

AN ACCELERATED GREEDY MISSING POINT ESTIMATION
PROCEDURE*R. ZIMMERMANN[†] AND K. WILLCOX[‡]

Abstract. Model reduction via Galerkin projection fails to provide considerable computational savings if applied to general nonlinear systems. This is because the reduced representation of the state vector appears as an argument to the nonlinear function, whose evaluation remains as costly as for the full model. Masked projection approaches, such as the missing point estimation and the (discrete) empirical interpolation method, alleviate this effect by evaluating only a small subset of the components of a given nonlinear term; however, the selection of the evaluated components is a combinatorial problem and is computationally intractable even for systems of small size. This has been addressed through greedy point selection algorithms, which minimize an error indicator by sequentially looping over all components. While doable, this is suboptimal and still costly. This paper introduces an approach to accelerate and improve the greedy search. The method is based on the observation that the greedy algorithm requires solving a sequence of symmetric rank-one modifications to an eigenvalue problem. For doing so, we develop fast approximations that sort the set of candidate vectors that induce the rank-one modifications, without requiring solution of the modified eigenvalue problem. Based on theoretical insights into symmetric rank-one eigenvalue modifications, we derive a variation of the greedy method that is faster than the standard approach and yields better results for the cases studied. The proposed approach is illustrated by numerical experiments, where we observe a speed-up by two orders of magnitude when compared to the standard greedy method while arriving at a better quality reduced model.

Key words. model reduction, dimensionality reduction, rank-one modifications of the symmetric eigenproblem, missing point estimation, discrete empirical interpolation, DEIM, gappy proper orthogonal decomposition, POD, masked projection

AMS subject classifications. 65F30, 65F20, 37M99

DOI. 10.1137/15M1042899

1. Introduction. Consider a general spatio-temporal dynamical system in semi-discretized form,

$$(1.1) \quad \frac{\partial}{\partial t} x(t, \mu) = f(x(t, \mu)), \quad x(t_0, \mu) = x_{0, \mu},$$

where $x(t, \mu) \in \mathbb{R}^n$ is the spatially discretized *state vector* of dimension n , μ denotes additional system parameters, and $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ may be nonlinear. The goal of model reduction is to replace the system (1.1) with a system with many fewer degrees of freedom $p \ll n$. Projection-based model reduction methods approach this goal by projecting (1.1) onto a p -dimensional subspace \mathcal{U} . This leads to a reduced system that can be solved efficiently if the operator f can be represented by an affine combination of linear operators; see, e.g., [3, sect. 2.3]. Otherwise, evaluating the right-hand

*Submitted to the journal's Methods and Algorithms for Scientific Computing section October 6, 2015; accepted for publication (in revised form) June 15, 2016; published electronically September 7, 2016.

<http://www.siam.org/journals/sisc/38-5/M104289.html>

Funding: The first author was supported by a fellowship awarded by the German Research Foundation (DFG), grant Zi 1250/2-1. The second author was supported by the U.S. Department of Energy, awards DE-FG02-08ER2585 and DE-SC0009297, as part of the DiaMonD Multifaceted Mathematics Integrated Capability Center.

[†]Institute Computational Mathematics, TU Braunschweig, 38106, Germany, and Department of Aeronautics and Astronautics, MIT, Cambridge, MA 02139. Current address: Department of Mathematics and Computer Science, SDU Odense, 5320, Denmark (zimmermann@imada.sdu.dk).

[‡]Department of Aeronautics and Astronautics, MIT, Cambridge, MA 02139 (kwillcox@mit.edu).

side of the projected system requires a similar computational effort as for the original system. Nonlinear model reduction methods, such as missing point estimation (MPE) [1], masked projection [8], and the discrete empirical interpolation method (DEIM) [6] (based on the empirical interpolation method [2]) tackle this problem by employing a so-called mask matrix that reduces the number of components of f that enter the reduced model to a small fraction of the original model. Essentially, the same mathematical principles apply to sensor placement via the method of gappy proper orthogonal decomposition (POD) [18, 14], though in a slightly different context.

The definition of a mask matrix is as follows. Let $e_i \in \mathbb{R}^n$ denote the i th canonical unit vector, $i = 1, \dots, n$. Given a subset of indices $J = \{j_1, \dots, j_s\} \subset \{1, \dots, n\}$, the (column-orthogonal) matrix $P = (e_{j_1}, \dots, e_{j_s}) \in \mathbb{R}^{n \times s}$ is called *the mask matrix corresponding to the index set J* . Left-multiplication of a vector with the transpose of P realizes the projection onto the components in the same order as listed in J , i.e., $P^T y = (y_{j_1}, \dots, y_{j_s})^T \in \mathbb{R}^s \forall y \in \mathbb{R}^n$. For a rank- p matrix $U \in \mathbb{R}^{n \times p}$, we will call a mask matrix $P \in \mathbb{R}^{n \times s}$ *overdetermining with respect to U* if $s > p$ and $\text{rank}(P^T U) = p$. It will be called *uniquely determining with respect to U* if $s = p$ and $\text{rank}(P^T U) = p$. This reflects the fact that the *masked least-squares system*

$$\min_{\alpha \in \mathbb{R}^p} \|P^T U \alpha - P^T b\|, \quad U \in \mathbb{R}^{n \times p}, b \in \mathbb{R}^n,$$

is overdetermined if $s > p$ and uniquely determined if $s = p$. In the applications, the indices collected in J usually correspond to points in a discretized spatial domain and are therefore also referred to as *point indices* or sometimes simply as *points*. In [6], they are called *interpolation points*, since only uniquely determining mask matrices are considered there. We did not adopt this term, because our focus is on overdetermining mask matrices, which are associated not with data interpolation but with data regression.

In nonlinear model reduction, selecting the optimal set of point indices associated with a mask matrix (i.e., the optimal subset of components of f that are to be evaluated) is a combinatorial problem that is prohibitively costly to solve exactly. Therefore, [1, sect. V.A] introduces a greedy point selection algorithm, which minimizes an error indicator by sequentially looping over the full grid. The associated costs of this greedy algorithm are much lower than for the combinatorial problem but are still considerably high.

DEIM also comes with an algorithm for choosing the underlying point index set that is based on a similar error indicator. In the DEIM algorithm [6, Alg. 1], the number of selected points is directly linked to the dimension of the subspace \mathcal{U} . This means that it is not possible to select more point indices than there are basis vectors spanning \mathcal{U} . This is a limiting factor, since there are application scenarios where oversampling is beneficial (see [1]) or even explicitly required (see [13]).

In this work, we consider the problem of efficiently constructing mask matrices that are based on point index sets of cardinality larger than the number of basis vectors. To this end, we combine a generalized version of the error bound of [6, Lem. 3.2] with an accelerated version of the greedy point index search of [1, sect. V.A]. We show that the greedy point selection boils down to selecting the one vector, out of a finite set of candidates, that leads to the largest growth in the smallest eigenvalue of a modified eigenvalue problem. In order to find this vector, we develop approximations that sort the set of candidate vectors according to certain properties of their components without the requirement to actually solve the modified eigenvalue problem.

