

## RANDOMIZED DISCRETE EMPIRICAL INTERPOLATION METHOD FOR NONLINEAR MODEL REDUCTION\*

ARVIND K. SAIBABA<sup>†</sup>

**Abstract.** The discrete empirical interpolation method (DEIM) is a popular technique for nonlinear model reduction, and it has two main ingredients: an interpolating basis that is computed from a collection of snapshots of the solution, and a set of indices which determine the nonlinear components to be simulated. The computation of these two ingredients dominates the overall cost of the DEIM algorithm. To specifically address these two issues, we present randomized versions of the DEIM algorithm. There are three main contributions of this paper. First, we use randomized range finding algorithms to efficiently find an approximate DEIM basis. Second, we develop randomized subset selection tools, based on leverage scores, to efficiently select the nonlinear components. Third, we develop several theoretical results that quantify the accuracy of the randomization on the DEIM approximation. We also present numerical experiments that demonstrate the benefits of the proposed algorithms.

**Key words.** model reduction, randomized algorithms, discrete empirical interpolation method, subset selection, subspace iteration

**AMS subject classifications.** 15A23, 65F30, 65N22, 93B40, 93C15, 93C20

**DOI.** 10.1137/19M1243270

**1. Introduction.** Detailed mathematical models of weather prediction and neuroscience routinely generate large-scale problems having over a billion unknowns. The goal of model reduction is to replace computationally expensive full-scale models by reduced order models (ROMs) that are cheaper to evaluate and that preserve the important underlying physics in the full-scale models. Development of effective ROMs will enable efficient and accurate simulation of a wide range of detailed complex physical phenomena as well as benefit a host of applications in inverse problems, data assimilation, design, control, optimization, and uncertainty quantification.

In a typical model reduction technique, there are two distinct phases: an *offline phase* in which the full model is simulated for a range of parameters or specifications and the outputs of this simulations are used to construct the ROM, and an *online phase*, in which the ROM is simulated for the desired parameter, or specification. A successful ROM has two features that are hard to achieve simultaneously: the ROM should be accurate over the desired range of parameters and specifications, and the online phase should be inexpensive—the dominant computational cost should be in the offline phase. A popular method for model reduction is proper orthogonal decomposition (POD), which has been successfully used in a wide range of partial differential equation (PDE)-based applications, and is reviewed in subsection 2.1. While the POD approach has broad applicability, the computational efficiency of POD demands that the underlying PDE has to be linear, or the parameter dependence has to be of a specific type, e.g., affine parameter dependence. To address this deficiency, many methods have been proposed in the literature such as the gappy POD interpolation method,

\*Submitted to the journal's Methods and Algorithms for Scientific Computing section February 7, 2019; accepted for publication (in revised form) February 26, 2020; published electronically May 19, 2020.

<https://doi.org/10.1137/19M1243270>

**Funding:** This work was partially supported by the National Science Foundation through grants DMS 1720398 and DMS 1821149.

<sup>†</sup>Department of Mathematics, North Carolina State University, Raleigh, NC 27695 (asaibab@ncsu.edu).

the empirical interpolation method (EIM), and its discrete variant, the discrete empirical interpolation method (DEIM). Reviews of various model reduction techniques are provided in the survey paper [3] and in recent books and monographs [17, 25].

The DEIM interpolation framework computes an approximation of a nonlinear function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$  by the means of a basis  $W \in \mathbb{R}^{n \times r}$  used to interpolate the function, and a set of indices, defining a point selection operator  $S \in \mathbb{R}^{n \times r}$ , at which the nonlinear function is evaluated. In subsection 2.2, we explain how  $W$  and  $S$  can be used to approximate the function  $f$ ; here, we describe the major bottlenecks in computing the DEIM approximation.

- The basis  $W$  is constructed as follows: several representative samples—also called snapshots—of the function  $f(\cdot)$  are collected and arranged as columns of a matrix, known as the snapshot matrix. The left singular vectors of this snapshot matrix form the desired DEIM basis—henceforth, we call this the standard basis. The dimension of the subspace spanned by the DEIM basis, denoted by  $r$ , depends on the number of dominant singular values of the snapshot matrix. Computing a truncated singular value decomposition (SVD) costs  $\mathcal{O}(rn_s n)$ , where  $n_s$  is the number of snapshots; our approach replaces the SVD by a randomized SVD.
- Finding a set of good indices is a combinatorially hard problem known as *subset selection* (in the DEIM literature, this is known as point selection, which we also adopt in this manuscript). Various deterministic subset selection techniques have been proposed in the literature: these are based on pivoted LU factorization [7, 28], pivoted QR factorization [9], and strong rank-revealing QR factorization [10]. The computational cost is roughly  $\mathcal{O}(nr^2)$ ; we use randomized subset selection techniques, which lowers this cost and can exploit parallelism.

Both of these operations are computationally expensive when  $n$  is large. Our paper specifically addresses these computational challenges using randomized algorithms, thereby enabling efficient large-scale implementation of DEIM.

To motivate the development of a randomized algorithm for DEIM, we briefly review randomized algorithms in other applications. Recently, randomized algorithms have been developed for accurate low-rank approximations to matrices arising from large datasets. The basic idea of these methods is to use random sampling to identify a subspace which approximately captures the range of the matrix [16]. The matrix is then projected onto this subspace, and then deterministic linear algebraic techniques can be used to manipulate the projected matrix to obtain the desired low-rank approximation. In addition to low-rank approximations, randomized methods (based on leverage score sampling and other subset selection techniques) have been developed for other linear algebraic problems such as least squares problems, regression, and computation of interpretable decompositions such as CX/CUR decompositions; see recent survey articles [20, 8]. Randomized methods have several advantages over their corresponding classical counterparts: typically, they are computationally efficient, numerically robust, and easy to implement, and they scale well in a distributed computing setting and have well-developed error analysis.

*Contributions and contents.* We present randomized algorithms for DEIM that enable nonlinear dimensionality reduction for several large-scale applications.

We present randomized approaches (section 3) for efficient construction of a DEIM basis  $\widehat{W}$ . The algorithms come in two flavors depending on whether the target rank  $r$  is known or not. When the target rank is known, we present a basic version and

a more accurate version based on subspace iteration. When the target rank  $r$  is unknown, we provide an adaptive algorithm for computing  $\widehat{W}$ . The computational and storage advantages of the randomized algorithms for constructing the DEIM basis are highlighted.

We present a detailed analysis of the error (section 4) that occurs when a randomized basis is used instead of the standard DEIM basis. A crucial component of our analysis involves the largest canonical angle between the subspaces spanned by the standard basis  $W$  and an approximate basis  $\widehat{W}$ . This analysis is applicable to any approximate basis  $\widehat{W}$ , and therefore this has broad appeal beyond the context of randomized algorithms. We also present specific results that explicitly account for the randomness on the accuracy of the DEIM approximation.

We propose two randomized point selection methods for DEIM approximation (section 5): leverage score sampling and hybrid algorithms. For each method, we present theoretical bounds on the number of points required for the desired accuracy. The hybrid point selection technique combines the computational advantages of the randomized methods with the accuracy of deterministic methods. These sampling methods have been proposed in the context of matrix CUR decompositions; our analysis for the DEIM approximation is new.

Numerical experiments (section 6) demonstrate the computational benefits, the accuracy of the randomized approaches, and insight into the choice of parameters for various algorithms presented in this paper.

*Related work.* The idea of using randomization to accelerate computations in model reduction appears to be relatively new, and here we briefly review the literature. Randomized matrix methods similar to the ones used in this paper have been used to approximate POD [33, 34, 2] and dynamic mode decomposition [12, 11, 4]. Recent work in [1] uses randomization for reducing the cost associated with multiple right-hand sides in nonlinear model reduction. The resulting ROM dramatically reduces the cost of solving a PDE-based inverse problem. However, none of these references directly tackle the DEIM approximation, which is the central focus of this paper. Randomized sampling approaches for choosing the DEIM indices have been proposed in [9], but no analysis of the randomization was presented. Another noteworthy paper [23] uses randomized oversampling to address stability issues with DEIM.

**2. Preliminaries.** We briefly review the POD and DEIM approaches for model order reduction.

**2.1. Proper orthogonal decomposition.** We explain the POD approach in the context of a nonlinear dynamical system that takes the form

$$(2.1) \quad \frac{dx(t)}{dt} = Mx(t) + f(x(t)), \quad x(0) = x_0,$$

where  $M \in \mathbb{R}^{n \times n}$ . When  $n$  is large, the simulation of the dynamical system can be computationally expensive, and we turn to ROMs to lower the computational cost. In the POD approach, the dynamical system is first simulated numerically, and the “snapshots” of the system at multiple times  $0 \leq t_1 < \dots < t_{n_t} \leq T$ , as  $x_j = x(t_j)$  for  $j = 1, \dots, n_t$ , are collected into the POD snapshot matrix

$$F = [x_1 \quad \dots \quad x_{n_t}] \in \mathbb{R}^{n \times n_t}.$$

We then compute its thin SVD  $F = Y\Sigma Z^\top$ . The POD approach chooses the  $k \leq \rho = \text{rank}(F)$  singular vectors corresponding to the largest singular values, collected in a

matrix  $V_k = Y(:, 1 : k)$  (using MATLAB notation). The basis, thus obtained, also solves the following optimization problem:

$$(2.2) \quad \min_{V \in \mathbb{R}^{n \times k}; V^\top V = I_k} \|(I_n - VV^\top)F\|_F^2 = \sum_{j=k+1}^{n_t} \sigma_j^2(F),$$

where  $\sigma_j(F)$  are the singular values of  $F$ . The choice of snapshots is an important issue in determining an effective POD basis and is discussed in [25, 3, 17]. Assuming that we have an effective basis  $V_k$ , the solution  $x(t)$  can be approximated to be constrained in the span of the basis of the columns of  $V_k$ , i.e.,  $x(t) \approx V_k \hat{x}(t)$ . Next, the reduced system is obtained by a Galerkin projection onto  $\mathcal{R}(V_k)$ ; the dynamics of  $\hat{x}(t)$  is given by

$$\frac{d\hat{x}(t)}{dt} = V_k^\top M V_k \hat{x}(t) + V_k^\top f(V_k \hat{x}(t)), \quad \hat{x}(t) = V_k^\top x_0.$$

This approach, known as the *POD-Galerkin* method, is an effective way of reducing the dimensionality of the system of equations represented in (2.1). This approach is more generally applicable to other applications such as parameterized PDEs.

We briefly discuss the computational cost of the POD-Galerkin approach. Note that the matrix  $\hat{M} \equiv V_k^\top M V_k$  can be precomputed. If the nonlinear term  $f = 0$ , then the cost of simulating the reduced system for  $\hat{x}(t)$  is independent of  $n$ , the dimension of the full order system. However, the evaluation of the nonlinear term  $V_k^\top f(V_k \hat{x}(t))$  has computational complexity that depends on  $n$ . As a result, evaluation of this term may still be as expensive as solving the original system.

**2.2. DEIM approximation.** The DEIM approximation was proposed to address the deficiency of the POD-Galerkin approach for nonlinear dynamical systems. Given a collection of snapshots of the full order dynamical system

$$A = [f(x_1) \quad \dots \quad f(x_{n_s})],$$

a projection basis is computed by retaining the  $r$  left-singular vectors corresponding to the top- $r$  singular values. We denote this standard DEIM basis by  $W \in \mathbb{R}^{n \times r}$ . The DEIM approach then selects  $s$  distinct rows from the matrix  $W$ ; in particular, denoting the row indices  $\{t_1, \dots, t_s\}$ , we define the *selection operator*

$$S = [e_{t_1} \quad \dots \quad e_{t_s}] \in \mathbb{R}^{n \times s},$$

where  $e_{t_i}$  is the  $t_i$ th column of the  $n \times n$  identity matrix.

We define the DEIM projector  $\mathbb{D} \equiv W(S^\top W)^\dagger S^\top$  that satisfies  $\mathbb{D}^2 = \mathbb{D}$  (i.e.,  $\mathbb{D}$  is idempotent) and if  $\mathbb{D} \neq 0$  and  $\mathbb{D} \neq I_n$ , then

$$(2.3) \quad \|\mathbb{D}\|_2 = \|I_n - \mathbb{D}\|_2.$$

The equality follows from the property of oblique projectors [30]. Since  $S$  and  $W$  have orthonormal columns and  $\|\cdot\|_2$  is unitarily invariant, we also have the equality  $\|\mathbb{D}\|_2 = \|(S^\top W)^\dagger\|_2$ . Given the DEIM projector, the DEIM approximation of  $f$  can be expressed as

$$(2.4) \quad \hat{f}_{\text{DEIM}} \equiv \mathbb{D}f.$$

We recapitulate a few properties of the DEIM projector that will be useful in subsequent analysis. Let  $P_W \equiv WW^\top$  and  $P_S \equiv SS^\top$  be two orthogonal projectors corresponding to the bases  $W$  and  $S$ , respectively. Suppose that  $\text{rank}(\mathbb{D}) =$

$\text{rank}(S^\top W) = r = s$ . The following relations hold:

$$(2.5) \quad \mathbb{D} = \mathbb{D}P_S = P_W\mathbb{D},$$

$$(2.6) \quad \mathbb{D}P_W = P_W, \quad I_n - \mathbb{D} = (I_n - \mathbb{D})(I_n - P_W),$$

$$(2.7) \quad P_S\mathbb{D} = P_S, \quad I_n - \mathbb{D} = (I_n - P_S)(I_n - \mathbb{D}).$$

*Remark 1.* In the original DEIM approach [7], the number of selected indices  $s$  equals the dimension of the DEIM basis  $r$ . The case  $s \neq r$  has its origins in gappy POD (see [25, Remark 10.4] and references therein). The implications for the interpolation and projection properties of the DEIM operator have been discussed in detail in [10, section 3]. Of importance in our analysis will be the case  $s \geq r$  and  $\text{rank}(\mathbb{D}) = r$ . In this case, (2.5) and (2.6) hold, but (2.7) no longer holds; see [10, section 3].

