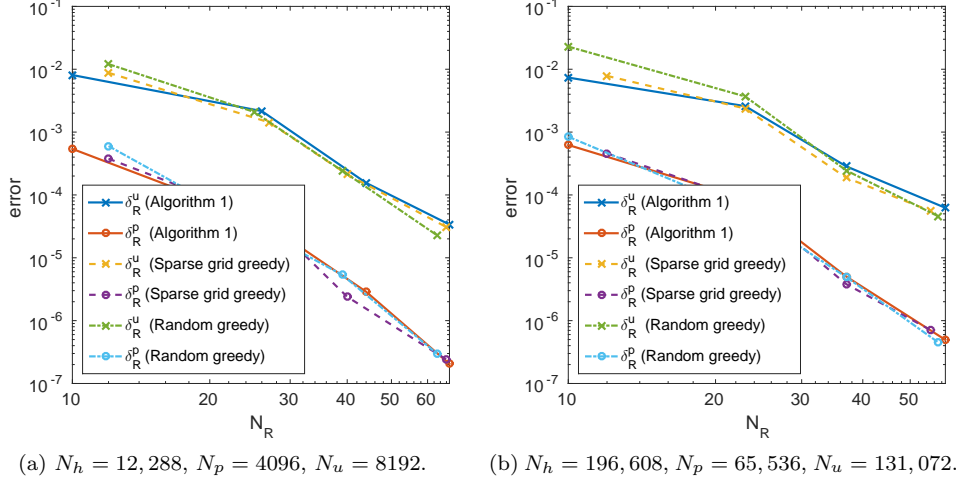


FIG. 6.2. Errors δ_R^u and δ_R^p for [Example 6.1](#) with $M = 8$ as $\epsilon_1 \rightarrow 0$, where Q_u and Q_p are constructed using [Algorithm 1](#) with training set Θ_4 , the greedy algorithm with training set Θ_4 , or the greedy algorithm with $N_4 = |\Theta_4|$ randomly generated samples.

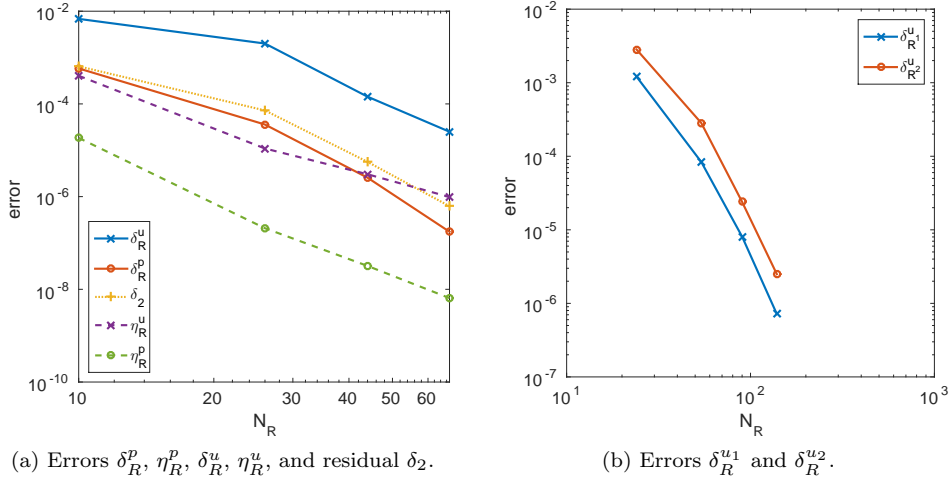


FIG. 6.3. (a) Errors $\delta_R^p, \eta_R^p, \delta_R^u, \eta_R^u$, and maximum relative residual δ_2 for [Example 6.1](#) (KL1) with $M = 8$ and $f = 0$, (b) velocity component errors δ_R^{u1} and δ_R^{u2} for [Example 6.1](#) with $M = 12$ and $f(x_1, x_2) = 2(x_1 + x_2 - x_1^2 - x_2^2)$. In both experiments, $N_h = 12,288, N_p = 4096, N_u = 8192$.

Unlike the standard greedy RBM, [Algorithm 1](#) uses the discrete relative residual error as an error indicator. In [Figure 6.3a](#) we plot the errors $\delta_R^p, \eta_R^p, \delta_R^u$ and η_R^u for [Example 6.1](#) (the KL1

case) again, as well as δ_2 , as we vary the tolerance ϵ_1 . Recall that we have $\delta_2 \approx \delta_1 \leq \epsilon_1$. We observe that δ_R^p , δ_R^u and δ_2 decay at a similar rate, although δ_R^u is larger than δ_2 . However, for the expectation, η_R^u is close to δ_2 . While it is not certified in the usual sense, our RBM does a good enough job in delivering estimates of the quantities of interest in this problem.