Organization. Section 2 gives the required background on greedy MPE. Section 3 features known and new theoretical insights on symmetric rank-one eigenvalue modifications. In section 4, we introduce the accelerated greedy surrogate and propose variations for improving its performance. An engineering application is discussed in section 5. Section 6 summarizes the findings.

2. Problem statement. In this section, the basic principle of masked projection schemes is reviewed. We then introduce a minor generalization to the standard DEIM error bound [6, Lem. 3.2] and state a new analogue for masked least-squares residuals. Finally, the greedy algorithm of [1, sect. V.A] is restated. This algorithm will act as a benchmark reference for the new method developed in section 3.

2.1. Masked projection. Let $y \in \mathbb{R}^n$ denote an arbitrary vector and let $U \in \mathbb{R}^{n \times p}$ be a column-orthogonal matrix. We may think of y as a state solution or other vector of interest computed using a given simulation code and of the columns of U as a basis of a p -dimensional subspace $\mathcal{U} \subset \mathbb{R}^n$. In the introductory example (1.1), y would be the vector obtained from the nonlinear right-hand side evaluated at a certain parameter condition $y = f(x(t_i, \mu_i)) \in \mathbb{R}^n$. Let $J = \{j_1, \dots, j_s\} \subset \{1, \dots, n\}$ be a subset of indices with corresponding mask matrix $P = (e_{j_1}, \dots, e_{j_s}) \in \mathbb{R}^{n \times s}$. Throughout, we assume that we have at least as many indices to start with as there are basis vectors in U , i.e., $|J| = s \geq p$. This initial set of indices is needed so that the underlying masked least-squares problem is not rank-deficient and consequently, the masked projector is well-defined. In practice, it could be obtained, for example, by applying the standard DEIM point selection algorithm [6, Alg. 1] or by the recent variation [7].

The best approximation to y contained in \mathcal{U} is the orthogonal projection $\tilde{y} = UU^T y$, which corresponds to the solution to the least-squares problem $\min_{\alpha \in \mathbb{R}^p} \|y - U\alpha\|^2$ with the optimal argument being $\alpha^* = U^T y$. The best approximation to y contained in \mathcal{U} that is solely determined on the information at the point indices associated with the mask matrix P is $\hat{y} = U(U^T P P^T U)^{-1} U^T P P^T y$, which corresponds to the solution to the masked least-squares problem $\min_{\alpha \in \mathbb{R}^p} \|P^T y - P^T U \alpha\|^2$. Introducing the oblique projector

$$(2.1) \quad \Pi : y \mapsto \hat{y} = U(U^T P P^T U)^{-1} U^T P P^T y,$$

this solution can be expressed in concise form as $\hat{y} = \Pi(y)$. We will call Π the *masked projection* onto $\text{colspan}(U)$. It is well-defined whenever $P^T U$ features the (maximum possible) rank p .

2.2. Error bounds for masked projection schemes. The next proposition gives an estimate on the error introduced by the masked projection (2.1). It is a minor generalization of [6, Lem. 3.2] to the case of overdetermining mask matrices.

PROPOSITION 2.1. *Let $y \in \mathbb{R}^n$. The error between the orthogonal projection $\tilde{y} = UU^T y$ of y onto $\text{colspan}(U)$ and the masked projection $\hat{y} = \Pi(y)$ of y onto $\text{colspan}(U)$ is bounded by*

$$\|\tilde{y} - \hat{y}\| \leq \|\Pi\| \|y - UU^T y\|, \quad \text{where } \|\Pi\| = \frac{1}{\sigma_{\min}(P^T U)}.$$

The same bound holds for the error between y and its masked approximation,

$$(2.2) \quad \|y - \hat{y}\| \leq \|\Pi\| \|y - UU^T y\|.$$

Proof. The proof of [6, Lem. 3.2] transfers to this setting. Hence, we are only left with computing the norm of the oblique projector Π associated with an overdetermining mask matrix. For doing so, let $WSR^T = P^T U$ be the thin SVD with $W \in \mathbb{R}^{s \times p}$, $R \in \mathbb{R}^{p \times p}$ orthogonal. Then

$$\begin{aligned} \|\Pi\| &= \|U(U^T P P^T U)^{-1} U^T P P^T\| = \|URS^{-2} R^T R S W^T P^T\| = \|URS^{-1} W^T P^T\| \\ &= \sqrt{\lambda_{\max}(RS^{-1} W^T P^T P W S^{-1} R^T)} = \sqrt{\lambda_{\max}(S^{-2})} = 1/\sigma_{\min}(P^T U). \end{aligned}$$

Note that this is not an estimate but the exact value of the 2-norm of the projector Π . \square

Remark 1.

1. Since $\sigma_{\min}(P^T U) \leq \sigma_{\max}(P^T U) = \|P^T U\| \leq \|P\| \|U^T\| = 1$, the best possible value for the bound $\|\Pi\|$ is $1/\sigma_{\min}(P^T U) = 1$ and is, e.g., achieved for $P = I_{n \times n}$. Because $1/\sigma_{\min}(P^T U) \leq 1/\sigma_{\min}(P^T U)^2 = \|(U^T P P^T U)^{-1}\|$, a less sharp but maybe more familiar estimate for the above expression is $\|\tilde{y} - \hat{y}\| \leq \|(U^T P P^T U)^{-1}\| \|y - U U^T y\|$.
2. A comparable result is [1, Thm. 5], which arrives at $\|(U^T P P^T U)^{-1} - I\|$ as the key quantity for the so-called alias error introduced by the masked projection. The main difference between this result and the approaches pursued here (and in [6]) is that the derivation in [1] is with respect to a different non-Euclidean inner product. Note, however, that the authors of [1] eventually promote use of the condition number $\text{cond}(U^T P P^T U) = \|U^T P P^T U\| \cdot \|(U^T P P^T U)^{-1}\|$ as the objective function for a greedy point-selection strategy, which is related to the estimate given by Proposition 2.1. The method proposed in this work transfers to the condition number objective.

Proposition 2.1 bounds the error between the orthogonal projection of a vector $y \in \mathbb{R}^n$ onto a subspace \mathcal{U} and the oblique masked projection $\hat{y} = \Pi y$ onto \mathcal{U} . We now state an analogous error bound for a solution to a least-squares system and its masked counterpart, based on the associated residuals. Consider the problem $\min_{\alpha \in \mathbb{R}^p} \|AU\alpha - b\|^2$ with $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. The solution is $U\tilde{\alpha}$ with

$$(2.3) \quad \tilde{\alpha} = \arg \min_{\alpha \in \mathbb{R}^p} \|AU\alpha - b\|^2 = (U^T A^T AU)^{-1} U^T A^T b.$$

Given an index subset $J := \{j_1, \dots, j_s\} \subset \{1, \dots, m\}$ with corresponding mask matrix $P = (e_{j_1}, \dots, e_{j_s}) \in \mathbb{R}^{m \times s}$, the masked least-squares problem induced by P is

$$(2.4) \quad \hat{\alpha} = \arg \min_{\alpha \in \mathbb{R}^p} \|P^T AU\alpha - P^T b\|^2 = (U^T A^T P P^T AU)^{-1} U^T A^T P P^T b.$$

The next proposition bounds the associated errors.