We also have the following error in the DEIM approximation.

**LEMMA 2.1.** *Let the DEIM approximation be defined in (2.4), and let  $\text{rank}(\mathbb{D}) = \text{rank}(S^\top W) = r$  with  $s \geq r$ :*

$$\|f - \hat{f}_{\text{DEIM}}\|_2 \leq \|\mathbb{D}\|_2 \|(I_n - WW^\top)f\|_2.$$

*Proof.* See [7, Lemma 3.2]. □

*Remark 2.* The error in the function approximation  $\|(I_n - WW^\top)f\|_2$  depends on the quality of the basis  $W$ ; a priori bounds for this term are difficult to derive. In this work, we make the rather strong assumption that  $\|(I_n - WW^\top)f\|_2 \approx \sigma_{r+1}(A)$  (see [25, equation (10.20)] and the discussion surrounding it for a justification). That is, the error depends on the largest discarded singular value of the snapshot matrix.

The quantity  $\|\mathbb{D}\|_2$  can be interpreted as the condition number of the DEIM approximation. In what follows, we also refer to it as the DEIM error constant, and it depends on the particular point selection technique that is used. A discussion of this is provided in section 5. When  $S$  and  $W$  have orthonormal columns and  $\text{rank}(S^\top W) = r$ , then  $\|\mathbb{D}\|_2 = \|(S^\top W)^\dagger\|_2$ .

It remains to be shown how to use the DEIM approximation along with the POD. Suppose that the DEIM basis  $W$  and the selection operator  $S$  have been determined. In the dimension-reduced version of the nonlinear dynamical system, replace  $f$  by  $\hat{f}_{\text{DEIM}}$ , i.e.,

$$V_k^\top f(V_k \hat{x}(t)) \approx (V_k^\top W)(S^\top W)^\dagger S^\top f(V_k \hat{x}(t)).$$

In the offline stage, the matrices  $(V_k^\top W)$  and a factorization of  $S^\top W$  can be precomputed. This involves a computational cost of  $\mathcal{O}(nkr + r^3)$ . In the online stage, instead of evaluating the nonlinear function, only an  $s \geq r$  selected number of components of the function will be evaluated at the indices that determine the columns of the selection operator  $S$ .

The computation of a standard DEIM basis  $W$  and the interpolating indices that define  $S$  constitute the major bottlenecks in the large-scale implementation of DEIM. In section 3, we develop randomized algorithms to accelerate the computation of the basis  $W$ , and in section 5, we develop efficient randomized point selection algorithms.

**3. Randomized algorithm for DEIM basis.** Let  $A \in \mathbb{R}^{n \times n_s}$  with  $n \geq n_s$  and  $r \leq \text{rank}(A)$  be the matrix of snapshots. We can write the SVD of  $A$  as

$$A = [W_1 \quad W_2] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} \begin{bmatrix} Z_1^\top \\ Z_2^\top \end{bmatrix}.$$

Here  $W_1 \in \mathbb{R}^{n \times r}$  and  $W_2 \in \mathbb{R}^{n \times (n-r)}$  contain the left-singular vectors, whereas  $Z_1 \in \mathbb{R}^{n_s \times r}$  and  $Z_2 \in \mathbb{R}^{n_s \times (n_s-r)}$  contain the right-singular vectors. The matrices  $\Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$  and  $\Sigma_2 = \text{diag}(\sigma_{r+1}, \dots, \sigma_{n_s}) \in \mathbb{R}^{(n-r) \times (n_s-r)}$  contain the singular values of  $A$  in decreasing order. Furthermore,  $\|\Sigma_1^{-1}\|_2 = 1/\sigma_r$  and  $\|\Sigma_2\|_2 = \sigma_{r+1}$ . Write  $A = A_r + A_{r,\perp}$ , where, by the Eckart–Young theorem,  $A_r \equiv W_1 \Sigma_1 Z_1^\top$  is the best rank- $r$  approximation to  $A$  [29, section 4.5].

The standard DEIM approximation uses  $W = W_1$ . The randomized DEIM approximation replaces the exact SVD with a randomized SVD. Here and henceforth, we refer to the basis generated using a randomized algorithm as the R-DEIM basis, and the resulting approximation as the R-DEIM approximation.

**3.1. Randomized SVD.** We briefly review an idealized version of the randomized range finding algorithm. Suppose the target rank is known and denoted by  $r$ . Draw a standard Gaussian random matrix  $\Omega \in \mathbb{R}^{n \times r}$  (i.e., a matrix with entries independent and identically distributed normal variables having mean 0 and variance 1). Form the matrix  $Y = A\Omega$  (which is often called the “sketch matrix” or “sketch”), and compute an orthonormal basis for  $\mathcal{R}(Y)$  using the thin QR factorization  $Y = QR$ . In practice, instead of  $r$  columns,  $\ell = r + p$  columns are drawn; here  $p > 0$  is a small oversampling parameter. A low-rank approximation to  $A$  can be obtained as  $A \approx Q(Q^\top A)$ . Compute the top  $r$  left-singular vectors of  $B \equiv Q^\top A$  by  $W_r$ , and obtain the approximate basis as  $\widehat{W} = QW_r$ . This is summarized in Algorithm 1.

**Input:** Snapshot matrix  $A \in \mathbb{R}^{n \times n_s}$ , target rank  $r$ , oversampling parameter  $p \geq 1$  such that  $r + p \leq \min\{n, n_s\}$ .

**Output:** Basis  $\widehat{W}$  with orthonormal columns.

- 1 Draw a standard Gaussian matrix  $\Omega \in \mathbb{R}^{n \times (r+p)}$ .
- 2 Form  $Y = A\Omega$  and compute thin QR factorization  $Y = QR$ .
- 3 Form  $B = Q^\top A$ . Let  $W_r$  be the left-singular vectors corresponding to top  $r$  singular values of  $B$ .
- 4 Form  $\widehat{W} \leftarrow QW_r$ .

**Algorithm 1:** Basic randomized range finding algorithm.

The error in the low-rank approximation can be obtained by [16, Theorem 10.6] (when  $p \geq 2$  and  $n \geq n_s$ ):

$$\mathbb{E}_\Omega \|(I_n - QQ^\top)A\|_2 \leq \left(1 + \sqrt{\frac{r}{p-1}}\right) \sigma_{r+1} + \frac{e\sqrt{r+p}}{p} \left(\sum_{j=r+1}^{n_s} \sigma_j^2\right)^{1/2}.$$

However, the error in the low-rank representation does not fully explain the error in the R-DEIM approximation. As shown in section 4, we will need to bound the canonical angles between the subspaces spanned by the singular vectors.

In practice, the target rank  $r$  may not be known in advance. In the standard DEIM approach, the rank  $r$  is chosen based on the decay of the singular values of  $A$ . However, the exact singular values are not known to us, and therefore a new approach is needed. We use the approach in [35] that adaptively determines the target rank  $r$ .

**3.2. Adaptive randomized range finder.** In the standard DEIM approach, the truncation index is taken to be the smallest index  $r$  which satisfies

$$\frac{\|A - A_r\|_F^2}{\|A\|_F^2} = \frac{\sum_{j=r+1}^{n_s} \sigma_j^2(A)}{\sum_{j=1}^{n_s} \sigma_j^2(A)} \leq \epsilon_{\text{tol}}^2.$$

The matrix  $A_r$  was defined at the start of this section. This condition ensures that the relative error of the low-rank representation, as measured in the Frobenius norm, is smaller than a positive user-defined tolerance  $\epsilon_{\text{tol}}$ . Based on this criterion, we require the R-DEIM basis  $\widehat{W}$  to satisfy the condition

$$(3.1) \quad \|A - \widehat{W}\widehat{W}^\top A\|_F^2 \leq \epsilon_{\text{tol}}^2 \|A\|_F^2.$$

Several adaptive randomized range finding algorithms were presented in the literature [16, 21, 35, 13]. In this paper, we adopt the approach presented in [35]. An outline of this algorithm is given in Algorithm 2; however, we refer the reader to [35, Algorithm 3] for details regarding the implementation. Another variation [35, Algorithm 4] combines an adaptive strategy with the subspace iteration for enhanced accuracy.

**Input:** Snapshot matrix  $A \in \mathbb{R}^{n \times n_s}$ , block size  $b \geq 1$ , tolerance  $\epsilon_{\text{tol}} > 0$ , factor  $\alpha$ .  
**Output:** Basis  $\widehat{W}$  with orthonormal columns.

- 1 Compute  $\alpha = \|A\|_F^2$ .
- 2  $W \leftarrow []$ .
- 3 Parameter  $\beta = 0$ . // Norm of  $B = Q^\top A$ .
- 4 **while**  $\beta \leq \alpha(1 - \epsilon_{\text{tol}}^2)$  **do**
- 5     Draw a standard Gaussian matrix  $\Omega \in \mathbb{R}^{n_s \times b}$ .
- 6     Compute  $Z = (A - WB)\Omega$ , and the thin QR factorization  $QR = Z$ .
- 7     Orthogonalize:  $Q \leftarrow (I_n - WW^\top)Q$ .
- 8     Compute  $B' = Q^\top A - Q^\top WB$ .
- 9     Extend  $W = [W \quad Q]$  and  $B = \begin{bmatrix} B \\ B' \end{bmatrix}$ .
- 10    Update norm:  $\beta \leftarrow \beta + \|Q^\top A\|_F^2$ .
- 11 **end**
- 12  $\widehat{W} \leftarrow W$ .

**Algorithm 2:** Adaptive randomized range finding algorithm [35].

**3.3. Computational advantages.** The standard DEIM approach computes the compact SVD of the snapshot matrix; this is expensive and costs  $\mathcal{O}(nn_s^2)$  flops, assuming that  $n_s \leq n$ . On the other hand, the randomized approach only requires  $\mathcal{O}(rnn_s)$  flops and is computationally advantageous when  $r \ll n_s$ .

Besides this, there are several benefits of this approach that are worth pointing out. First, the rank  $r$  need not be known a priori and is determined adaptively (Algorithm 2). Second, the sketch  $Y = A\Omega$  can be computed by taking advantage of the sequential nature of the snapshot generation. Third, the DEIM basis can be efficiently updated. See below for more details regarding the last two points.

**3.3.1. Reduced storage costs.** In model reduction techniques involving dynamical systems, the snapshots used to compute the DEIM basis are generated sequentially by a time-stepping method. When a fine-scale spatial discretization is used, the number of degrees of freedom  $n$  can be large and the cost of storing many snapshots can be overwhelmingly, even prohibitively, large. We can take advantage of the sequential nature of the snapshot generation to reduce storage costs.

Suppose that the target rank  $r$  is known in advance (for simplicity, we do not include oversampling). The only step that involves manipulating the snapshots is the computation of the sketch  $Y = A\Omega$ . Note that we can alternatively express this sum of rank-1 outer products as

$$Y = \sum_{j=1}^{n_s} A(:,j)\Omega(j,:).$$

Here  $A(:,j)$  are the columns of the snapshot matrix, and  $\Omega(j,:)$  are the rows of  $\Omega$ . This formula means that once each snapshot  $A(:,j)$  is generated, the sketch can be updated appropriately using  $A(:,j)\Omega(j,:)$ , and then the snapshot can be discarded. If the target rank  $r$  is much smaller than the number of snapshots  $n_s$ , the storage cost is lowered to  $\mathcal{O}(nr)$  instead of  $\mathcal{O}(nn_s)$ , and these savings may be substantial. An alternative option is maintaining two sets of sketches as advocated in [31].

**3.3.2. Adapting the basis.** Adapting the basis becomes necessary in certain applications [24]; for example, in the offline stage, the snapshot matrix  $A$  may be constructed with the objective of making the DEIM approximation accurate over the entire parameter range, but computational considerations may constrain the approximation to be accurate only over a certain region in parameter space. The DEIM approximation may be used as a surrogate for the original function in an optimization setting. As the optimization routine makes progress, the DEIM approximation may not be accurate if the optimization path deviates from the region of accurate DEIM approximation; in this case, a good strategy may be to update the DEIM basis based on the optimization trajectory.