Coercivity. Finally, recall that it is desirable to have coercivity with respect to $\|\cdot\|_{H(\text{div}, \mathcal{D})}$ so that we obtain a priori error estimates for the velocity error using standard theory. Here, $\|\vec{v}\|_V^2 = \|\vec{v}\|_{L^2(\mathcal{D})}^2 + \|\nabla \cdot \vec{v}\|_{L^2(\mathcal{D})}^2$. For a given $\mathbf{y} \in \Gamma$ let

$$\delta_R^{u_1}(\mathbf{y}) = \frac{\|\vec{u}_h(\cdot, \mathbf{y}) - \vec{u}_R(\cdot, \mathbf{y})\|_{L^2(\mathcal{D})}}{\|\vec{u}_h(\cdot, \mathbf{y})\|_{L^2(\mathcal{D})}}, \quad \delta_R^{u_2}(\mathbf{y}) = \frac{\|\nabla \cdot (\vec{u}_h(\cdot, \mathbf{y}) - \vec{u}_R(\cdot, \mathbf{y}))\|_{L^2(\mathcal{D})}}{\|\nabla \cdot (\vec{u}_h(\cdot, \mathbf{y}))\|_{L^2(\mathcal{D})}}.$$

If both $\delta_R^{u_1}(\mathbf{y})$ and $\delta_R^{u_2}(\mathbf{y})$ decay at the same rate as N_R increases, this suggests that the velocity approximation is stable with respect to the $H(\text{div}, \mathcal{D})$ norm. In Figure 6.3b, we plot $\delta_R^{u_1}(\mathbf{y})$ and $\delta_R^{u_2}(\mathbf{y})$ for a single representative $\mathbf{y} \in \Gamma$ for a test problem and observe that this is the case.

6.2. Groundwater flow test problem. Finally, we consider a test problem of the form (3.5) with $f(\mathbf{x}) = 0$ from the software package PIFISS [44]. Let the spatial domain $\mathcal{D} = \cup_{k=1}^5 D_k$ be made up of 5 non-overlapping subdomains D_k , as depicted in Figure 6.4, which is based on the geometry of a real underground location, see [40]. On the boundary, we set $p = 1$ on the left edge (the inflow), $p = 0$ on the right edge (the outflow), and choose homogeneous Neumann conditions for the velocity on all other edges. We use the MATLAB PDE toolbox to construct two non-uniform spatial meshes. For the coefficient, let $a_M(\mathbf{x}, \mathbf{y})|_{D_k} = y_k$, $i = 1, \dots, 5$ so realisations of $a_M(\mathbf{x}, \mathbf{y})$ are piecewise constant in each subdomain D_k . For the high fidelity approximation, for each $\mathbf{y} \in \Gamma$, we apply lowest-order triangular Raviart–Thomas elements. The snapshot pair $(\vec{u}_h(\cdot, \mathbf{y}), p_h(\cdot, \mathbf{y}))$ corresponding to the specific choice of parameters $\mathbf{y} = [1.04, 17.2, 0.31, 2.60, 17.2]^\top$, which comes from the deterministic groundwater flow model described in [40], is plotted in Figure 6.5.

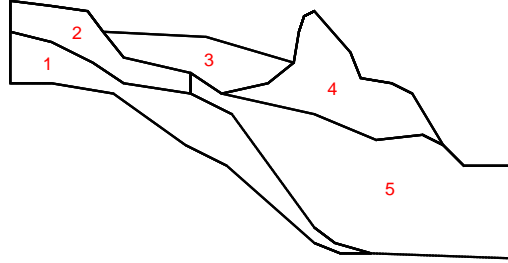


FIG. 6.4. Computational domain \mathcal{D} for the groundwater flow test problem.