PROPOSITION 2.2. *Let $\tilde{\alpha} \in \mathbb{R}^p$ be the solution to $\min_{\alpha \in \mathbb{R}^p} \|AU\alpha - b\|^2$ and let $\hat{\alpha} \in \mathbb{R}^p$ be the solution to its masked counterpart $\min_{\alpha \in \mathbb{R}^p} \|P^T AU\alpha - P^T b\|^2$. Moreover, let $QSV^T \stackrel{\text{SVD}}{=} AU$ be the thin SVD of AU . The error between $U\tilde{\alpha}$ and $U\hat{\alpha}$ is bounded by*

$$\|U\tilde{\alpha} - U\hat{\alpha}\| \leq \frac{1}{\sigma_{\min}(AU)\sigma_{\min}(P^T Q)} \|res(U\tilde{\alpha})\|,$$

where $\|res(U\tilde{\alpha})\| = \|AU\tilde{\alpha} - b\|$ is the value of the residual of the least-squares-optimal solution in $\text{colspan}(U)$. In terms of the residuals, the full and the masked least-squares solution exhibit an error of

$$\|res(U\tilde{\alpha}) - res(U\hat{\alpha})\| \leq \frac{1}{\sigma_{\min}(P^T Q)} \|res(U\tilde{\alpha})\|.$$

Proof. Let $\tilde{\alpha}, \hat{\alpha}$ be as introduced in (2.3) and (2.4), respectively. Using $QSV^T \stackrel{\text{SVD}}{=} AU$, we obtain

$$\tilde{\alpha} = VS^{-1}Q^Tb, \quad \hat{\alpha} = VS^{-1}(Q^T P P^T Q)^{-1}Q^T P P^T b.$$

Hence,

$$\begin{aligned} \|U\tilde{\alpha} - U\hat{\alpha}\| &= \|\tilde{\alpha} - \hat{\alpha}\| = \|VS^{-1}((Q^T P P^T Q)^{-1}Q^T P P^T b - Q^T b)\| \\ &\leq \|S^{-1}\| \cdot \|(Q^T P P^T Q)^{-1}Q^T P P^T (b - Q Q^T b)\| \\ &\leq \|S^{-1}\| \cdot \|(Q^T P P^T Q)^{-1}Q^T P P^T\| \cdot \|res(U\tilde{\alpha})\|, \end{aligned}$$

since $res(U\tilde{\alpha}) = \|AU\tilde{\alpha} - b\| = \|Q Q^T b - b\|$. The norm of the oblique projector $(Q^T P P^T Q)^{-1}Q^T P P^T$ can be computed as in Proposition 2.1. For the error bound on the residuals, the factor $\|S^{-1}\| = \sigma_{\min}(AU)$ drops out of the inequalities, because $AU\tilde{\alpha} = Q Q^T b$, $AU\hat{\alpha} = Q(Q^T P P^T Q)^{-1}Q^T P P^T b$. \square

2.3. Greedy point index selection strategies. The basic objective in masked projection schemes is to minimize the distance between a vector and its masked projection onto a given subspace. A bound for this distance is given by Proposition 2.1. Since the subspace is usually to be considered as fixed in the applications, the only way to tune the bound is via the selection of the index set that defines the associated mask matrix. This leads to the following combinatorial point selection strategy:

$$\text{PSS1: } \min_{J \subset \{1, \dots, n\}, |J|=s} \left\{ \frac{1}{\sigma_{\min}(P^T U)} \mid P = (e_{j_1}, \dots, e_{j_s}) \right\}.$$

In an analogous way, the residual-based error bound of Proposition 2.2 leads to the following combinatorial point selection strategy:

$$\text{PSS2: } \min_{J \subset \{1, \dots, m\}, |J|=s} \left\{ \frac{1}{\sigma_{\min}(P^T Q)} \mid P = (e_{j_1}, \dots, e_{j_s}) \right\},$$

where $QSV^T \stackrel{\text{SVD}}{=} AU$ and $A \in \mathbb{R}^{m \times n}$, $U \in \mathbb{R}^{n \times p}$ are as in Proposition 2.2. In engineering applications, the matrix A that defines the least-squares problem in Proposition 2.2 may depend on a parameter vector μ (e.g., parameters that account for varying system properties such as material constants or boundary conditions). Hence, $A = A(\mu)$ and as a consequence $Q = Q(\mu)$. But this, in turn, makes the optimal mask matrix P that is implicitly specified by PSS2 also parameter-dependent. Formally, this means that for each change in μ , the combinatorial problem PSS2 has to be solved anew, which can be a drawback. In practice, both point selection strategies may be executed via the greedy algorithm proposed in [1, sect. V.A].

A single step in the greedy optimization works as follows: Let $J_s = (j_1, \dots, j_s) \subset \{1, \dots, n\}$ be the point index set at iteration s with corresponding mask matrix $P_s = (e_{j_1}, \dots, e_{j_s})$. The next point index j_{s+1} is chosen via Algorithm 1.

The exhaustive greedy search is costly, since it loops over $n - s \approx n$ indices at each iteration and thus requires solving about n SVD problems of size $s \times p$. Assuming that the latter task is performed in $\mathcal{O}(sp^2)$, the total computational effort is of the order $\mathcal{O}(nsp^2)$ which is larger than $\mathcal{O}(np^3)$. In order to cut these costs, two different point screening criteria are proposed in [1, sect. V.B], both of them working by an a priori reduction of the full point index set to a smaller subset of candidate points that exhibit the best values according to the chosen screening criterion. After this initial reduction, the algorithm proceeds as described above.

Algorithm 1. Exhaustive greedy MPE for maximizing $\sigma_{\min}(P_{s+1}^T U)$.

Input: $U \in \mathbb{R}^{n \times p}$, $J_s = \{j_1, \dots, j_s\} \subset \{1, \dots, n\}$, $P_s = (e_{j_1}, \dots, e_{j_s}) \in \mathbb{R}^{n \times s}$, where

$s \geq p$.