Randomized algorithms allow the user the flexibility to readily update the DEIM basis by simply updating the sketch  $Y$ , instead of recomputing the SVD. Suppose the  $j$ th column of the snapshot matrix needs to be replaced by  $a_j$ ; we make the assumption that the target rank  $r$  remains the same. The corresponding sketch can be replaced as  $Y \leftarrow Y + (a_j - A(:,j))\Omega$ ; a thin QR can be performed to obtain the basis, and the entire snapshot matrix is not necessary for adapting the basis. It is easily seen how to simultaneously replace a block of columns.

**3.4. Improved accuracy via subspace iteration.** For some applications, the decay in the singular values may not be rapid enough to ensure that the resulting subspace computed  $\widehat{W}$  is accurate. The basic idea is to replace the sketch  $Y = A\Omega$  in Algorithm 1 with the sketch  $Y = (AA^\top)^q A\Omega$ . Essentially this means running  $q$  steps of subspace iteration where  $q$  is a nonnegative integer. However, it is well known that a direct computation of the sketch  $Y = (AA^\top)^q A\Omega$  is numerically unstable and is significantly affected by round-off error. To address this issue, numerically stable methods alternate the matrix-vector products involving  $A$  with a QR factorization. Algorithm 3 gives an idealized version of the algorithm that will be suitable for analysis in subsection 4.3. For more details on a numerically stable implementation, see [16, 26].



**Input:** Snapshot matrix  $A \in \mathbb{R}^{n \times n_s}$ , target rank  $r$ , oversampling parameter  $p \geq 1$  such that  $r + p \leq \min\{n, n_s\}$ , number of iterations  $q \geq 0$ .

**Output:** Basis  $\widehat{W}$  with orthonormal columns.

- 1 Draw a standard Gaussian matrix  $\Omega \in \mathbb{R}^{n_s \times (r+p)}$ .
- 2 Form  $Y = (AA^\top)^q A \Omega$  and compute thin QR factorization  $Y = QR$ .
- 3 Form  $B = Q^\top A$ . Let  $W$  be the left-singular vectors of  $B$  and set  $W_r = W(:, 1:r)$ .
- 4 Form  $\widehat{W} \leftarrow QW_r$ .

**Algorithm 3:** Idealized randomized subspace iteration for range finding. Call as  $[\widehat{W}] = \text{RandSubspace}(A, r, p, q)$ .

**4. Error analysis.** In subsection 4.2, we derive bounds for the accuracy of the DEIM approximation, when a perturbed DEIM basis  $\widehat{W}$  is used instead of the standard DEIM basis  $W$ . The bounds are applicable whether  $\widehat{W}$  is obtained using a randomized algorithm or using any other approximation algorithm. For example, due to the inexactness in the function evaluations, the snapshot matrix  $A$  may be perturbed by  $\Delta A$ . In this case, the left-singular vectors  $\widehat{W}$  of  $A + \Delta A$  can be used as the perturbed DEIM basis. The results in this subsection are applicable to this setting as well. In subsection 4.3, we derive bounds for the angles between the subspaces spanned by columns of  $W$  and  $\widehat{W}$  when  $\widehat{W}$  is obtained using the randomized range finding algorithms. We then use these bounds to fully quantify the error in the R-DEIM approximation.

**4.1. Notation and canonical angles.** Let  $W \in \mathbb{R}^{n \times r}$  be the standard DEIM basis, obtained from the first  $r$  left-singular vectors of  $A$ , and let  $S \in \mathbb{R}^{n \times s}$  be the selection operator. For generality, we will assume that the selection operator contains columns from the identity matrix but may be scaled (see section 5 for examples). The following orthogonal projectors will be of use in what follows:

$$P_W \equiv WW^\top, \quad P_S \equiv SS^\dagger.$$

We note that (2.5) and (2.6) still hold if  $s \geq r$  and  $\text{rank}(S^\top W) = r$ , and all three relations hold if  $s = r$  and  $\text{rank}(S^\top W) = r$ .

To distinguish from the standard DEIM basis, denote  $\widehat{W} \in \mathbb{R}^{n \times r}$  as the “perturbed” basis (obtained, for example, from Algorithm 1) with orthonormal columns and the corresponding selection operator  $\widehat{S} \in \mathbb{R}^{n \times s}$ ; assume that  $\text{rank}(\widehat{S}^\top \widehat{W}) = r$ . Define the corresponding projectors

$$P_{\widehat{W}} \equiv \widehat{W}\widehat{W}^\top, \quad P_{\widehat{S}} \equiv \widehat{S}\widehat{S}^\dagger, \quad \mathbb{D} \equiv \widehat{W}(\widehat{S}^\top \widehat{W})^\dagger \widehat{S}^\top.$$

We allow for the possibility that the selection operator  $\widehat{S}$  and  $S$  may be different; these may be computed based on the perturbed DEIM basis  $\widehat{W}$  and the standard DEIM basis  $W$ , respectively. Throughout this section, we assume that the DEIM projectors  $\mathbb{D}, \widehat{\mathbb{D}}$  do not equal either the zero matrix or the identity matrix.

The analysis of the error in the DEIM approximation requires computing the overlap between two subspaces of  $\mathbb{R}^n$  which, in turn, can be described in terms of canonical angles. We now briefly review some definitions and properties of canonical angles; see [29, Chapter I.5]. Denote the range spaces of  $W$  and  $\widehat{W}$  by  $\mathcal{W}$  and  $\widehat{\mathcal{W}}$ ,

respectively; both these subspaces have dimension  $r$ . The principal or canonical angles between  $\mathcal{W}$  and  $\widehat{\mathcal{W}}$  are  $\theta_1, \dots, \theta_r = \theta_{\max}$  and satisfy

$$0 \leq \theta_1 \leq \dots \leq \theta_{\max} \leq \pi/2.$$

The canonical angles can be computed using the SVD. Denote the singular values  $\sigma_1 \geq \dots \geq \sigma_r$  of  $W^\top \widehat{W}$ ; then the canonical angles are  $\theta_i = \arccos(\sigma_i)$ , and furthermore,

$$(4.1) \quad \sin \theta_{\max} = \|P_W - P_{\widehat{W}}\|_2 = \|(I_n - P_W)P_{\widehat{W}}\|_2 = \|(I_n - P_{\widehat{W}})P_W\|_2.$$

Similarly, let  $\psi_{\max}$  denote the largest canonical angle between the pair of subspaces  $\mathcal{R}(S)$  and  $\mathcal{R}(\widehat{S})$ . Equalities analogous to (4.1) also hold for  $\psi_{\max}$ .

**4.2. DEIM approximation.** We present two theorems that quantify the error in the perturbed DEIM approximation. We use the notation that was established in subsection 4.1. We remind the reader that both the basis  $\widehat{W}$  and the selection operator  $\widehat{S}$  may be different than the standard DEIM basis  $W$  and selection operator  $S$ , respectively.

**THEOREM 4.1.** *Let  $\widehat{\mathbb{D}}f$  be the perturbed DEIM approximation to  $\mathbb{D}f$ , and assume that  $s \geq r$  and  $\text{rank}(\widehat{\mathbb{D}}) = r$ . The approximation error satisfies*

$$\|f - \widehat{\mathbb{D}}f\|_2 \leq \|\widehat{\mathbb{D}}\|_2 (\|(I_n - P_W)f\|_2 + \sin \theta_{\max} \|P_W f\|_2).$$

*Proof.* From the expression  $f - \widehat{\mathbb{D}}f = (I_n - \widehat{\mathbb{D}})(P_W + I_n - P_W)f$  and by the triangle inequality

$$\|f - \widehat{\mathbb{D}}f\|_2 \leq \|(I_n - \widehat{\mathbb{D}})P_W f\|_2 + \|(I_n - \widehat{\mathbb{D}})(I_n - P_W)f\|_2.$$

Recall that since  $s \geq r$  and  $\text{rank}(\widehat{\mathbb{D}}) = \text{rank}(\widehat{S}^\top \widehat{W}) = r$ , therefore  $I_n - \widehat{\mathbb{D}} = (I_n - \widehat{\mathbb{D}})(I_n - P_{\widehat{W}})$ . By assumption  $\widehat{\mathbb{D}} \neq 0$  and  $\widehat{\mathbb{D}} \neq I_n$ , and therefore, by (2.3),  $\|I_n - \widehat{\mathbb{D}}\|_2 = \|\widehat{\mathbb{D}}\|_2$ . Using these identities, we get

$$\|f - \widehat{\mathbb{D}}f\|_2 \leq \|\widehat{\mathbb{D}}\|_2 (\|(I_n - P_{\widehat{W}})P_W f\|_2 + \|(I_n - P_W)f\|_2).$$

Use submultiplicativity

$$\|(I_n - P_{\widehat{W}})P_W f\|_2 \leq \|(I_n - P_{\widehat{W}})P_W\|_2 \|P_W f\|_2 = \sin \theta_{\max} \|P_W f\|_2,$$

and plug this into the previous equation to obtain the advertised result.  $\square$

The first term is similar to the error in the DEIM approximation Lemma 2.1. The second term is the additional error introduced by using the perturbed DEIM basis  $\widehat{W}$  and is quantified by the sine of the largest canonical angle between the subspaces  $\mathcal{W}$  and  $\widehat{\mathcal{W}}$ —if these subspaces are identical, and if  $S$  equals  $\widehat{S}$ , then the error reduces to the standard DEIM approximation.

It is worth comparing this error bound with that of the standard DEIM approximation Lemma 2.1. A little bit of algebra reveals that (if  $P_W f \neq 0$ )

$$\|f - \widehat{\mathbb{D}}f\|_2 \leq \kappa \|(I_n - P_W)f\|_2,$$

where  $\kappa$  is an amplification factor, and

$$\kappa = \left(1 + \frac{\sin \theta_{\max}}{\|(I_n - P_W)f\|_2 / \|P_W f\|_2}\right) \|\widehat{\mathbb{D}}\|_2.$$

An important distinction between the original DEIM result and Theorem 4.1 is that the condition number now appears to explicitly depend on the function  $f$ .

A different proof technique leads to a qualitatively different bound that includes both angles  $\theta_{\max}$  and  $\psi_{\max}$ . It requires the additional assumption that the number of selected indices  $s$  equals the dimension of the DEIM basis  $r$ .

**THEOREM 4.2.** *Let  $\widehat{\mathbb{D}}f$  be the perturbed DEIM approximation to  $\mathbb{D}f$ . Assume that  $s = r$  and  $\text{rank}(\mathbb{D}) = \text{rank}(\widehat{\mathbb{D}}) = r$ . Then, the approximation error is*

$$(4.2) \quad \|f - \widehat{\mathbb{D}}f\|_2 \leq \|\mathbb{D}\|_2 \|(I_n - P_W)f\|_2 + \|\mathbb{D}\|_2 \|\widehat{\mathbb{D}}\|_2 (\sin \psi_{\max} \|(I_n - P_W)f\|_2 + \sin \theta_{\max} \|P_S f\|_2).$$

*Proof.* The proof uses the decomposition

$$I_n - \widehat{\mathbb{D}} = I_n - \mathbb{D} + (I_n - \widehat{\mathbb{D}})\mathbb{D} - \widehat{\mathbb{D}}(I_n - \mathbb{D}).$$

Applying the triangle inequality results in

$$\|(I_n - \widehat{\mathbb{D}})f\|_2 \leq \|(I_n - \mathbb{D})f\|_2 + \|(I_n - \widehat{\mathbb{D}})\mathbb{D}f\|_2 + \|\widehat{\mathbb{D}}(I_n - \mathbb{D})f\|_2.$$

The first term is the standard DEIM error and is bounded by  $\|\mathbb{D}\|_2 \|(I_n - P_W)f\|_2$ . For the subsequent terms, using the analogues of (2.5)–(2.7), we have the equalities

$$\begin{aligned} (I_n - \widehat{\mathbb{D}})\mathbb{D} &= (I_n - \widehat{\mathbb{D}})(I_n - P_{\widehat{W}})P_W \mathbb{D} P_S, \\ \widehat{\mathbb{D}}(I_n - \mathbb{D}) &= \widehat{\mathbb{D}}P_{\widehat{S}}(I_n - P_S)(I_n - \mathbb{D})(I_n - P_W). \end{aligned}$$

Therefore, with repeated application of the submultiplicativity inequality

$$\|(I_n - \widehat{\mathbb{D}})\mathbb{D}f\|_2 = \|(I_n - \widehat{\mathbb{D}})(I_n - P_{\widehat{W}})P_W \mathbb{D} P_S f\|_2 \leq \sin \theta_{\max} \|I_n - \widehat{\mathbb{D}}\|_2 \|\mathbb{D}\|_2 \|P_S f\|_2.$$

The identity (2.3), along with the assumption  $\widehat{\mathbb{D}} \neq 0$  and  $\widehat{\mathbb{D}} \neq I_n$ , completes the second term. The last term is obtained in a similar manner.  $\square$

The interpretation of this theorem is similar to that of Theorem 4.1. If  $\mathcal{R}(S) = \mathcal{R}(\widehat{S})$  and  $\mathcal{R}(W) = \mathcal{R}(\widehat{W})$ , then the two trailing terms drop out, and we are left with the DEIM error Lemma 2.1. If  $W = \widehat{W}$  but  $\mathcal{R}(S) \neq \mathcal{R}(\widehat{S})$ , then  $\sin \theta_{\max} = 0$ . Conversely, if  $W \neq \widehat{W}$  but  $\mathcal{R}(S) = \mathcal{R}(\widehat{S})$ , then  $\sin \psi_{\max} = 0$ .