Now, we treat the coefficients as uncertain, and choose $a_M(\mathbf{x}, \mathbf{y})|_{D_k} = \exp(y_k)$ for $k = 1, \dots, 5$, where each y_k is the image of a Gaussian random variable $\xi_k \sim N(\mu_k, \sigma_k)$, with mean μ_k and standard deviation σ_k chosen so that

$$\begin{aligned} \mathbb{E}[\exp(\xi_1)] &= 1.04, & \text{Var}(\exp(\xi_1)) &= 0.832, \\ \mathbb{E}[\exp(\xi_2)] &= 17.2, & \text{Var}(\exp(\xi_2)) &= 13.76, \\ \mathbb{E}[\exp(\xi_3)] &= 0.31, & \text{Var}(\exp(\xi_3)) &= 0.248, \\ \mathbb{E}[\exp(\xi_4)] &= 2.60, & \text{Var}(\exp(\xi_4)) &= 2.08, \\ \mathbb{E}[\exp(\xi_5)] &= 17.2, & \text{Var}(\exp(\xi_5)) &= 13.76. \end{aligned}$$

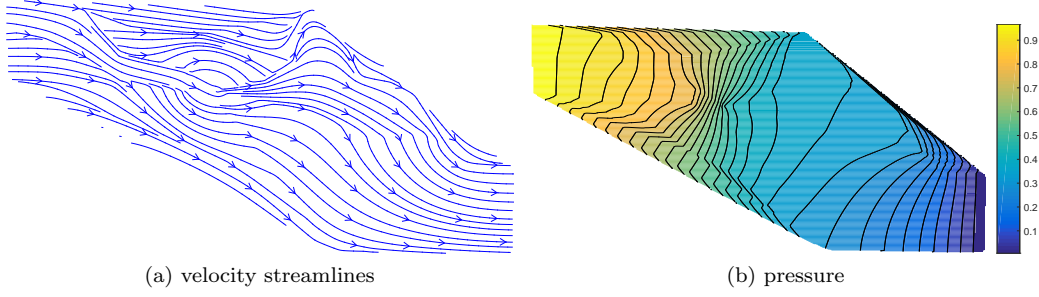


FIG. 6.5. High fidelity approximation to the groundwater flow problem with $N_h = 13,686$, $N_p = 5424$, $N_u = 8262$ corresponding to the inputs $y_1 = 1.04$, $y_2 = 17.2$, $y_3 = 0.31$, $y_4 = 2.60$, $y_5 = 17.2$.

That is, on each subdomain D_k , a_M is a lognormal random variable. Then, we have

$$a_M(\mathbf{x}, \mathbf{y})^{-1} = \sum_{k=1}^5 \exp(-y_k) 1_{D_k}(\mathbf{x}).$$

This is an affine function of $M = K = 5$ parameters and so we do not need to perform DEIM. We construct RB matrices Q_u and Q_p using Algorithm 1 with tolerance ϵ_1 and the training set Θ_l is chosen to be a sparse grid based on Gauss points. Note that Gauss points are not nested. When we update the RB matrices Q_u and Q_p at each iteration of Algorithm 1 we also compute the offline matrices and vectors in (4.7) and the quantities needed to evaluate the error indicator (4.3). Since K is small here, we find that this is the cheapest offline strategy. In Table 6.7, we record the average time taken to assemble and solve the high fidelity system (3.10) and the reduced system (3.19) as we decrease the tolerance ϵ_1 . In all cases we see that $N_R \ll N_h$ and so the cost of solving (3.19) is substantially cheaper than solving (3.10).

TABLE 6.7

Average time in seconds to assemble and solve the high fidelity systems (3.10) (T) and the RB systems (3.19) (t) over 100 points $\mathbf{y} \in \Gamma$ for the groundwater flow problem on two spatial meshes with training set Θ_4 .

N_h	N_p	N_u	ϵ_1	N_R	T	t
13,686	5424	8262	1e-3	39	7.32e-2	6.64e-4
			1e-4	60	6.93e-2	1.04e-3
			1e-5	82	6.87e-2	1.61e-3
			1e-6	106	6.70e-2	2.56e-3
217,464	86,784	130,680	1e-3	36	2.13e0	5.57e-4
			1e-4	59	2.14e0	1.06e-3
			1e-5	84	2.14e0	1.75e-3
			1e-6	108	2.08e0	5.03e-3

Finally, we compare the time taken to estimate $\mathbb{E}[\tilde{u}]$ and $\mathbb{E}[p]$ using the RB-SCMFEM approximation and the high fidelity SCMFEM approximation, as we increase the number of collocation points $N_l = |\Theta_l|$. The offline stage of the RB-SCMFEM scheme involves constructing the reduced matrices Q_u and Q_p , and the online stage involves computing the approximation to $\mathbb{E}[\tilde{u}]$ and $\mathbb{E}[p]$. The results are presented in Table 6.8. Since Gaussian random variables are unbounded, the set of sparse grid points Θ_l is also unbounded as we increase the level number l . We see that N_R increases as we increase l . In all cases, however, N_R is small enough that applying the RB-SCMFEM scheme is significantly cheaper than applying the high fidelity SCMFEM scheme. In particular,

TABLE 6.8

Time in seconds for the RB-SCMFEM scheme and high fidelity SCMFEM scheme for the groundwater flow problem on two spatial meshes with $\epsilon_1 = 10^{-5}$ as we increase N_l .