1: $\sigma_{opt} = 0$, $\bar{J}_s = \{1, \dots, n\} \setminus J_s$

2: **for** $j \in \bar{J}_s$ **do**

3: $\tilde{P} = (P_s, e_j) \in \mathbb{R}^{n \times (s+1)}$

4: Compute $\sigma_j = \sigma_{\min}(\tilde{P}^T U)$

5: **if** $\sigma_j > \sigma_{opt}$ **then**

6: $\sigma_{opt} = \sigma_j$

7: $j_{opt} = j$

8: **end if**

9: **end for**

Output: $J_{s+1} = J_s \cup \{j_{opt}\}$, $P_{s+1} = [P, e_{j_{opt}}]$, $\bar{J}_{s+1} = \bar{J}_s \setminus \{j_{opt}\}$

In contrast, we propose in this work a screening criterion that determines the next point index at each step of the greedy optimization loop based on an approximation that uses information obtained at the previous iteration. Therefore, our method does not replace but may be combined with the a priori approach of [1, sect. V.B].

3. Theoretical grounds for accelerated MPE. In this section, we develop the theoretical grounds for a new point selection criterion.

3.1. Greedy point selection formulated as rank-one SVD updates. First, we observe that step 4 of Algorithm 1. in fact requires solving a series of rank-one SVD update problems. Indeed, if $\Psi_s \Sigma_s \Phi_s^T \stackrel{\text{SVD}}{=} P_s^T U$ is the thin SVD with $\Psi_s \in \mathbb{R}^{s \times p}$, $\Sigma_s = \text{diag}(\sigma_{s,1}, \dots, \sigma_{s,p})$, $\Phi_s \in \mathbb{R}^{p \times p}$, then for any index $j_{s+1} \in \{1, \dots, n\} \setminus J_s$, we have

$$(3.1) \quad \begin{aligned} P_{s+1}^T U &= \begin{pmatrix} P_s^T U \\ e_{j_{s+1}}^T U \end{pmatrix} = \begin{pmatrix} \Psi_s \Sigma_s \Phi_s^T \\ (u_{j_{s+1},1}, \dots, u_{j_{s+1},p}) \end{pmatrix} \\ &= \begin{pmatrix} \Psi_s & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \Sigma_s \\ u_{[j_{s+1},:]} \Phi_s \end{pmatrix} \Phi_s^T, \end{aligned}$$

where we have used the MATLAB-inspired notation $u_{[j_{s+1},:]}$ to denote the j_{s+1} th row of U . The task is therefore to recompute the singular values after adding a row to the previous-step singular value matrix Σ_s . This corresponds to the following symmetric rank-one eigenvalue modification:

$$(3.2) \quad U^T P_{s+1} P_{s+1}^T U = \Phi_s (\Sigma_s^2 + v v^T) \Phi_s^T, \text{ where } v = \Phi_s^T u_{[j_{s+1},]}^T.$$

The candidate vectors v that arise during the greedy point selection procedure at iteration s are precisely the rows of the matrix $(U \Phi_s)$ corresponding to the point indices in $\bar{J}_s := \{1, \dots, n\} \setminus J_s$. Hence, we may recast the greedy point selection objective in the following way:

At iteration s , out of the row vectors v^j , $j \in \bar{J}_s := \{1, \dots, n\} \setminus J_s$ of $U \Phi_s$, determine the one that leads to the largest growth in the smallest eigenvalue of

$$(3.3) \quad M := \Sigma_s^2 + v^j (v^j)^T \in \mathbb{R}^{p \times p}.$$

Much is known about symmetric rank-one modifications of the form (3.3); see [17, 9, 15, 4, 5, 10]. Of particular importance for our purposes is the following theorem, which follows from the considerations [17, sect. 38–41] and can also be found in [5, Thm. 1].

THEOREM 3.1 (Wilkinson [17]). *Let $D \in \mathbb{R}^{p \times p}$ be diagonal and let $v \in \mathbb{R}^p$. Suppose that the eigenvalues of D are in descending order $d_1 \geq \dots \geq d_p$. Let $\lambda_1 \geq \dots \geq \lambda_p$ be the eigenvalues of $M := D + vv^T$ and let $\rho := \|v\|^2$. Then*

$$\lambda_i = d_i + \rho\mu_i, \text{ where } 0 \leq \mu_i \leq 1 \text{ and } \sum_{i=1}^p \mu_i = 1.$$

Moreover, $\lambda_1 \geq d_1 \geq \lambda_2 \geq \dots \geq d_{n-1} \geq \lambda_n \geq d_n$.

If the d_i are distinct and all entries of v are nonzero, the above inequalities hold in the strict sense.

By computing the characteristic polynomial of $M = D + vv^T$ (see [17, eq. 39.7], [9, sect. 5]), one can show that the shifted eigenvalues λ_i of M are the zeros of

$$(3.4) \quad f(\lambda) = 1 + \sum_{i=1}^p \frac{v_i^2}{d_i - \lambda}.$$

Given the modified λ_i , the associated eigenvectors are [5, sect. 4]

$$(3.5) \quad q_i = \frac{\tilde{q}_i}{\|\tilde{q}_i\|}, \text{ where } \tilde{q}_i = \left(\frac{v_1}{d_1 - \lambda_i}, \dots, \frac{v_p}{d_p - \lambda_i} \right)^T.$$

3.2. On the growth behavior of the rank-one modified eigenvalues. At iteration s of the greedy point selection Algorithm 1., our goal is to select the next point index j_{opt} by analyzing properties of the candidate row vectors v^j of $U\Phi_s$ that induce the rank-one modifications (3.2), rather than actually solving the modified eigenvalue/SVD problem. To this end, we investigate in this subsection the growth behavior of the modified eigenvalues depending on the components of the rank-one update vector.

LEMMA 3.2. *Let $D \in \mathbb{R}^{p \times p}$ be diagonal, strictly positive definite with mutually distinct eigenvalues. For $v \in \mathbb{R}^p$, let $M = D + vv^T$ and let $\lambda_k, k = 1, \dots, p$, be the eigenvalues of M . Then the partial derivative of λ_k by v_i is*

$$(3.6) \quad \partial_i \lambda_k(v) = \frac{-2}{\|\tilde{q}_k\|^2} \frac{v_i}{d_i - \lambda_k(v)},$$

where \tilde{q}_k is the k th (nonnormalized) eigenvector of M according to (3.5).