Which bound is better? Our analysis in Theorem 4.2, which uses the perturbation results of oblique projectors, it is likely to be suboptimal. Note that in the second expression, we have the multiplicative factor  $\|\mathbb{D}\|_2 \|\widehat{\mathbb{D}}\|_2$ ; since both terms are at least 1, this expression can be large and clearly undesirable. This is what we see in Figure 1.

*Illustration of the bounds.* This example is based on [9, Example 3.1]. Let

$$(4.3) \quad f(t; \mu) = 10 \exp(-\mu t) (\cos(4\mu t) + \sin(4\mu t)), \quad 1 \leq t \leq 6, 0 \leq \mu \leq \pi.$$

The snapshot set is generated by taking  $n_\mu = 100$  evenly spaced values of  $\mu$  and  $n = 10,000$  evenly spaced points in time. The thin SVD of this matrix is computed, and the left singular vectors corresponding to the first 34 modes are used to define  $W$ . We use Algorithm 2 to obtain a randomized basis  $\widehat{W}$ , and we use an oversampling parameter  $p = 20$ . We consider two different target ranks  $r = 10, 20$ . To report the

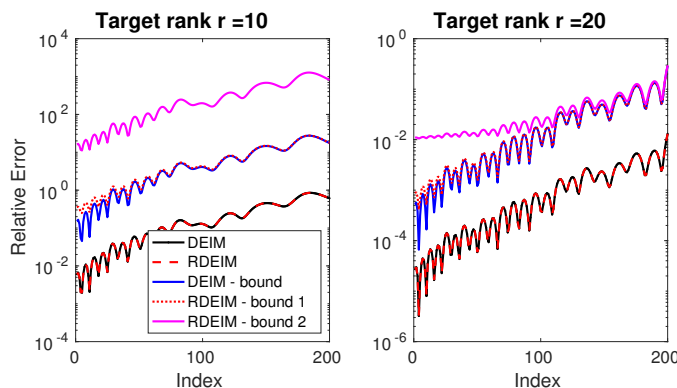


FIG. 1. Comparison of the R-DEIM error and the bounds in Theorem 4.1 (bound 1) and Theorem 4.2 (bound 2). For comparison we also include the DEIM error and the bound Lemma 2.1.

error we define the vectors  $f_j = [f(t_1; \mu_j) \ \dots \ f(t_n; \mu_j)]^\top$  for  $j = 1, \dots, n_\mu$ , and the relative error defined as

$$\text{Rel Err}(\mu_j) \equiv \frac{\|f_j - \mathbb{D}f_j\|_2}{\|f_j\|_2}, \quad j = 1, \dots, n_\mu.$$

The results of the comparison are provided in Figure 1. We also plot the bounds for the DEIM approximation Lemma 2.1 and the R-DEIM approximation Theorem 4.1. The point selection was done using the pivoted QR algorithm [9] (see subsection 4.4). We see that the error using the R-DEIM approximation closely follows the error of the DEIM algorithm. Although both the DEIM and the R-DEIM bounds overpredict the error, we see that the bound for R-DEIM is in close agreement with the bound for DEIM. We did not plot the bound from Theorem 4.2 because we did not find it to be very accurate.

**4.3. Accuracy of the subspaces.** The previous subsection reveals that the accuracy of R-DEIM depends on the largest canonical angle between the subspaces  $\mathcal{W}$  and  $\widehat{\mathcal{W}}$ , respectively. In this subsection, we shed more light on the accuracy of this quantity.

Assume that the snapshot matrix  $A$ , with singular vectors  $W = W_1$ , is perturbed to  $\widehat{A}$ . The perturbation may be either deterministic or random. Bounds for the canonical angles can be obtained from results known in the literature as the “sin theta” theorem. Below is one example of such a theorem.

LEMMA 4.3. Let  $\widehat{A} \in \mathbb{R}^{m \times n}$  be a perturbation of  $A \in \mathbb{R}^{m \times n}$  such that  $\sigma_r(A) - \sigma_{r+1}(\widehat{A}) > 0$ , and denote the left-singular vectors of  $\widehat{A}$  by  $\widehat{W}$ . Then

$$\sin \theta_{\max} \leq \frac{\max\{\|(A - \widehat{A})\widehat{Z}_1\|_2, \|(A^\top - \widehat{A}^\top)\widehat{W}_1\|\}}{\sigma_r - \sigma_{r+1}(\widehat{A})}.$$

*Proof.* This follows from [32, equation (4.15)].  $\square$

For some applications it may also be more convenient to bound the numerator with  $\|A - \widehat{A}\|_2$ .

When  $\widehat{W}$  is computed using Algorithm 3, these results can be made more precise.

**THEOREM 4.4.** Let  $\widehat{W} \in \mathbb{R}^{n \times r}$  be the DEIM basis obtained using Algorithm 3, and let  $p \geq 2$ . Assume that  $1 \leq r < \text{rank}(A)$  and the singular value ratio  $\gamma = \sigma_{r+1}/\sigma_r < 1$ , so that the subspace is well defined. Let the constant  $C$  be defined as

$$C \equiv \sqrt{\frac{r}{p-1}} + \frac{e\sqrt{(r+p)(n_s-r)}}{p}.$$

Then

$$\mathbb{E} \sin \theta_{\max} \leq \frac{\gamma^{2q+1}C}{1-\gamma}.$$

*Proof.* See [27, Theorem 4]. □

The interpretation of this theorem is that the largest canonical angle converges to 0 as the number of subspace iterations increases. More precisely, if we want  $\sin \theta_{\max}$  to be bounded, in expectation, by some positive parameter  $\varepsilon < 1$ , then the number of iterations (depending on  $\varepsilon$ ) should satisfy

$$q_\varepsilon \geq \frac{1}{2} \left( \frac{\log \varepsilon (1-\gamma) / (\gamma C)}{\log \gamma} \right).$$

This result also shows that the accuracy of the R-DEIM approximation depends on the singular value ratio  $\gamma$ —the smaller this ratio, the more accurate the subspace. When  $\gamma \approx 1$  the bound in Theorem 4.4 is devastating, but this is to be anticipated since this means that the subspace may be poorly defined.

**4.4. Randomized DEIM basis with deterministic point selection.** We briefly review the various choices of the selection operator  $S$  and review the error bounds associated with each choice. We refer to the condition number  $\|\mathbb{D}\|_2$  as the DEIM error constant. We then show how to combine the R-DEIM basis with existing point selection techniques.

In [7], a greedy approach (which we call the DEIM selection algorithm) was used to determine the point selection indices. For this algorithm, the DEIM error constant is bounded by  $\mathcal{O}(\sqrt{n})^r$ . However, numerical experiments showed that the bound for the DEIM error constant was pessimistic. Subsequent analysis in [28] improved this bound to  $\mathcal{O}(\sqrt{nr}2^r)$  and constructed an explicit matrix for which this bound could be attained asymptotically. They concluded that while the bound was large, numerical experiments showed that the point selection algorithm worked quite well, in practice. Recent work in [9] developed a different approach which used pivoted QR (PQR) on  $W^\top$  to obtain the selection operator  $S$ . As with the DEIM selection algorithm, the error constant can be large and could be attained by specially constructed adversarial cases. Numerical experiments suggest that the performance is comparable to the DEIM selection algorithm and is often better. More recently, the authors in [10] used the Gu–Eisenstat strong rank-revealing QR (sRRQR) algorithm [15] to obtain the bound

$$\|\mathbb{D}\|_2 \leq \sqrt{1 + \eta^2 r(n-r)} \equiv D_{\text{sRRQR}},$$

where  $\eta \geq 1$  is a user-specified parameter (the authors in [15] call this parameter  $f$ ). The DEIM error constant using sRRQR is significantly lower than that of the DEIM selection algorithm or the PQR algorithm; see Table 1 for the corresponding DEIM error constants. The cost of sRRQR is roughly  $\mathcal{O}(nr^2)$  ignoring logarithmic factors [15]. In numerical experiments, the performance of sRRQR is similar to that of PQR; therefore, we use the latter in our experiments because of its lower computational cost.

**Input:** Snapshot matrix  $A \in \mathbb{R}^{n \times n_s}$ , target rank  $r$ , oversampling parameter  $p \geq 1$ , number of iterations  $q \geq 0$ . User-defined parameter  $\eta \geq 1$ .

**Output:** Basis  $\widehat{W} \in \mathbb{R}^{n \times r}$  with orthonormal columns and selection operator  $\widehat{S} \in \mathbb{R}^{n \times r}$  defining the R-DEIM projector  $\widehat{\mathbb{D}} = \widehat{W}(\widehat{S}^\top \widehat{W})^\dagger \widehat{S}^\top$ .

- 1 Construct  $\widehat{W}$  as  $[\widehat{W}] = \text{RandSubspace}(A, r, p, q)$ .
- 2 Apply sRRQR algorithm with parameter  $\eta \geq 1$  to obtain

$$\widehat{W}^\top [\Pi_1 \quad \Pi_2] = Q [R_{11} \quad R_{12}].$$

Set  $\widehat{S} = \Pi_1 \in \mathbb{R}^{n \times r}$ .

**Algorithm 4:** Randomized DEIM approximation with sRRQR point selection.

Suppose the DEIM basis  $\widehat{W}$  is generated using the randomized algorithm Algorithm 3. We can apply sRRQR to  $\widehat{W}^\top$  to obtain

$$\widehat{W}^\top [\Pi_1 \quad \Pi_2] = Q [R_{11} \quad R_{12}].$$

Here  $[\Pi_1 \quad \Pi_2] \in \mathbb{R}^{n \times n}$  is a permutation matrix,  $Q \in \mathbb{R}^{r \times r}$  is orthogonal, and  $R_{11} \in \mathbb{R}^{r \times r}$  is upper triangular with positive diagonals. The selection operator  $S$  is taken to be  $\Pi_1 \in \mathbb{R}^{n \times r}$ , which selects well-conditioned rows of  $\widehat{W}$ . This is summarized in Algorithm 4. The following theorem analyzes the error in the resulting approximation.

**THEOREM 4.5.** Let  $\widehat{W}$  and  $\widehat{S}$  be the outputs of Algorithm 4, and define the R-DEIM approximation

$$\widehat{f}_{R-DEIM} = \widehat{W}(\widehat{S}^\top \widehat{W})^\dagger \widehat{S}^\top f.$$

With the assumptions and notation of Theorem 4.4, the expected error in the R-DEIM approximation satisfies

$$\mathbb{E}_\Omega \|f - \widehat{f}_{R-DEIM}\|_2 \leq D_{sRRQR} \left( \|(I_n - P_W)f\|_2 + \frac{\gamma^{2q+1}C}{1-\gamma} \|P_W f\|_2 \right).$$

*Proof.* By [10, Lemma 2.1] we obtain  $\|\widehat{\mathbb{D}}\|_2 = \|(\widehat{S}^\top \widehat{W})^{-1}\|_2 \leq D_{sRRQR}$ . This ensures that  $\text{rank}(\widehat{\mathbb{D}}) = \text{rank}(\widehat{S}^\top \widehat{W}) = r$ . The assumption  $1 \leq r < \text{rank}(A)$  ensures  $\widehat{\mathbb{D}} \neq 0$  or  $\widehat{\mathbb{D}} \neq I_n$ ; therefore, we can apply Theorem 4.1 to obtain

$$\|f - \widehat{f}_{R-DEIM}\|_2 \leq D_{sRRQR} (\|(I - P_W)f\|_2 + \sin \theta_{\max} \|P_W f\|_2).$$

We apply the result from Theorem 4.4 to the above equation, which completes the proof.  $\square$

**5. Randomized point selection.** To the best of our knowledge, the best known bounds for the DEIM error constant are obtained using the sRRQR algorithm; however, as mentioned earlier, the computational cost of the sRRQR algorithm can be high and may be prohibitively expensive for applications of interest. To tackle this computational challenge, sampling-based randomized approaches have been previously proposed [24, 9, 23].