N_h	N_p	N_u			RB-SCMFEM			SCMFEM
			l	N_l	N_R	offline	online	high fidelity
13,686	5424	8262	3	351	64	23.0	0.4	24.0
			4	1,471	82	40.2	2.4	98.0
			5	5,503	95	75.6	12.0	368.3
			6	18,943	103	183.9	48.8	1,292.0
217,464	86,784	130,680	3	351	64	510.9	0.4	746.0
			4	1,471	84	795.0	2.6	3,058.0
			5	5,503	96	1,002.1	14.3	11,523.5

from the last line of Table 6.8 we see that the high fidelity SCMFEM scheme is over 10 times as expensive as our reduced basis method.

7. Concluding remarks. In this paper we described the application of reduced basis methods to parameter-dependent saddle point problems of the form (1.1) and developed an efficient method for performing forward UQ for the groundwater flow problem (3.1). We presented a non-standard training algorithm for constructing a pair of compatible reduced bases that uses a multilevel approach based on sparse grid points, and a simple error indicator. We also described how to combine this with a DEIM approximation to the inverse of the diffusion coefficient. We demonstrated that with these variations we gain substantially lower offline times when compared to the standard greedy algorithm. We showed that assembling and solving the RB-DEIM system (3.21) is much cheaper than assembling and solving the high fidelity system (3.10). We combined the reduced basis method with a stochastic collocation mixed finite element method and implemented the resulting RB-DEIM-SCMFEM scheme for a range of test problems, demonstrating that significant computational savings can be made over standard high fidelity SCMFEM schemes, even when the offline times are taken into account. Our experiments reveal, however, that reduced basis methods are most effective for performing forward UQ for PDE models with quickly decaying coefficient terms, a modest number of parameters and low variances.

REFERENCES

- [1] I. BABUŠKA, F. NOBILE, AND R. TEMPONE, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 1005–1034, doi:10.1137/050645142.
- [2] I. BABUŠKA, R. TEMPONE, AND G. E. ZOURARIS, *Galerkin finite element approximations of stochastic elliptic partial differential equations*, SIAM Journal on Numerical Analysis, 42 (2004), pp. 800–825, doi:10.1137/S0036142902418680.
- [3] M. BARRAULT, Y. MADAY, N. C. NGUYEN, AND A. T. PATERA, *An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations*, Comptes Rendus Mathematique, 339 (2004), pp. 667–672, doi:10.1016/j.crma.2004.08.006.
- [4] V. BARTHELMANN, E. NOVAK, AND K. RITTER, *High dimensional polynomial interpolation on sparse grids*, Advances in Computational Mathematics, 12 (2000), pp. 273–288, doi:10.1023/A:1018977404843.
- [5] P. BENNER AND M. W. HESS, *Reduced basis modeling for uncertainty quantification of electromagnetic problems in stochastically varying domains*, in Scientific Computing in Electrical Engineering, Springer, 2016, pp. 215–222.
- [6] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta numerica, 14 (2005), pp. 1–137, doi:10.1017/S0962492904000212.
- [7] A. BESPALOV, C. E. POWELL, AND D. J. SILVESTER, *A priori error analysis of stochastic Galerkin mixed approximations of elliptic pdes with random data*, SIAM Journal on Numerical Analysis, 50 (2012), pp. 2039–2063.