Proof. We will make use of the implicit function theorem: Consider the characteristic function f from (3.4) as a function depending both on $v \in \mathbb{R}^p$ and $\lambda \in \mathbb{R}$, i.e., consider $\tilde{f} : \mathbb{R}^p \times \mathbb{R} \rightarrow \mathbb{R}$, $(v, \lambda) \mapsto 1 + \sum_{i=1}^p \frac{v_i^2}{d_i - \lambda}$. Let $v_0 \in \mathbb{R}^p$ be arbitrary and let $\lambda_{k,0}$ be the k th eigenvalue of the rank-one modification $D + v_0 v_0^T$, so that $\tilde{f}(v_0, \lambda_{k,0}) = 0$. Since $\partial_\lambda \tilde{f}(v_0, \lambda_{k,0}) = \sum_{i=1}^p \frac{v_{0,i}^2}{(d_i - \lambda_{k,0})^2} > 0$, the implicit function theorem applies and gives rise to a function $\lambda_k : v \mapsto \lambda_k(v)$ defined in a small neighborhood of $(v_0, \lambda_{k,0})$ such that $\tilde{f}(v, \lambda_k(v)) = 0$. Hence, $\lambda_k(v)$ parametrizes the k th eigenvalue depending on v . Implicit differentiation gives

$$\begin{aligned} \mathcal{D}_v \lambda_k(v) &= \frac{-1}{\partial_{\lambda} \tilde{f}(v, \lambda_k(v))} \mathcal{D}_v \tilde{f}(v, \lambda_k(v)) \\ &= \frac{-2}{\sum_{i=1}^p \frac{v_i^2}{(d_i - \lambda_k(v))^2}} \left(\frac{v_1}{d_1 - \lambda_k(v)}, \dots, \frac{v_p}{d_p - \lambda_k(v)} \right). \end{aligned}$$

Since differentiation is a local concept, the above argument gives the derivative of the k th eigenvalue $\lambda_k(v)$ for any $v \in \mathbb{R}^p \neq 0$. After observing that the denominator of the prefactor is precisely the squared norm of the modified eigenvector \tilde{q}_k as stated in (3.5), the claim follows. \square

Lemma 3.2 allows us to determine the growth characteristics of the smallest eigenvalue $\lambda_p(v)$ of the modified matrix $M = D + vv^T$ depending on the components of v .

THEOREM 3.3. *Let $i \in \{1, \dots, p-1\}$. Consider the behavior of λ_p along the i th coordinate, $v_i \mapsto \lambda_p((v_1, \dots, v_i, \dots, v_p))$. This function is*

$$\begin{cases} \text{strictly monotonically decreasing for } v_i > 0, \\ \text{strictly monotonically increasing for } v_i < 0. \end{cases}$$

For the last component $i = p$, the growth behavior is reversed: The function $v_p \mapsto \lambda_p((v_1, \dots, v_{p-1}, v_p))$ is

$$\begin{cases} \text{strictly monotonically increasing for } v_p > 0, \\ \text{strictly monotonically decreasing for } v_p < 0. \end{cases}$$

The smallest eigenvalue $\lambda_p(v)$ features local extrema at $v_i = 0$ along all of the above coordinate-line functions. The point $v = 0$ is a saddle point of $v \mapsto \lambda_p(v)$.

Proof. The growth behavior of $\lambda_p(v)$ along the coordinates is determined by the signs of the corresponding partial derivatives. The signs are readily determined by combining Lemma 3.2 and Theorem 3.1. \square

The statement of Theorem 3.3 is illustrated by figure 1, where we consider a positive definite ($p = 10$)-dimensional random diagonal matrix $D = \text{diag}(d_1, \dots, d_p)$ and a random vector $v \in \mathbb{R}^p$, with $0 < |v_i| < 1 \forall i = 1, \dots, p$; see Appendix A.

Given a fixed diagonal matrix D and a finite set of candidate vectors v^j , $j \in \{1, \dots, \tilde{n}\}$, Theorem 3.3 gives us important information on how to select a subset of vectors $\{v^{j_{opt}}, j_{opt} \in J_{opt}\}$ that is most promising in inducing a large growth in the smallest eigenvalue of the rank-one modification $M = D + v^j (v^j)^T$:

Select the vectors $v^{j_{opt}}$ that feature the largest absolute values in the ultimate component while all other components are comparably small.

A more precise analysis of the growth behavior of the smallest modified eigenvalue λ_p is given in the next lemma.

LEMMA 3.4. *Let $v \in \mathbb{R}^p$ with $v_i \neq 0 \forall i$ and let D be a positive definite diagonal matrix with diagonal entries in descending order $d_1 > \dots > d_p > 0$. Let $c_{up}(v) = 1 + \sum_{l=1}^{p-2} \frac{v_l^2}{d_l - d_{p-1}}$ and $c_{lo}(v) = 1 + \sum_{l=1}^{p-2} \frac{v_l^2}{d_l - d_p}$. Then*

$$\begin{aligned} (3.7) \quad & 0 < d_{p-1} - \lambda_p < d_{p-1} - \lambda_{p,0}(c_{up}) \\ & = \frac{\gamma_p}{2} - \frac{1}{2c_{up}} \left(v_{p-1}^2 + v_p^2 - \sqrt{(\beta_p - v_p^2)^2 + (\beta_p + v_{p-1}^2)^2 + 2v_{p-1}^2 v_p^2 - \beta_p^2} \right), \end{aligned}$$

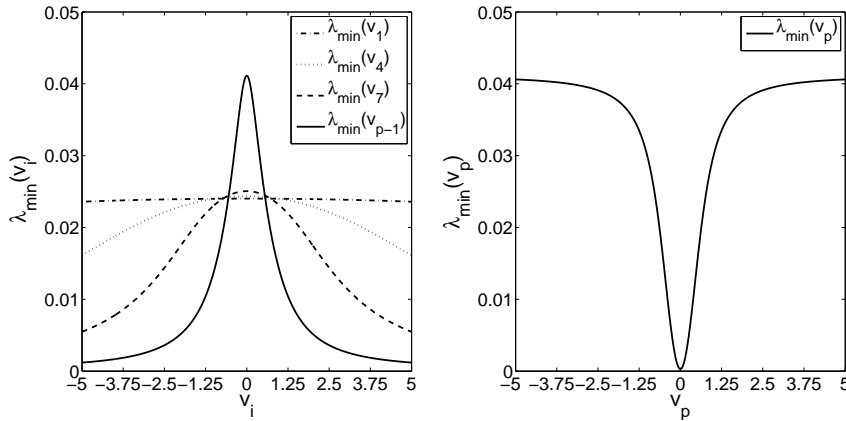


FIG. 1. Graphical illustration of Theorem 3.3. Left: The graphs of the functions $[-5, 5] \rightarrow \mathbb{R}, v_i \mapsto \lambda_p((v_1, \dots, v_i, \dots, v_p))$, $i = 1, 4, 7, p-1$ for a random vector $v \in \mathbb{R}^p$, $p = 10$, with all components $v_l, l \neq i$ kept fixed. Right: The graph of the function $[-5, 5] \rightarrow \mathbb{R}, v_p \mapsto \lambda_p((v_1, \dots, v_{p-1}, v_p))$ for the same random vector v with all components v_1, \dots, v_{p-1} kept fixed (in this case, $d_{p-1} = 4.112 \cdot 10^{-2}$, $d_p = 2.879 \cdot 10^{-4}$).