The sampling approach for point selection randomly samples  $s \geq r$  indices from the index set  $\{1, \dots, n\}$ , according to a prespecified discrete probability distribution, to determine the selection operator  $S$ . The computational cost of point selection by

sampling is  $\mathcal{O}(ns)$ ; in comparison, both the DEIM selection and PQR algorithms cost  $\mathcal{O}(nr^2)$ . Sampling-based techniques are readily parallelizable and therefore advantageous for large-scale problems. There are two competing issues to consider when deciding between sampling strategies: the DEIM error constant and the number of required samples. A low DEIM error constant is desirable but may require many samples  $s$ , which increases the computational cost. An example of a sampling strategy is uniform sampling, either with or without replacement. When the DEIM basis  $W$  has high coherence (for a definition, see (5.1) and the discussion below it), the number of samples required  $s$  can be large to ensure a small DEIM error constant. Therefore, we do not use uniform sampling for the point selection. See [19] for additional discussion on the effect of coherence on sampling from matrices with orthonormal columns. In this section, we propose randomized algorithms for the selection operator  $S$  and develop bounds for the proposed selection operators.

We propose two different randomized point selection techniques. The first method is based on leverage scores (see (5.1) for a definition). While the point selection stage is computationally efficient, the overall cost can be high since  $s \sim \mathcal{O}(r \log r)$  samples need to be drawn—this also corresponds to the number of points at which the nonlinear function  $f$  is evaluated. In certain applications, it is desirable to pick only  $r$  samples. To address this issue, we propose a hybrid point selection algorithm which retains the computational advantages of sampling-based point selection but only samples  $r$  indices, thereby retaining the favorable properties of deterministic methods.

**5.1. Randomized point selection with standard DEIM basis.** We first develop algorithms for randomized point selection with the standard DEIM basis  $W \in \mathbb{R}^{n \times r}$ . The analysis can be extended to the randomized DEIM basis and is considered in the next subsection.

The *leverage scores* of  $W$  are defined to be

$$(5.1) \quad \ell_j \equiv \|e_j^\top W\|_2^2, \quad j = 1, \dots, n,$$

where  $e_j$  is the  $j$ th column of an  $n \times n$  identity matrix. Equivalently, the leverage scores are the squared row norms of  $W$  or, alternatively, the diagonals of the projector  $P_W = WW^\top$ . The largest leverage score is known as the *coherence*, and the sum of the leverage scores satisfies  $\sum_{j=1}^n \ell_j = r$ . Based on the leverage scores, we can define the following discrete probability mass function (pmf)  $\pi_j = \ell_j/r$  for  $j = 1, \dots, n$ . In what follows, we instead use the related pmf

$$(5.2) \quad \pi_j^\beta = \frac{\beta \ell_j}{r} + \frac{(1-\beta)}{n}, \quad j = 1, \dots, n.$$

Here  $0 < \beta < 1$  is a user-defined constant, and the modified pmf is a convex combination of the leverage score pmf and the uniform pmf. This modified pmf is beneficial since it can handle rows with zero leverage scores.

*Leverage score point selection approach.* The leverage score approach constructs a sampling matrix by selecting indices  $\{t_1, \dots, t_s\}$ , independently and with replacement, from the index set  $\{1, \dots, n\}$ , with probabilities given by (5.2). We construct the selection operator  $S$  as follows: the  $j$ th column of  $S$  is  $e_{t_j}/\sqrt{s\pi_{t_j}^\beta}$  for  $j = 1, \dots, s$ . This is summarized in Algorithm 5. These columns are scaled in this manner to ensure that  $SS^\top$  equals the identity matrix  $I_n$  in expectation; that is,  $SS^\top$  is an unbiased estimator of the identity matrix.

LEMMA 5.1. Let  $W \in \mathbb{R}^{n \times r}$  be a matrix with orthonormal columns. Let the selection operator be constructed as described in Algorithm 5. Then  $\mathbb{E}[SS^\top] = I_n$ .

*Proof.* We write  $SS^\top$  as the outer product representation

$$SS^\top = \sum_{j=1}^s Y_j, \quad Y_j \equiv \frac{1}{s\pi_{t_j}^\beta} e_{t_j} e_{t_j}^\top.$$

It is easy to verify that  $\mathbb{E}[Y_j] = I_n/s$ . By the linearity of expectations, it follows that  $\mathbb{E}[SS^\top] = I_n$ .  $\square$

The number of selected indices  $s$  is chosen to be

$$(5.3) \quad C_{LS} \equiv \left\lceil \frac{2r}{\beta\epsilon^2} \log(r/\delta) \right\rceil,$$

where  $\lceil \cdot \rceil$  is the ceiling function. This choice will be justified in Theorem 5.2.

**Input:** Matrix  $W \in \mathbb{R}^{n \times r}$  with orthonormal columns, number of samples  $s \geq r$ . Probabilities  $\{\pi_j^\beta\}_{j=1}^n$  defined in (5.2).

**Output:** Matrix  $S \in \mathbb{R}^{n \times C_{LS}}$

```

1 for  $j = 1, \dots, s$  do
2   Select index  $t_j$ , independently and with replacement, from  $\{1, \dots, n\}$ 
   with probabilities  $\{\pi_j^\beta\}_{j=1}^n$ .
3   Set  $S(:, j) = \frac{1}{\sqrt{s\pi_{t_j}^\beta}} e_{t_j}$ .
4 end
```

**Algorithm 5:** Leverage score point selection. Call as  $[S] = \text{LeverageScorePS}(W, \{\pi_j^\beta\}_{j=1}^n, s)$ .

*Hybrid point selection approach.* The hybrid approach we propose has two stages: a randomized point selection stage, which uses the leverage score distribution to select  $C_{LS}$  indices, and a deterministic approach, which uses the sRRQR algorithm to choose  $r$  indices out of  $C_{LS}$ .

**1. Randomized stage.** In the first stage, we use leverage score approach (Algorithm 5) with  $s = C_{LS}$  and denote the point selection matrix  $S_1 \in \mathbb{R}^{n \times C_{LS}}$ . The resulting matrix  $S_1^\top W$  extracts  $C_{LS}$  rows from  $W$ , with appropriate scaling.

**2. Deterministic stage.** In the second stage, we apply sRRQR to  $W^\top S_1$  to select exactly  $r$  rows from the matrix  $S_1^\top W$ . Let  $S_2 \in \mathbb{R}^{C_{LS} \times r}$  denote the point selection matrix obtained using sRRQR.

Denote this composite selection matrix as  $S = S_1 S_2 \in \mathbb{R}^{n \times r}$ , which consists of columns from the  $n \times n$  identity matrix that are scaled by the appropriate factors. Similar algorithms have been proposed in [5, 6]. However, the specific choice of the sampling distribution and the subsequent analysis are different.

In the analysis of the DEIM approximation Lemma 2.1, the condition number is determined by the DEIM error constant  $\|\mathbb{D}\|_2$ . We derive bounds for this constant when the selection operator is obtained using the leverage score and the hybrid approaches.

THEOREM 5.2. Let  $W \in \mathbb{R}^{n \times r}$  be a fixed matrix with orthonormal columns, and let  $0 < \epsilon, \delta < 1$  be user-defined parameters. Let  $S_1 \in \mathbb{R}^{n \times C_{LS}}$  and  $S = S_1 S_2 \in \mathbb{R}^{n \times r}$  be



**Input:** Matrix  $W \in \mathbb{R}^{n \times r}$  with orthonormal columns, number of samples  $C_{LS}$ , parameter  $\eta \geq 1$ .  
 Probabilities  $\{\pi_j^\beta\}_{j=1}^n$  defined in (5.2).  
**Output:** Matrices  $S_1 \in \mathbb{R}^{n \times C_{LS}}$  and  $S_2 \in \mathbb{R}^{C_{LS} \times r}$  that define  $S = S_1 S_2$ .  
 /\* Stage 1. Randomized Stage \*/  
 1  $[S_1] = \text{LeverageScorePS}(W, \{\pi_j^\beta\}_{j=1}^n, C_{LS})$ .  
 /\* Stage 2. Deterministic Stage \*/  
 2 Perform sRRQR with parameter  $\eta$  on  $W^\top S_1$ .  
 $W^\top S_1 [\Pi_1 \ \Pi_2] = Q [R_{11} \ R_{12}]$ .  
 3 Set  $S_2 = \Pi_1$ .

**Algorithm 6:** Hybrid point selection.

the outputs of Algorithms 5 and 6, with the number of samples  $s = C_{LS}$ , and define the corresponding DEIM operators  $\mathbb{D}_{LS} \equiv W(S_1^\top W)^\dagger S_1^\top$  and  $\mathbb{D}_{Hy} \equiv W(S^\top W)^\dagger S^\top$ . With probability at least  $1 - \delta$ , the DEIM error constant for

- the leverage score approach satisfies

$$\|\mathbb{D}_{LS}\|_2 \leq \sqrt{\frac{n/C_{LS}}{(1-\beta)(1-\epsilon)}} \equiv D_{LS};$$

- the hybrid approach satisfies

$$\|\mathbb{D}_{Hy}\|_2 \leq D_{LS} \sqrt{1 + \eta^2 r (C_{LS} - r)} \equiv D_{Hy}.$$

For both the leverage score and the hybrid point selection approaches, the point selection operator  $S$  contains (appropriately scaled) columns from the identity matrix. A few differences from the standard DEIM approach are worth pointing out (assume that  $\mathbb{D} \neq 0$  and  $\mathbb{D} \neq I_n$ ):

1. The matrix  $S$  no longer has orthonormal columns; therefore,

$$\|\mathbb{D}\|_2 = \|I_n - \mathbb{D}\|_2 = \|(S^\top W)^\dagger S^\top\|_2.$$

The first equality holds because  $\mathbb{D}$  is an oblique projector; see (2.3).

2. Second, the DEIM implementation has to be altered appropriately. There are two steps: first, the components of  $f(\cdot)$  as determined by  $S$  are extracted, and second, these components are scaled by the corresponding scaling factor.
3. We can combine Theorem 5.2 along with Lemma 2.1 to derive the error in the DEIM approximation. With the assumption and notation of Theorem 5.2, the following bounds hold with probability at least  $1 - \delta$ :

$$\|f - \mathbb{D}_{LS} f\|_2 \leq D_{LS} \|(I_n - P_W) f\|_2, \quad \|f - \mathbb{D}_{Hy} f\|_2 \leq D_{Hy} \|(I_n - P_W) f\|_2.$$

This is obtained by combining Theorem 5.2 with Lemma 2.1.

*Theorem 5.2.* The DEIM operators satisfy the inequalities

$$\|\mathbb{D}_{LS}\|_2 = \|W(S_1^\top W)^\dagger S_1^\top\|_2 \leq \|(S_1^\top W)^\dagger\|_2 \|S_1\|_2$$

and

$$\|\mathbb{D}_{Hy}\|_2 = \|W(S^\top W)^\dagger S^\top\|_2 \leq \|(S^\top W)^\dagger\|_2 \|S\|_2.$$

*Leverage scores approach.* We have to find upper bounds for  $\|S_1\|_2$  and  $\|(S_1^\top W)^\dagger\|_2$ . To bound  $\|(W^\top S)^\dagger\|_2$ , we first observe that

$$\pi_j^\beta \geq \frac{\beta \ell_j}{r}.$$

If we take the number of columns of  $S_1$  taken to be  $s = C_{\text{LS}}$ , the operator  $S_1$  satisfies the conditions of [18, Theorem 6.2]. It follows from this theorem that with probability at least  $1 - \delta$

$$(5.4) \quad 1 - \epsilon \leq \sigma_r^2(S_1^\top W) \quad \text{or} \quad \|(S_1^\top W)^\dagger\|_2 \leq \frac{1}{\sqrt{1 - \epsilon}}.$$

We now bound  $\|S_1\|_2$ . If  $\beta = 1$ , the pmf only contains contributions from the leverage scores. The norm  $\|S_1\|_2$  may be unbounded if zero leverage scores are encountered. However, if  $\beta \neq 0$ , since  $\pi_j^\beta \geq \frac{1-\beta}{n}$ ,

$$(5.5) \quad \|S_1\|_2 \leq \max_{1 \leq j \leq n} \sqrt{\frac{1}{C_{\text{LS}} \pi_j^\beta}} \leq \sqrt{\frac{n}{C_{\text{LS}}(1 - \beta)}}.$$

Putting this together, we get

$$\|\mathbb{D}_{\text{LS}}\|_2 \leq \|(S_1^\top W)^\dagger\|_2 \|S_1\|_2 \leq \sqrt{\frac{n/C_{\text{LS}}}{(1 - \epsilon)(1 - \beta)}}.$$