- [8] P. BINEV, A. COHEN, W. DAHMEN, R. DEVORE, G. PETROVA, AND P. WOJTASZCZYK, *Convergence rates for greedy algorithms in reduced basis methods*, SIAM journal on mathematical analysis, 43 (2011), pp. 1457–1472.
- [9] D. BOFFI, F. BREZZI, AND M. FORTIN, *Mixed finite element methods and applications*, Springer, 2013.
- [10] S. BOYAVAL, C. LE BRIS, T. LELIVRE, Y. MADAY, N. NGUYEN, AND A. PATERA, *Reduced basis techniques for stochastic problems*, Archives of Computational Methods in Engineering, 17 (2010), pp. 435–454, doi:10.1007/s11831-010-9056-z.
- [11] S. C. BRENNER AND R. SCOTT, *The mathematical theory of finite element methods*, vol. 15, Springer Science & Business Media, 2008.
- [12] S. CHATURANTABUT AND D. C. SORENSEN, *Nonlinear model reduction via discrete empirical interpolation*, SIAM Journal on Scientific Computing, 32 (2010), pp. 2737–2764.
- [13] P. CHEN AND A. QUARTERONI, *A new algorithm for high-dimensional uncertainty quantification based on dimension-adaptive sparse grid approximation and reduced basis methods*, Journal of Computational Physics, 298 (2015), pp. 176–193.
- [14] P. CHEN, A. QUARTERONI, AND G. ROZZA, *A weighted reduced basis method for elliptic partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 51 (2013), pp. 3163–3185.
- [15] P. CHEN, A. QUARTERONI, AND G. ROZZA, *Comparison between reduced basis and stochastic collocation methods for elliptic problems*, Journal of Scientific Computing, 59 (2014), pp. 187–216, doi:10.1007/s10915-013-9764-2.
- [16] P. CHEN, A. QUARTERONI, AND G. ROZZA, *Multilevel and weighted reduced basis method for stochastic optimal control problems constrained by Stokes equations*, Numerische Mathematik, 133 (2016), pp. 67–102.
- [17] P. CHEN AND C. SCHWAB, *Sparse-grid, reduced-basis bayesian inversion*, Computer Methods in Applied Mechanics and Engineering, 297 (2015), pp. 84–115.
- [18] K. CLIFFE, I. G. GRAHAM, R. SCHEICHL, AND L. STALS, *Parallel computation of flow in heterogeneous media modelled by mixed finite elements*, Journal of Computational Physics, 164 (2000), pp. 258–282.
- [19] M. EIGEL, C. J. GITTELSON, C. SCHWAB, AND E. ZANDER, *Adaptive stochastic Galerkin FEM*, Computer Methods in Applied Mechanics and Engineering, 270 (2014), pp. 247–269.
- [20] H. C. ELMAN AND V. FORSTALL, *Preconditioning techniques for reduced basis methods for parameterized elliptic partial differential equations*, SIAM Journal on Scientific Computing, 37 (2015), pp. S177–S194, doi:10.1137/140970859.
- [21] H. C. ELMAN AND V. FORSTALL, *Numerical solution of the steady-state Navier-Stokes equations using empirical interpolation methods*, arXiv preprint arXiv:1605.06138, (2016).
- [22] H. C. ELMAN AND Q. LIAO, *Reduced basis collocation methods for partial differential equations with random coefficients*, SIAM/ASA Journal on Uncertainty Quantification, 1 (2013), pp. 192–217, doi:10.1137/120881841.
- [23] B. GANIS, H. KLIE, M. F. WHEELER, T. WILDEY, I. YOTOV, AND D. ZHANG, *Stochastic collocation and mixed finite elements for flow in porous media*, Computer methods in applied mechanics and engineering, 197 (2008), pp. 3547–3559.
- [24] A.-L. GERNER AND K. VEROY, *Certified reduced basis methods for parametrized saddle point problems*, SIAM Journal on Scientific Computing, 34 (2012), pp. A2812–A2836.
- [25] A. D. GORDON AND C. E. POWELL, *On solving stochastic collocation systems with algebraic multigrid*, IMA Journal of Numerical Analysis, (2011).
- [26] I. G. GRAHAM, R. SCHEICHL, AND E. ULLMANN, *Mixed finite element analysis of lognormal diffusion and multilevel monte carlo methods*, Stochastics and Partial Differential Equations Analysis and Computations, 4 (2016), pp. 41–75.
- [27] A. GREENBAUM, *Iterative methods for solving linear systems*, vol. 17, SIAM, 1997.
- [28] M. A. GREPL, Y. MADAY, N. C. NGUYEN, AND A. T. PATERA, *Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations*, ESAIM: Mathematical Modelling and Numerical Analysis, 41 (2007), pp. 575–605.
- [29] M. D. GUNZBURGER, C. G. WEBSTER, AND G. ZHANG, *Stochastic finite element methods for partial differential equations with random input data*, Acta Numerica, 23 (2014), pp. 521–650, doi:10.1017/S0962492914000075.
- [30] B. HAASDONK, K. URBAN, AND B. WIELAND, *Reduced basis methods for parameterized partial differential equations with stochastic influences using the karhunen-loeve expansion*, SIAM/ASA Journal on Uncertainty Quantification, 1 (2013), pp. 79–105.
- [31] J. S. HESTHAVEN, G. ROZZA, AND B. STAMM, *Certified reduced basis methods for parametrized partial differential equations*, SpringerBriefs in Mathematics, (2015).
- [32] Q. LIAO AND G. LIN, *Reduced basis ANOVA methods for partial differential equations with high-dimensional random inputs*, Journal of Computational Physics, 317 (2016), pp. 148–164.
- [33] G. J. LORD, C. E. POWELL, AND T. SHARDLOW, *An introduction to computational stochastic PDEs*, Cambridge University Press, 2014.
- [34] I. MARTINI, G. ROZZA, AND B. HAASDONK, *Reduced basis approximation and a-posteriori error estimation for*