where λ_p is the smallest eigenvalue of the rank-one modification $M = D + vv^T$, $\gamma_p = d_{p-1} - d_p > 0$ is the gap between the second to last and the last eigenvalue of D , $\beta_p = c_{up}(d_{p-1} - d_p)$ and $\lambda_{p,0}(c) = \frac{\alpha_1(c)}{2} - \sqrt{\frac{\alpha_1(c)^2}{4} - \alpha_2(c)}$, with

$$\alpha_1(c) := d_{p-1} + d_p + \frac{v_{p-1}^2 + v_p^2}{c}, \quad \alpha_2(c) := d_{p-1}d_p + \frac{v_{p-1}^2d_p + v_p^2d_{p-1}}{c}.$$

The term $\lambda_{p,0}(c_{up})$ approximates the exact modified λ_p up to an error of

$$|\lambda_{p,0}(c_{up}) - \lambda_p| < \lambda_{p,0}(c_{lo}) - \lambda_{p,0}(c_{up}).$$

If the component v_p is large enough, we can simplify the estimate (3.7). To this end, suppose further that the ultimate component v_p is such that $v_p^2 > c_{up}(d_{p-1} - d_p)$. Then

$$(3.8) \quad 0 < d_{p-1} - \lambda_p < \frac{v_{p-1}^2(d_{p-1} - d_p)}{v_p^2 - c_{up}(d_{p-1} - d_p)}.$$

Before we prove the lemma, some remarks are due.

Remark 2.

1. The radicand in (3.7) is always strictly positive. By (3.7), one can show again that a small v_{p-1} and a large v_p lead to the largest growth in the smallest eigenvalue. It is the alternation of signs in the binomial terms in the radicand that causes this behavior. The simplification (3.8) is given in order to emphasize this fact.

2. In the limit case $v_p = 0$, the estimate (3.7) reduces to the trivial estimate $d_{p-1} - \lambda_p \leq d_{p-1} - d_p$, i.e., $\lambda_p \geq d_p$. This is in line with the theory, because the smallest eigenvalue of the original diagonal matrix D remains unperturbed for $v_p = 0$.

3. In the limit case $v_{p-1} = 0$, the estimate (3.7) reduces to

$$d_{p-1} - \lambda_p \leq \begin{cases} d_{p-1} - (d_p + \frac{v_p^2}{c_{up}}) & \text{if } v_p^2 < c_{up}(d_{p-1} - d_p), \\ 0 & \text{if } v_p^2 \geq c_{up}(d_{p-1} - d_p). \end{cases}$$

This reflects that λ_p cannot grow larger than d_{p-1} in accordance with Theorem 3.1.

4. Since the eigenvalues of a Hermitian matrix are real and are given by the roots of the real-valued characteristic function $f(\lambda) = 1 + \sum_{i=1}^p \frac{|v_i|^2}{d_i - \lambda}$ corresponding to (3.4), the above estimates transfer to the Hermitian case after replacing v_i^2 with $|v_i|^2$, throughout.

Proof of Lemma 3.4. Let $\varphi(v, \lambda) = \sum_{l=1}^{p-2} \frac{v_l^2}{d_l - \lambda}$. By the characteristic equation, (3.4), it holds that

$$\frac{v_{p-1}^2}{d_{p-1} - \lambda_p} = \frac{v_p^2}{\lambda_p - d_p} - 1 - \varphi(v, \lambda_p).$$

Because $d_p < \lambda_p < d_{p-1}$, it follows that

$$1 < c_{lo}(v) := 1 + \sum_{l=1}^{p-2} \frac{v_l^2}{d_l - d_p} < 1 + \varphi(v, \lambda_p) < 1 + \sum_{l=1}^{p-2} \frac{v_l^2}{d_l - d_{p-1}} =: c_{up}(v).$$

Therefore,

$$(3.9) \quad \frac{v_{p-1}^2}{d_{p-1} - \lambda_p} - \frac{v_p^2}{\lambda_p - d_p} + c_{up} > 0 > \frac{v_{p-1}^2}{d_{p-1} - \lambda_p} - \frac{v_p^2}{\lambda_p - d_p} + c_{lo}.$$

Both the expression on the left-hand side and the expression on the right-hand side increase monotonically in $\lambda_p \in [d_p, d_{p-1}]$. Since both are negative for λ_p sufficiently close to d_p and positive for λ_p sufficiently close to d_{p-1} , both feature exactly one zero in the interval $[d_p, d_{p-1}]$. The respective zeros can be determined via solving the quadratic equations that are obtained by multiplying (3.9) with $(d_{p-1} - \lambda_p)(\lambda_p - d_p) > 0$. They are given by $\lambda_{p,0}(c) = \frac{\alpha_1(c)}{2} - \sqrt{\frac{\alpha_1(c)^2}{4} - \alpha_2(c)}$, where

$$\alpha_1(c) = d_{p-1} + d_p + \frac{v_{p-1}^2 + v_p^2}{c}, \quad \alpha_2(c) = d_{p-1}d_p + \frac{v_{p-1}^2d_p + v_p^2d_{p-1}}{c}$$

for $c = c_{up}$ and $c = c_{lo}$, respectively. (The second solution to each of the quadratic equations is dispensed with by showing that it is actually larger than d_{p-1} .) Because of the monotonic growth, we know from (3.9) that

$$d_{p-1} > \lambda_{p,0}(c_{lo}) > \lambda_p > \lambda_{p,0}(c_{up}) > d_p,$$

where λ_p is the exact eigenvalue of the rank-one modified matrix. In particular, $d_{p-1} - \lambda_p < d_{p-1} - \lambda_{p,0}(c_{up})$. Substituting the expression for $\lambda_{p,0}(c_{up})$ and rearranging terms gives (3.7).

If $v_p^2 > c_{up}(d_{p-1} - d_p)$, we have the simplified estimate

$$\frac{v_{p-1}^2}{d_{p-1} - \lambda_p} > \frac{v_p^2}{\lambda_p - d_p} - c_{up} > \frac{v_p^2}{d_{p-1} - d_p} - c_{up} > 0,$$

which leads to (3.8) as claimed. □

The next corollary allows us to understand how λ_p may be pushed toward the upper boundary d_{p-1} .

COROLLARY 3.5. *Let $v \in \mathbb{R}^p$ with $v_i \neq 0 \forall i$. Consider the entries $v_i, i = 1, \dots, p-2$, as fixed. Let $v_p^2 > (d_{p-1} - d_p)c_{up}$, where $c_{up} = c_{up}(v) = 1 + \sum_{l=1}^{p-2} \frac{v_l^2}{d_l - d_{p-1}}$. Then*

$$\lambda_p(v) \rightarrow d_{p-1} \text{ for } \begin{cases} v_{p-1} & \rightarrow 0 \text{ or} \\ v_p & \rightarrow \infty. \end{cases}$$