*Hybrid approach.* Similar to the previous part of the proof, we have to bound  $\|(S^\top W)^\dagger\|_2$  and  $\|S\|_2$ . Applying sRRQR to  $(W^\top S_1)$  gives

$$(W^\top S_1) \begin{bmatrix} \Pi_1 & \Pi_2 \end{bmatrix} = Q \begin{bmatrix} R_{11} & R_{12} \end{bmatrix},$$

where  $Q \in \mathbb{R}^{r \times r}$  is an orthogonal matrix;  $R_{11} \in \mathbb{R}^{r \times r}$  is upper triangular; and  $\begin{bmatrix} \Pi_1 & \Pi_2 \end{bmatrix} \in \mathbb{R}^{C_{\text{LS}} \times C_{\text{LS}}}$  is a permutation matrix with  $\Pi_1 \in \mathbb{R}^{C_{\text{LS}} \times r}$ . Recall that  $S_2 = \Pi_1$  and that  $S = S_1 S_2$ . Then, the singular value bounds [15, Lemma 3.1] ensure that

$$\frac{\sigma_r(S_2^\top W)}{\sqrt{1 + \eta^2 r(C_{\text{LS}} - r)}} \leq \sigma_r(S^\top W).$$

Therefore,

$$\|(S^\top W)^{-1}\|_2 \leq \sqrt{1 + \eta^2 r(C_{\text{LS}} - r)} \|(S_1^\top W)^\dagger\|_2 \leq \sqrt{\frac{1 + \eta^2 r(C_{\text{LS}} - r)}{1 - \epsilon}}.$$

The bound for  $\|(S_1^\top W)^\dagger\|_2$  follows from (5.4). To finish the proof, it remains to bound  $\|S\|_2 \leq \|S_1\|_2 \|S_2\|_2$ . Since  $S_2$  contains columns from the identity matrix,  $\|S_2\|_2 = 1$  and we have an upper bound for  $\|S_1\|_2$  in (5.5). Combining all the intermediate steps, the bound for  $\mathbb{D}_{\text{Hy}}$  then follows readily.  $\square$

To shed light on the DEIM error constants, we give some representative values. Suppose  $\beta = 1/2$ ,  $\eta = 2$ , and  $\epsilon = 9/10$ , the number of samples required is

$$C_{\text{LS}} \leq 5r \log(r/\delta),$$

and

$$D_{\text{LS}} = \sqrt{\frac{20n}{C_{\text{LS}}}}, \quad D_{\text{Hy}} \leq D_{\text{LS}} \sqrt{1 + 20r^2 \log(r/\delta)}.$$

In terms of asymptotic complexity, the DEIM error constant for the leverage score algorithm is  $\mathcal{O}(\sqrt{n/C_{\text{LS}}})$ , whereas for the hybrid algorithm it is  $\mathcal{O}(\sqrt{nr})$ . This is to be compared with the sRRQR algorithm for which the DEIM error constant is  $\mathcal{O}(\sqrt{nr})$ . In terms of computational costs, the cost of computing and sampling the leverage scores is  $\mathcal{O}(nr)$  with an additional  $\mathcal{O}(r^2 C_{\text{LS}})$  for factorizing  $S^\top W$ . The hybrid algorithm also requires  $\mathcal{O}(nr)$  for computing the leverage scores and sampling. In the second stage, sRRQR is applied to a matrix of size  $r \times C_{\text{LS}}$ ; this cost is  $\mathcal{O}(r^2 C_{\text{LS}})$  and is independent of  $n$ . An additional cost for factorizing  $S^\top W$  is  $\mathcal{O}(r^3)$ . This is summarized in Table 1. In summary, the hybrid approach is both computationally efficient compared to other point selection methods and has comparable DEIM error constants.

TABLE 1

Summary of various point selection techniques. Here  $0 < \epsilon, \delta < 1$  are user-defined parameters. We take the parameter  $\beta = 1/2$ . For the computational cost, terms that do not depend on  $n$  are not considered. A note about the \* entries: the corresponding DEIM error constants each hold independently with probability at least  $1 - \delta$ .

Method	# indices	Comp. cost	DEIM error constant	Reference
DEIM	$r$	$\mathcal{O}(nr^2)$	$\mathcal{O}(\sqrt{nr}2^r)$	[28]
PQR	$r$	$\mathcal{O}(nr^2)$	$\mathcal{O}(\sqrt{n}2^r)$	[9]
sRRQR	$r$	$\mathcal{O}(nr^2)$	$\mathcal{O}(\sqrt{nr})$	[10]
LS*	$\mathcal{O}\left(\frac{r \log(r/\delta)}{\epsilon^2}\right)$	$\mathcal{O}(nr)$	$\mathcal{O}\left(\sqrt{\frac{n\epsilon^2}{r \log(r/\delta)(1-\epsilon)}}\right)$	Theorem 5.2
Hybrid*	$r$	$\mathcal{O}(nr)$	$\mathcal{O}\left(\sqrt{\frac{nr}{(1-\epsilon)}}\right)$	Theorem 5.2

The important point is that the error constants of the hybrid point selection approach are comparable with the best known deterministic bounds (using sRRQR); however, the computational cost is far less than that of sRRQR or the other deterministic approaches. We advocate the hybrid approach since it has reasonable computational cost, is accurate, and selects exactly  $r$  indices from the nonlinear function.

**5.2. Randomized point selection with randomized DEIM basis.** Thus far, we have described randomized point selection techniques assuming the availability of the standard DEIM basis  $W$ . If the standard basis  $W \in \mathbb{R}^{n \times r}$  is computationally intensive to compute, we can alternatively use an R-DEIM basis (obtained, for example, using Algorithm 1 or Algorithm 3). The leverage scores, and the corresponding sampling probabilities  $\{\pi_j^\beta\}_{j=1}^n$ , are now computed corresponding to the basis  $\widehat{W}$  rather than the standard DEIM basis  $W$ . To determine the selection operator, one may use either randomized point selection technique—Algorithm 5 or Algorithm 6. In the following result, we quantify the error in the resulting DEIM approximation. The main challenge is now there are two sources of randomness: the sampling matrix  $\Omega$  as well as the sampling strategy that determines the selection operator  $S$ .

**THEOREM 5.3.** Let  $\widehat{W} \in \mathbb{R}^{n \times r}$  be obtained using Algorithm 3 with oversampling parameter  $p \geq 2$ , and let  $\widehat{S} \in \mathbb{R}^{n \times r}$  be obtained using the hybrid point selection algorithm, Algorithm 6, to define the randomized DEIM operator  $\widehat{\mathbb{D}}_{Hy} = \widehat{W}(\widehat{S}^\top \widehat{W})^\dagger \widehat{S}^\top$ . Consider the same assumptions as in Theorem 4.4. Let  $0 < \delta < 1/2$  be a user-defined parameter. With probability at least  $1 - 2\delta$

$$\|f - \widehat{\mathbb{D}}_{Hy} f\|_2 \leq D_{Hy} \left( \|(I_n - P_W)f\|_2 + \frac{\gamma^{2q+1}}{1-\gamma} C_d \|P_W f\|_2 \right),$$

where  $D_{Hy}$  is as defined in Theorem 5.2 and

$$C_d \equiv \frac{e\sqrt{r+p}}{p+1} \left(\frac{2}{\delta}\right)^{1/(p+1)} \left(\sqrt{n_s - r} + \sqrt{r+p} + \sqrt{2 \log \frac{2}{\delta}}\right).$$

*Proof.* Define the event

$$\mathcal{E} = \left\{ \Omega \left| \sin \theta_{\max} \leq \frac{\gamma^{2q+1}}{1-\gamma} C_d \right. \right\}.$$

Combining [27, Theorems 4 and 6], the probability of the complementary event satisfies  $\mathbb{P}(\mathcal{E}^c) \leq \delta$ . Similarly, define the event

$$\mathcal{F} = \left\{ \hat{S}, \Omega \left| \|f - \mathbb{D}_{Hy} f\|_2 > D_{Hy} \left( \|(I_n - P_W)f\|_2 + \frac{\gamma^{2q+1}}{1-\gamma} C_d \|P_W f\|_2 \right) \right. \right\}.$$

By Theorem 5.2,  $\mathbb{P}(\|\mathbb{D}_{Hy}\|_2 \leq D_{Hy} | \mathcal{E}) \geq 1 - \delta$  and  $\text{rank}(\mathbb{D}_{Hy}) = r$ . The assumption  $1 \leq r < \text{rank}(A)$  ensures  $\mathbb{D}_{Hy} \neq 0$  and  $\mathbb{D}_{Hy} \neq I_n$ . By Theorem 4.1, we have

$$\mathbb{P} \left( \|f - \mathbb{D}_{Hy} f\|_2 \leq D_{Hy} \left( \|(I_n - P_W)f\|_2 + \frac{\gamma^{2q+1}}{1-\gamma} C_d \|P_W f\|_2 \right) \middle| \mathcal{E} \right) \geq 1 - \delta,$$

or the complementary event  $\mathcal{F}$  satisfies  $\mathbb{P}(\mathcal{F} | \mathcal{E}) \leq \delta$ . By the law of total probability,

$$\mathbb{P}(\mathcal{F}) = \mathbb{P}(\mathcal{F} | \mathcal{E}) \mathbb{P}(\mathcal{E}) + \mathbb{P}(\mathcal{F} | \mathcal{E}^c) \mathbb{P}(\mathcal{E}^c) \leq \mathbb{P}(\mathcal{F} | \mathcal{E}) + \mathbb{P}(\mathcal{E}^c).$$

Therefore,  $\mathbb{P}(\mathcal{F}) \leq \delta + \delta = 2\delta$ . The complementary event satisfies the advertised bound.  $\square$

A similar result can be derived for the leverage score approach, but we omit the details.

**6. Numerical experiments.** In subsection 6.1 we investigate the accuracy of the DEIM basis generated using the randomized algorithms discussed in section 3. In subsection 6.2 we investigate the performance of the randomized point selection algorithms proposed in section 5. In subsection 6.3 we apply these randomized algorithms to a large-scale PDE-based application. All the timing results were computed on a computing cluster in which each node has an Intel(R) Xeon(R) CPU E5-2690 processor, with 8-core CPUs at 2.90GHz and 128GB of DDR3 RAM. The code was implemented and tested in MATLAB 2018a, and the operating system was Ubuntu 16.04.

**6.1. Example 1: Randomized range finder.** In our first example, we consider the setup of the synthetic example in [22, section 2.3]. The spatial domain and the parameter domain are both taken to be  $\mathcal{D}_s = [0, 1]^2$ . We define the function  $g$  as

$$g(x_1, x_2; \mu_1, \mu_2) \equiv \frac{1}{\sqrt{h(x_1; \mu_1) + h(x_2; \mu_2) + 0.1^2}},$$

where  $h(z; \mu) = ((1 - z) - (0.99 \cdot \mu - 1))^2$ . The function that is to be interpolated is

$$(6.1) \quad f(x_1, x_2; \mu_1, \mu_2) = g(x_1, x_2; \mu_1, \mu_2) + g(1 - x_1, 1 - x_2; 1 - \mu_1, 1 - \mu_2) \\ + g(1 - x_1, x_2; 1 - \mu_1, \mu_2) + g(x_1, 1 - x_2; \mu_1, 1 - \mu_2).$$

Depending on the parameter  $\mu$ , the function  $f$  has a sharp peak in one of the four corners of  $\mathcal{D}_s$ . The function is discretized on a  $100 \times 100$  grid in  $\mathcal{D}_s$  with  $n = 10,000$ , and parameter samples are drawn from a  $25 \times 25$  equispaced grid in  $\mathcal{D}$ . These  $n_s = 625$  snapshots are stored in the snapshot matrix  $A$  and are used to construct the DEIM approximation.

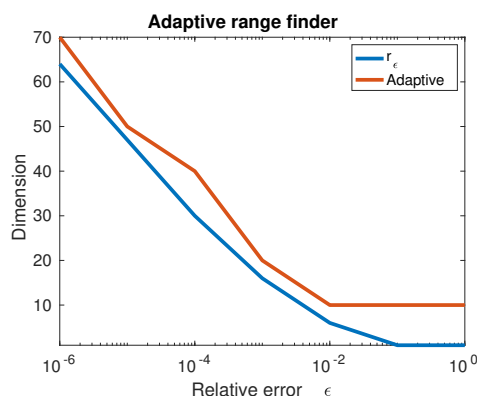


FIG. 2. The dimension of the basis returned by the adaptive range finding algorithm compared with the truncation indices based on a cutoff tolerance.

**6.1.1. Adaptive range finder.** If the target subspace dimension is not known a priori, we use the adaptive procedure outlined in subsection 3.2. With the same settings as in the previous example, we use Algorithm 2 for determining the range. For comparison, we consider the standard DEIM basis. Given a user-defined tolerance  $\epsilon > 0$ , the dimension of the standard DEIM basis  $W$  (assuming  $n \neq n_s$ ) is taken to be

$$r_\epsilon(A) = \min \left\{ r \left| \sum_{k=r+1}^{n_s} \sigma_k^2(A) > \epsilon \sum_{k=1}^{n_s} \sigma_k^2(A) \right. \right\},$$

since this ensures that  $W$  satisfies

$$\|(I_n - WW^\top)A\|_F^2 \leq \epsilon \|A\|_F^2.$$

Note that we say “the dimension of a basis” even though dimension is an attribute of the subspace spanned by the basis vectors.