- the coupled Stokes–Darcy system*, Advances in Computational Mathematics, 41 (2015), pp. 1131–1157.
- [35] F. NOBILE, R. TEMPONE, AND C. G. WEBSTER, *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM Journal on Numerical Analysis, 46 (2008), pp. 2309–2345, doi:10.1137/060663660, arXiv:<http://dx.doi.org/10.1137/060663660>.
 - [36] C. C. PAIGE AND M. A. SAUNDERS, *Solution of sparse indefinite systems of linear equations*, SIAM journal on numerical analysis, 12 (1975), pp. 617–629.
 - [37] C. E. POWELL, D. SILVESTER, AND V. SIMONCINI, *An efficient reduced basis solver for stochastic Galerkin matrix equations*, Preprint, (2015), http://eprints.ma.man.ac.uk/2350/01/Powell_Silvester_Simoncini.pdf.
 - [38] C. E. POWELL AND D. J. SILVESTER, *Optimal preconditioning for Raviart–Thomas mixed formulation of second-order elliptic problems*, SIAM journal on matrix analysis and applications, 25 (2003), pp. 718–738.
 - [39] A. QUARTERONI, A. MANZONI, AND F. NEGRI, *Reduced basis methods for partial differential equations: an introduction*, Springer International Publishing, 2016.
 - [40] B. RIVIÈ, M. F. WHEELER, AND C. BAUMANN, *Part II. discontinuous Galerkin method applied to a single phase flow in porous media*. <https://www.ices.utexas.edu/media/reports/1999/9910.pdf>, 1999.
 - [41] G. ROZZA, D. B. P. HUYNH, AND A. MANZONI, *Reduced basis approximation and a posteriori error estimation for Stokes flows in parametrized geometries: roles of the inf-sup stability constants*, Numerische Mathematik, 125 (2013), pp. 115–152.
 - [42] G. ROZZA AND K. VEROY, *On the stability of the reduced basis method for Stokes equations in parametrized domains*, Computer methods in applied mechanics and engineering, 196 (2007), pp. 1244–1260.
 - [43] T. RUSTEN AND R. WINTHER, *A preconditioned iterative method for saddlepoint problems*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 887–904.
 - [44] D. J. SILVESTER AND C. E. POWELL, *Potential Incompressible Flow and Iterative Solver Software (PIFISS) version 1.0*, February 2007. <http://www.maths.manchester.ac.uk/~djs/ifiss/pifiss.html>.
 - [45] R. C. SMITH, *Uncertainty quantification: theory, implementation, and applications*, vol. 12, SIAM, 2013.
 - [46] S. A. SMOLYAK, *Quadrature and interpolation formulas for tensor products of certain classes of functions*, Dokl. Akad. Nauk SSSR, 4 (1963), p. 123.
 - [47] A. M. STUART, *Inverse problems: a Bayesian perspective*, Acta Numer., 19 (2010), pp. 451–559, doi:10.1017/S0962492910000061.
 - [48] T. J. SULLIVAN, *Introduction to Uncertainty Quantification*, vol. 63, Springer, 2016.
 - [49] L. N. TREFETHEN AND D. BAU III, *Numerical linear algebra*, vol. 50, SIAM, 1997.
 - [50] D. XIU AND J. S. HESTHAVEN, *High-order collocation methods for differential equations with random inputs*, SIAM Journal on Scientific Computing, 27 (2005), pp. 1118–1139, doi:10.1137/040615201.