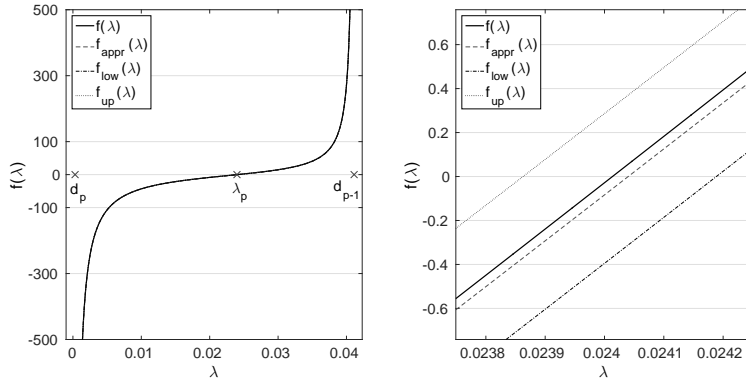


FIG. 2. Graphs of the characteristic function $f(\lambda)$ defined in (3.4) and its approximations implicitly defined by (3.10) for the choices of $c = c(v, \delta)$, $\delta \in \{d_p, \frac{1}{2}(d_{p-1} + d_p), d_{p-1}\}$ corresponding to f_{low} , f_{appr} , and f_{up} , respectively, in the interval (d_p, d_{p-1}) . The underlying data set is listed in Appendix A. Left: On this scale the plots virtually coincide. Right: Detailed view close to the zero locations. This figure illustrates the error bound given in Lemma 3.4.

The corollary follows from the estimate (3.8). Note that none of the limit processes $v_i \rightarrow 0$, $i = 1, \dots, p-2$, necessarily enforce that $\lambda_p(v) \rightarrow d_{p-1}$.

The proof of Lemma 3.4 suggests that we may approximate the unique zero of the characteristic function (3.4) in $[d_p, d_{p-1}]$ by computing the unique zero of

$$(3.10) \quad \frac{v_{p-1}^2}{d_{p-1} - \lambda_p} - \frac{v_p^2}{\lambda_p - d_p} + c = 0$$

in the same interval. Reasonable choices for the additive constant are $c = c(v, \delta) = 1 + \sum_{i=1}^{p-2} \frac{v_i^2}{d_i - \delta}$. The choices of $\delta = d_p$ and $\delta = d_{p-1}$ lead to the lower and upper bounds c_{lo} and c_{up} , respectively. If the growth in λ_p is comparably moderate, a better approximation may be achieved by using the interval center $\delta = \frac{1}{2}(d_{p-1} + d_p)$. However, this choice does not provide a bound.

Figure 2 displays the graph of the function $f(\lambda)$ defined in (3.4) and the graphs of the functions implied by the right hand side of (3.10) for the choices of $c = c(v, \delta)$, $\delta \in \{d_p, \frac{1}{2}(d_{p-1} + d_p), d_{p-1}\}$.

4. Accelerated missing point selection. In this section, we introduce a fast surrogate for the exhaustive greedy point index selection of Algorithm 1. based on the theoretical findings of the previous section. We give preliminary illustrations of the new approach's performance. After this, we point out an inherent limitation of the exhaustive greedy approach that is also shared by the new surrogate and we propose a suitable remedy.

4.1. Greedy point selection using eigenvalue bound estimates. Given a positive definite diagonal matrix D and a finite set of vectors $\{v^j, j \in \bar{J}\}$, where \bar{J} denotes the set of indices that have not yet been considered in the MPE, the greedy objective to pick the one vector $v^{j^{opt}}$ that leads to the largest growth in the smallest eigenvalue of $D + vv^T$ can be reasonably tackled by sorting the set of candidate vectors according to the estimate (3.7). This gives the following surrogate for the exhaustive greedy search:

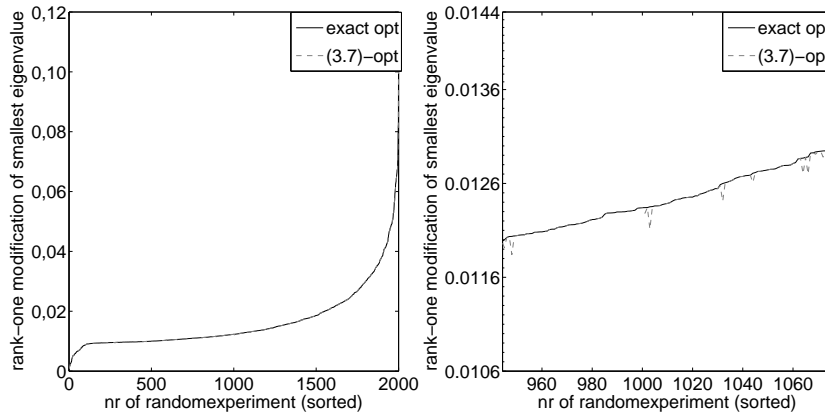


FIG. 3. Results of 2,000 random experiments, each featuring a random positive definite diagonal matrix $D \in \mathbb{R}^{10}$ and set of 1,000 candidate shift vectors $\{v^j, j \in 1, \dots, 1,000\} \subset \mathbb{R}^{10}$. The graphs on the left display the eigenvalues associated with the optimal rank-one modification determined by exhaustively solving the modified eigenvalue problem for each of the 1,000 candidate vectors compared to the eigenvalues achieved by using the best shift vector determined by the (3.7)-criterion. For better readability, the experiments are sorted by size. In 1,788 of the 2,000 cases, both criteria coincided, see left graph. Right: Detailed view of the region where the maximum error occurred.

Minimize the right-hand side of (3.7) over $v^j, j \in \bar{J}$. Add the unit vector $e_{j_{opt}}$ corresponding to the optimal index j_{opt} as a new column to the mask matrix.

A vectorized pseudocode is listed in Appendix B. The associated computational costs per iteration are $\mathcal{O}(np^2)$, the dominating operation being the matrix product in step 3 of Algorithm 4. The performance of this surrogate is visualized for random data in Figure 3.

In 2,000 random experiments, the (3.7)-criterion picked the same vector as the exhaustive minimization in 90% of the cases. In the remaining cases it picked a vector that induced a rank-one modification that is very close to the optimal one. The maximum absolute error is $2.39 \cdot 10^{-4}$. The errors summed over all experiments divided by the number of conducted experiments total $5.8 \cdot 10^{-6}$.

As explained in section 3.2, by using (3.7), we replace the modified eigenvalue problem with computing the zero $\lambda_{p,0}(c_{up})$ of (3.9) rather than the zero of the characteristic function (3.4). This gives a lower bound on the exact eigenvalue λ_p . Figure 4 displays the surfaces of the approximate eigenvalue $\lambda_{p,0}(c_{up})(v_{p-1}, v_p)$ and the exact eigenvalue $\lambda_p(v_{p-1}, v_p)$ for a shift vector $v \in \mathbb{R}^p$ with fixed components $v_1, \dots, v_{p-2} \neq 0$ on the square $(v_{p-1}, v_p) \in [0, 0.5] \times [0, 0.5]$. On the right edge of the figure, the truncated parabola of Remark 2, item 3, is observable. Both surfaces agree up to a maximum absolute error of $1.508 \cdot 10^{-6}$ and a maximum relative error of $8.112 \cdot 10^{-5}$. Both surfaces have z -values bounded from below by the smallest eigenvalue d_p and from above by the second-to-smallest eigenvalue d_{p-1} of the unperturbed diagonal matrix D . In the case considered in the figure, these read $d_p = 0.0130$ and $d_{p-1} = 0.0241$. Note that the closer we push λ_p toward d_{p-1} , the better the approximation via (3.10) with $c = c_{up}$.