The standard DEIM basis has the smallest dimension satisfying the above equality; this follows from the optimality of the SVD. Therefore, the dimension of the basis returned by Algorithm 2 must be at least as large as the dimension of the standard DEIM basis. In this experiment, we investigate how close these two dimensions are. The tolerance  $\epsilon$  is varied as  $\epsilon \in \{10^2, \dots, 10^{-6}\}$ . For the adaptive algorithm, the block size was set to be 10, and the maximum number of iterations was taken to be 40. In Figure 2, we compare the two dimensions depending on the cutoff tolerance  $\epsilon$ . From the figure, it is clear that as the tolerance  $\epsilon$  decreases, both dimensions increase. Second, it can be seen that both dimensions are in good agreement, and the dimension returned by the adaptive randomized algorithm is only slightly larger than  $r_\epsilon$ , demonstrating that it can be used in real applications. The accuracy of the R-DEIM basis is further investigated in subsection 6.1.3.

**6.1.2. Subspace iterations  $q$  and oversampling parameter  $p$ .** In our next experiment, we investigate the effects of the number of subspace iterations  $q$  and the oversampling parameter  $p$ . Theorem 4.4 guarantees that with increasing iterations  $q$  the factor  $\gamma^{2q+1}$  subdues the influence of the constant  $C$ , since  $\gamma$  is assumed to be less than 1. Similarly, note that the oversampling parameter  $p$  appears in the denominator of  $C$ ; therefore, increasing the oversampling parameter  $p$  results in more accurate subspace computation. Both [14, 16] recommend choosing  $p = 10 - 20$ . We

found this choice of parameters to be satisfactory in our experiments, and Figure 3 confirms these findings.

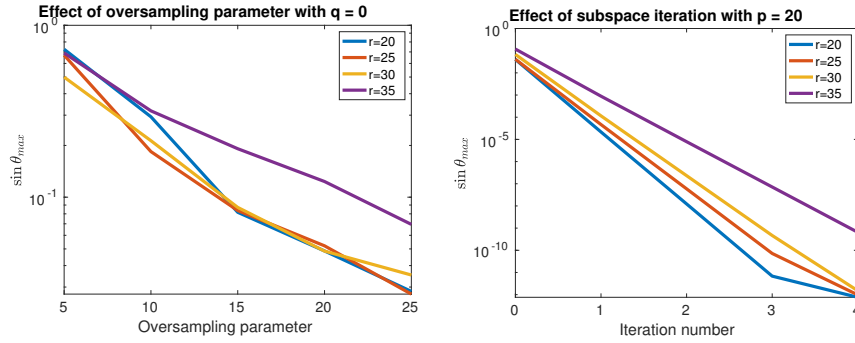


FIG. 3. Example 1: Effect of (left) increasing oversampling parameter  $p \in \{5, 10, 15, 20, 25\}$  (we fix  $q = 0$ ) and (right) increasing subspace iterations  $q \in \{0, 1, 2, 3, 4\}$  (we fix  $p = 20$ ) on the accuracy of the R-DEIM basis measured as  $\sin \theta_{\max} = \|P_W - P_{\hat{W}}\|_2$ .

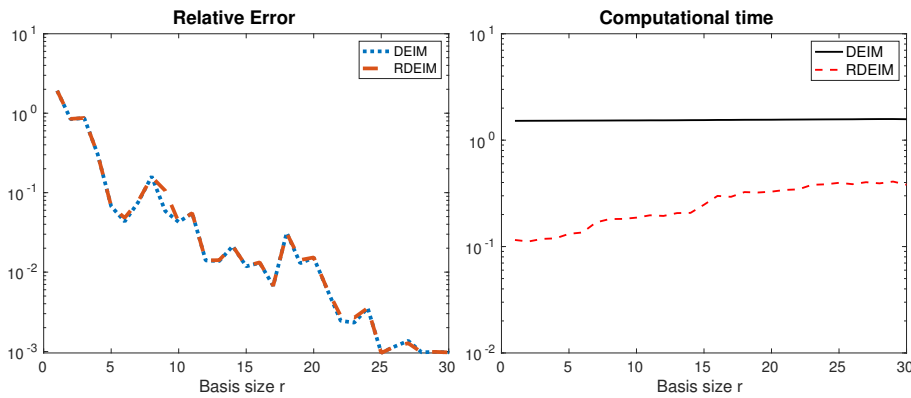


FIG. 4. Example 1: (Left) Relative error using the DEIM and the R-DEIM approximation plotted against basis dimension  $r$ . R-DEIM basis was computed using Algorithm 1 with  $r = 30$  and oversampling parameter  $p = 10$ . (Right) Timing results using DEIM and R-DEIM plotted against basis dimension  $r$ .

**6.1.3. Accuracy of the R-DEIM basis.** In this experiment, we compare the accuracy of the R-DEIM basis with that of the standard DEIM basis for Example 1. For the R-DEIM approximation, we use an oversampling parameter  $p = 10$ . The dimension of the R-DEIM basis is varied until  $r = 30$ , and the error is compared with the standard DEIM approximation. We used the PQR algorithm for point selection. To account for the randomness, the resulting error in the R-DEIM approximation was averaged over 100 runs. As can be seen from Figure 4, the error in the R-DEIM approximation is comparable to the error in the standard DEIM approximation. We also see that the time for computing the R-DEIM approximation is far lower than the time for the DEIM approximation.

**6.2. Example 2: Point selection.** This example is a continuation of Example 1, but we now focus on the randomized point selection. For this example, we use the

standard DEIM basis, i.e., the basis computed using the left-singular vectors of the snapshot matrix  $A$ .

We first compare the two randomized point selection techniques—leverage scores (LS), and the hybrid point selection algorithms—with the PQR algorithm. Recall that theory suggests that we have to choose the number of samples according to the formula in Theorem 5.2. Suppose we choose the parameters  $\epsilon = 0.99$ , which ensures  $1/\sqrt{1-\epsilon} = 10$  and  $\delta = 0.01$ . Our numerical experiments showed that the number of samples required by the LS point selection algorithm appear to be an overestimate. In fact, the number of samples can sometimes exceed  $n$ —which is antithetical to the spirit of the DEIM approximation. In practice, we found  $\lceil 3r \log r \rceil$  samples sufficient to provide accurate approximations. We used the same number of samples for the hybrid point selection algorithm.

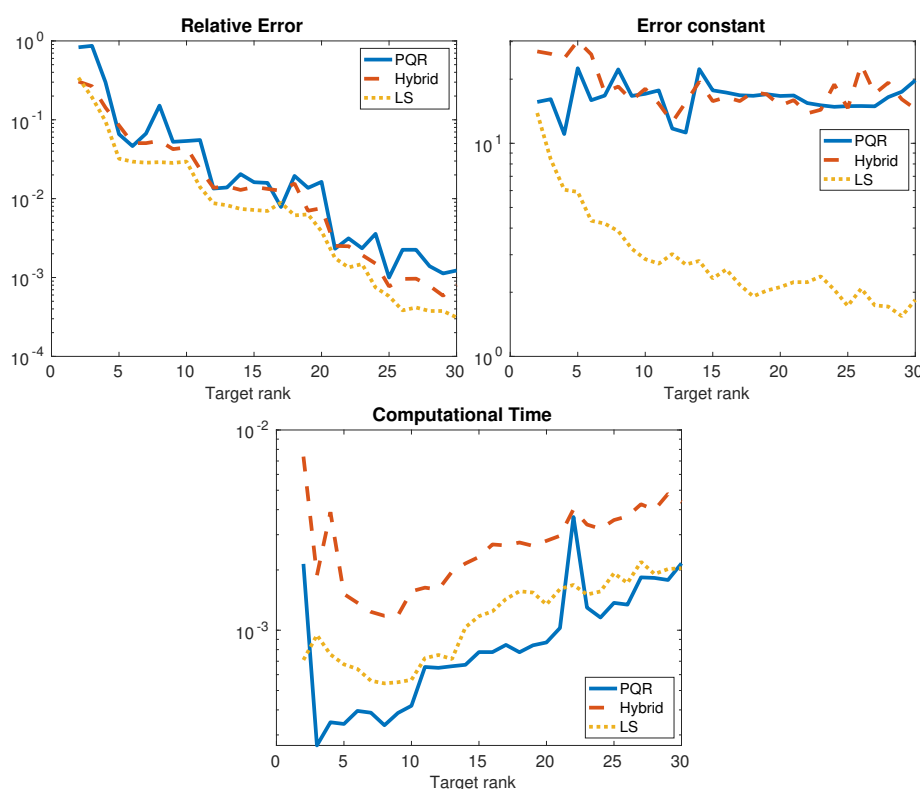


FIG. 5. Example 2: (Top left) Error in the DEIM approximation, (top right) the DEIM error constants  $\|\mathbb{D}\|_2$  using the PQR algorithm and the hybrid approach Algorithm 6, and (bottom) the computational time for all the point selection methods.

In Figure 5, we compare the error in the DEIM approximations and the corresponding error constants  $\|\mathbb{D}\|_2$  for the various algorithms. The error constant is much smaller for the LS point selection algorithm compared to both the hybrid and deterministic approaches. This is because the number of samples in the LS point selection algorithm is much larger than  $r$ . However, the accuracy of the DEIM approximation is comparable for all three methods and does not appear to be significantly affected by the choice of the point selection method. The computational time for each point selection method is also reported; the timings were averaged over 10 runs.

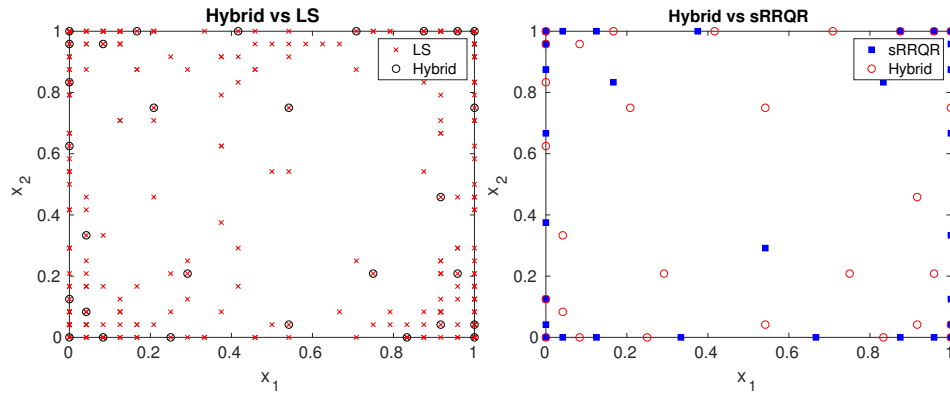


FIG. 6. Example 2: The indices chosen by the various point selection methods.

To provide more insight into the hybrid point selection algorithm, we compare it with the sRRQR and LS point selection algorithms. In Figure 6, we plot the point selected by the LS sampling approach in the left panel. By construction, the hybrid approach subsamples from the points selected by the LS algorithm, and these points are overlaid in the figure. In the right panel of the same figure, the hybrid points are compared with those selected by sRRQR. It is interesting to note that several points overlap between the hybrid and the sRRQR approaches, even though they were selected using different algorithms.

Our conclusion is that the proposed hybrid point selection algorithm is a good compromise between the deterministic (e.g., PQR, sRRQR) and randomized point selection algorithms. Compared to the deterministic algorithms, it is computationally efficient and has comparable accuracy. The hybrid point selection algorithm also has comparable accuracy to the LS algorithm but is advantageous since only  $r$  components of the nonlinear function need to be evaluated.

**6.3. Example 3: PDE-based application.** We consider an example from [25, section 8.4] involving a parameterized advection-diffusion PDE that models, for example, the evolution of the concentration of a pollutant. Consider the following PDE defined on a domain  $\mathcal{D} = [0, 1]^2$  with boundary  $\partial\mathcal{D}$ :

$$(6.2) \quad -\mu_1 \Delta u + \mathbf{b}(\mu_2) \cdot \nabla u + a_0 u = s(\mathbf{x}; \boldsymbol{\mu}), \quad \mathbf{x} \in \mathcal{D},$$

$$(6.3) \quad \mu_1 \mathbf{n} \cdot \nabla u = 0, \quad \mathbf{x} \in \partial\mathcal{D}.$$

Here,  $\boldsymbol{\mu} = [\mu_1 \ \dots \ \mu_5]$ ,  $a_0$  is a positive constant, and  $\mathbf{n}$  is the normal vector. The wind velocity is taken to be  $\mathbf{b}(\mu_2) = [\cos \mu_2, \sin \mu_2]$ , which is a constant in space but depends nonlinearly on the parameter  $\mu_2$ . The source term  $s(\boldsymbol{\mu})$  has the form of a Gaussian function centered at  $(\mu_3, \mu_4)$  and spread  $\mu_5$ , i.e.,

$$s(\mathbf{x}; \boldsymbol{\mu}) = \exp \left( -\frac{(x_1 - \mu_3)^2 + (x_2 - \mu_4)^2}{\mu_5^2} \right).$$

The cost of solving the PDE for many different values of  $\boldsymbol{\mu}$  is high, and therefore, it is computationally beneficial to develop a reduced order model (ROM) that makes the online solution of the PDE feasible. The nonlinear dependence on the parameters arises from the source term  $s(\mathbf{x}; \boldsymbol{\mu})$ . To tackle this, we use the POD-DEIM approach.



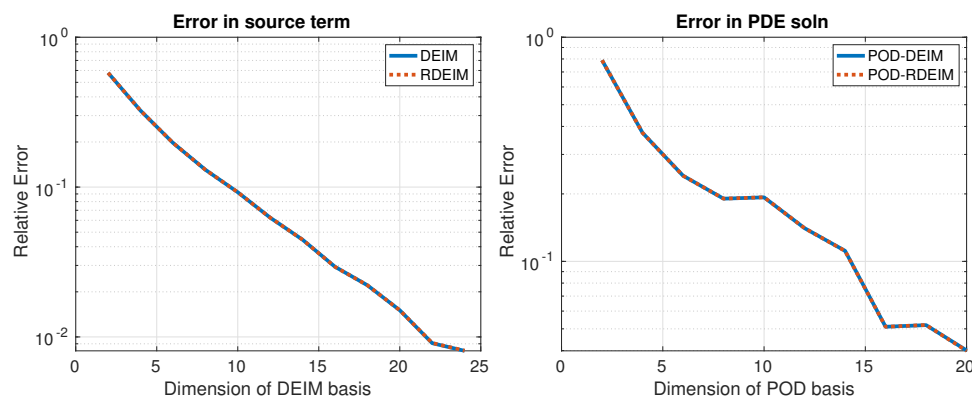


FIG. 7. (Left) The error in the DEIM and R-DEIM approximation for the source term  $s(\mathbf{x}; \boldsymbol{\mu})$ . The R-DEIM basis was constructed using Algorithm 1; the oversampling parameter was  $p = 20$ . We used the hybrid algorithm for the point selection. (Right) The error in the DEIM and R-DEIM approximation for the PDE.

Our first experiment is similar to that in [25, section 10.5.1]. We first consider the cost of approximating the source term  $s(\mathbf{x}; \boldsymbol{\mu})$  over the range of parameters  $\mu_3 \in [0.2, 0.8]$ ,  $\mu_4 \in [0.15, 0.35]$ , and  $\mu_5$  is chosen between  $[0.1, 0.35]$ . A training set for  $\boldsymbol{\mu}$  is generated by Latin hypercube sampling with  $n_s = 1000$  training points. Two different approximations are generated using DEIM and R-DEIM. The number of DEIM basis vectors used was  $r = 24$  and was determined based on the singular value decay of the snapshot matrix. The R-DEIM basis is also fixed to be of size  $r = 24$  and was computed using Algorithm 1 with an oversampling parameter  $p = 20$ . The error is computed by averaging over 200 different randomly generated test points. The results are displayed in Figure 7. The hybrid point selection algorithm is used for both the standard DEIM basis and the randomized DEIM basis—the parameters used are the same as those in subsection 6.2. As can be seen, the error of the two different methods is comparable. For this application, the number of quadrature nodes is 165,888. It is worth mentioning that the CPU time for computing the compact SVD is 7.67 seconds, whereas the CPU time for the randomized range finder is 0.25 seconds. The computational time for the point selection was comparatively 0.06 seconds. The speedup is more impressive for larger problems; see [2] for more detailed comparisons on large-scale problems.

Our second experiment is similar to [25, section 10.5.2], in which the parameters  $\mu_1 = 0.03$  and  $\mu_5 = 0.25$  are fixed and the remaining parameters are taken to be in the range  $\mu_2 \in [0, 2\pi]$ ,  $\mu_3 \in [0.2, 0.8]$ , and  $\mu_4 \in [0.15, 0.35]$ . The goal is then to compute a ROM for solving the PDE (6.1) for the above chosen range of parameters. The source term is approximated using the DEIM and the R-DEIM approaches, as described in the previous experiment (the dimension  $r = 24$  was used for both approximations). The POD algorithm is computed using 1000 snapshots, and the POD dimension  $k = 20$  is used to construct the reduced basis space. The error is shown in the right panel of Figure 7, where the POD-R-DEIM approximation is compared against the POD-DEIM approximation. It is seen that the errors are comparable, which validates our approach.

**7. Conclusions.** We have provided randomized algorithms for tackling the two main bottlenecks of the standard DEIM algorithm. First, we proposed several ran-

domized algorithms for approximately computing the DEIM basis and highlighted various benefits of the randomized algorithms including adaptivity. Second, we proposed two randomized point selection algorithms; one is based on leverage score sampling, and the other combines leverage score and rank-revealing factorizations. We provided detailed analysis of the error in the resulting algorithms that clearly shows the trade-off between computational cost and accuracy. The proposed algorithms are more efficient than the standard techniques and have comparable accuracy. Numerical experiments in section 6 confirm these findings and give insight into the choice of parameters.

**Acknowledgments.** The author would like to thank Ilse C. F. Ipsen and Zlatko Drmač for discussions and Ivy Huang for her help with this paper.

## REFERENCES

- [1] S. S. ASLAN, E. DE STURLER, AND M. E. KILMER, *Randomized approach to nonlinear inversion combining random and optimized simultaneous sources and detectors*, SIAM J. Sci. Comput., 41 (2019), pp. B229–B249, <https://doi.org/10.1137/17M1143319>, 2017.
- [2] C. BACH, F. DUDDECK, AND L. SONG, *Fixed-precision randomized low-rank approximation methods for nonlinear model order reduction of large systems*, Internat. J. Numer. Methods Engrg., 119 (2019), pp. 687–711.
- [3] P. BENNER, S. GUGERCIN, AND K. WILLCOX, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev., 57 (2015), pp. 483–531, <https://doi.org/10.1137/130932715>.
- [4] D. A. BISTRAN AND I. M. NAVON, *Randomized dynamic mode decomposition for nonintrusive reduced order modelling*, Internat. J. Numer. Methods Engrg., 112 (2017), pp. 3–25.
- [5] C. BOUTSIDIS, M. W. MAHONEY, AND P. DRINEAS, *An improved approximation algorithm for the column subset selection problem*, in Proceedings of the 2009 Annual ACM–SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2009, pp. 968–977, <https://doi.org/10.1137/1.9781611973068.105>.
- [6] M. E. BROADBENT, M. BROWN, K. PENNER, I. C. IPSEN, AND R. REHMAN, *Subset selection algorithms: Randomized vs. deterministic*, SIAM Undergrad. Res. Online, 3 (2010), pp. 50–71, <https://doi.org/10.1137/09S010435>.
- [7] S. CHATURANTABUT AND D. C. SORESENSEN, *Nonlinear model reduction via discrete empirical interpolation*, SIAM J. Sci. Comput., 32 (2010), pp. 2737–2764, <https://doi.org/10.1137/090766498>.
- [8] P. DRINEAS AND M. W. MAHONEY, *RandNLA: Randomized numerical linear algebra*, Commun. ACM, 59 (2016), pp. 80–90.
- [9] Z. DRMAČ AND S. GUGERCIN, *A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions*, SIAM J. Sci. Comput., 38 (2016), pp. A631–A648, <https://doi.org/10.1137/15M1019271>.
- [10] Z. DRMAČ AND A. K. SAIBABA, *The discrete empirical interpolation method: Canonical structure and formulation in weighted inner product spaces*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1152–1180, <https://doi.org/10.1137/17M1129635>.
- [11] N. B. ERICHSON, L. MATHELIN, S. L. BRUNTON, AND J. N. KUTZ, *Randomized Dynamic Mode Decomposition*, preprint, <https://arxiv.org/abs/1702.02912>, 2017.
- [12] N. B. ERICHSON AND C. DONOVAN, *Randomized low-rank dynamic mode decomposition for motion detection*, Comput. Vis. Image Understand., 146 (2016), pp. 40–50.
- [13] C. GORMAN, G. CHÁVEZ, P. GHYSELS, T. MARY, F.-H. ROUET, AND X. S. LI, *Matrix-Free Construction of HSS Representation Using Adaptive Randomized Sampling*, preprint, <https://arxiv.org/abs/1810.04125>, 2018.
- [14] M. GU, *Subspace iteration randomization and singular value problems*, SIAM J. Sci. Comput., 37 (2015), pp. A1139–A1173, <https://doi.org/10.1137/130938700>.
- [15] M. GU AND S. C. EISENSTAT, *Efficient algorithms for computing a strong rank-revealing QR factorization*, SIAM J. Sci. Comput., 17 (1996), pp. 848–869, <https://doi.org/10.1137/0917055>.
- [16] N. HALKO, P. G. MARTINSSON, AND J. A. TROPP, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288, <https://doi.org/10.1137/090771806>.

- [17] J. S. HESTHAVEN, G. ROZZA, AND B. STAMM, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, Springer, New York, 2016.
- [18] J. T. HOLODNAK AND I. C. F. IPSEN, *Randomized approximation of the Gram matrix: Exact computation and probabilistic bounds*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 110–137, <https://doi.org/10.1137/130940116>.
- [19] I. C. F. IPSEN AND T. WENTWORTH, *The effect of coherence on sampling from matrices with orthonormal columns, and preconditioned least squares problems*, SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1490–1520, <https://doi.org/10.1137/120870748>.
- [20] M. W. MAHONEY, *Randomized algorithms for matrices and data*, Found. Trends Mach. Learn., 3 (2011), pp. 123–224.
- [21] P.-G. MARTINSSON AND S. VORONIN, *A randomized blocked algorithm for efficiently computing rank-revealing factorizations of matrices*, SIAM J. Sci. Comput., 38 (2016), pp. S485–S507, <https://doi.org/10.1137/15M1026080>.
- [22] B. PEHERSTORFER, D. BUTNARU, K. WILLCOX, AND H.-J. BUNGARTZ, *Localized discrete empirical interpolation method*, SIAM J. Sci. Comput., 36 (2014), pp. A168–A192, <https://doi.org/10.1137/130924408>.
- [23] B. PEHERSTORFER, Z. DRMAČ, AND S. GUGERCIN, *Stabilizing Discrete Empirical Interpolation via Randomized and Deterministic Oversampling*, preprint, <https://arxiv.org/abs/1808.10473>, 2018.
- [24] B. PEHERSTORFER AND K. WILLCOX, *Online adaptive model reduction for nonlinear systems via low-rank updates*, SIAM J. Sci. Comput., 37 (2015), pp. A2123–A2150, <https://doi.org/10.1137/140989169>.
- [25] A. QUARTERONI, A. MANZONI, AND F. NEGRI, *Reduced Basis Methods for Partial Differential Equations: An Introduction*, La Matematica per il 92, Springer, Cham, 2015.
- [26] Y. SAAD, *Numerical Methods for Large Eigenvalue Problems: Revised Edition*, Classics Appl. Math. 66, SIAM, Philadelphia, 2011, <https://doi.org/10.1137/1.9781611970739>.
- [27] A. K. SAIBABA, *Randomized subspace iteration: Analysis of canonical angles and unitarily invariant norms*, SIAM J. Matrix Anal. Appl., 40 (2019), pp. 23–48, <https://doi.org/10.1137/18M1179432>.
- [28] D. C. SORESENSEN AND M. EMBREE, *A DEIM induced CUR factorization*, SIAM J. Sci. Comput., 38 (2016), pp. A1454–A1482, <https://doi.org/10.1137/140978430>.
- [29] G. W. STEWART AND J. G. SUN, *Matrix Perturbation Theory*, Computer Science and Scientific Computing, Academic Press, Boston, MA, 1990.
- [30] D. B. SZYLD, *The many proofs of an identity on the norm of oblique projections*, Numer. Algorithms, 42 (2006), pp. 309–323.
- [31] J. A. TROPP, A. YURTSEVER, M. UDELL, AND V. CEVHER, *Practical sketching algorithms for low-rank matrix approximation*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1454–1485, <https://doi.org/10.1137/17M1111590>.
- [32] P. Å. WEDIN, *On angles between subspaces of a finite dimensional inner product space*, in Matrix Pencils, Lecture Notes in Math. 973, Springer, Berlin, 1983, pp. 263–285.
- [33] D. YU AND S. CHAKRAVORTY, *A randomized proper orthogonal decomposition technique*, in Proceedings of the 25th American Control Conference (ACC), IEEE, Washington, DC, 2015, pp. 1137–1142.
- [34] D. YU AND S. CHAKRAVORTY, *A computationally optimal randomized proper orthogonal decomposition technique*, in Proceedings of the 26th American Control Conference (ACC), IEEE, Washington, DC, 2016, pp. 3310–3315.
- [35] W. YU, Y. GU, AND Y. LI, *Efficient randomized algorithms for the fixed-precision low-rank matrix approximation*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1339–1359, <https://doi.org/10.1137/17M1141977>.