At this point, we have established the estimate (3.7) as a fast and accurate surrogate for the exact modified rank-one eigenvalue problem and thus for the exhaustive

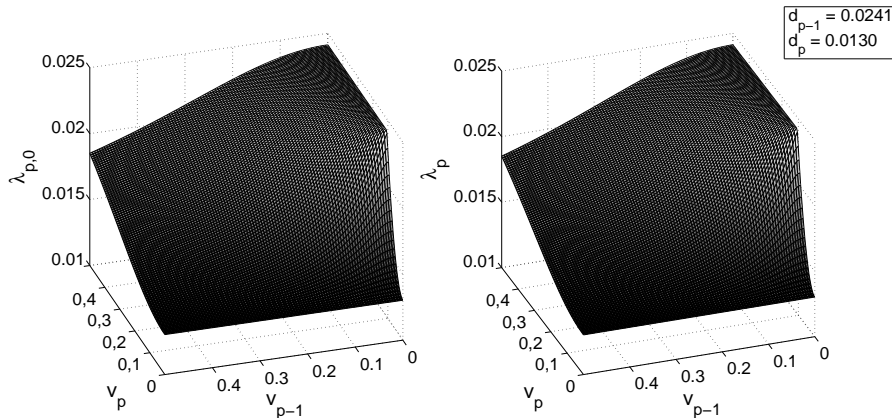


FIG. 4. Left: surface $(v_{p-1}, v_p) \mapsto \lambda_{p,0}(v_{p-1}, v_p)$ of the approximation of the smallest eigenvalue introduced in the proof of Lemma 3.4. Right: surface of the exact rank-one modification $(v_{p-1}, v_p) \mapsto \lambda_p(v_{p-1}, v_p)$, where λ_p is computed by solving the eigenvalue problem $D + vv^T$. In both cases, the components (v_1, \dots, v_{p-2}) of v are kept fixed.

greedy method Algorithm 1.. Surprisingly, section 3 even implies ways of improving the exhaustive algorithm. This will be explained in the next section.

4.2. A cumulative limitation of the exhaustive greedy search. The theoretical investigations from section 3 show that at each iteration of the greedy point selection method, the penultimate singular value bounds the growth of the smallest singular value. Moreover, the greedy selection comes closest to the objective of minimizing the error bound if it picks a shift vector with a small penultimate component and a large last entry; see figure 4. While this leads to the largest leap in the smallest singular value, as a side effect, it also minimizes the impact on the penultimate singular value. (In the extreme case of a zero second-to-last entry, the penultimate singular value stays fixed.) This means that at the following iteration, it is almost the same bound that limits the growth of the smallest singular value. In this way, the greedy optimum at one iteration narrows the possible growth at the next iteration. By Theorem 3.3, the exhaustive greedy method will exhibit this behavior to the *highest possible extent*. Due to the high accuracy of the (3.7)-criterion, this approximation will suffer from the same effect.

4.3. Toward an improved greedy search. The considerations in the previous subsection provide guidance on how to improve the greedy approach for the underlying combinatorial problem: Given a candidate set of shift vectors, we should find a balance between determining the shift vector that has the strongest impact on the smallest eigenvalue—which corresponds to greedily decreasing the error bound—and selecting shift vectors that have a strong impact on the remaining eigenvalues. From Theorem 3.3, it is clear that these objectives oppose each other.

Note that the methodology introduced in Lemma 3.4 transfers to the other eigenvalues. If $D = \text{diag}(d_1, \dots, d_p)$, the dominating terms in the sum expression of the characteristic equation (3.4) that steer the behavior of the modified eigenvalue $\lambda_{p-l} \in (d_{p-l}, d_{p-l-1})$ are the summands with indices $p-l-1$ and $p-l$. From (3.4) we obtain the estimate

$$\frac{v_{p-l-1}^2}{d_{p-l-1} - \lambda_{p-l}} - \frac{v_{p-l}^2}{\lambda_{p-l} - d_{p-l}} + c_{up,l} > 0, \quad \text{where } c_{up,l} = 1 + \sum_{k \neq p-l-1, p-l} \frac{v_k^2}{d_k - d_{p-l-1}}.$$

Algorithm 2. Fast greedy MPE with “modulo-three” target switching.

```

1:  $\bar{J}_s = \{1, \dots, n\} \setminus J_s$ 
2: while  $|J_s| \leq \text{maxpoints}$  do
3:   if  $\text{mod}(s, 3) \leq 1$  then
4:      $l_t = 0$ 
5:   else
6:      $l_t = 1$ 
7:   end if
8:   determine  $j_{opt}$  by sorting  $\{v^j, j \in \bar{J}_s\}$  according to (4.1) for  $l = l_t$ .
9:   update:  $J_{s+1} = J_s \cup \{j_{opt}\}$ ,  $\bar{J}_{s+1} = \bar{J}_s \setminus \{j_{opt}\}$ 
10:   $s = s + 1$ 
11: end while

```

In a fashion analogous to Lemma 3.4, this leads to a certified approximation of λ_{p-l} . Denoting the gap between the $(p-l-1)$ th and the $(p-l)$ th eigenvalue of D by $\gamma_{p-l} = (d_{p-l-1} - d_{p-l})$, and setting $\beta_{p-l} = c_{up,l}\gamma_{p-l}$, the eigenvalue error bound corresponding to the exact λ_{p-l} is

$$(4.1) \quad d_{p-l-1} - \lambda_{p-l} < \frac{\gamma_{p-l}}{2} - \frac{1}{2c_{up,l}} \left(v_{p-l-1}^2 + v_{p-l}^2 - \text{sgn}(c_{up,l}) \sqrt{(\beta_{p-l} - v_{p-l}^2)^2 + (\beta_{p-l} + v_{p-l-1}^2)^2 + 2v_{p-l-1}^2 v_{p-l}^2 - \beta_{p-l}^2} \right).$$

With inequality (4.1), we have an efficient tool at hand to compute fast approximations to any eigenvalue of the rank-one modified eigenproblems that result from step 4 of Algorithm 1.. However, there is one subtlety that has to be considered. For $l \geq 1$, the constant $c_{up,l}$ may take negative values. One can show that if $c_{up,l} < 0$, then the solution to the underlying quadratic equation with the positive sign in front of the square root term gives the sought-after unique zero in the interval (d_{p-l}, d_{p-l-1}) .

In order to improve on the exhaustive greedy search, we propose two algorithms that alternate between directly addressing the smallest eigenvalue and targeting the larger eigenvalues. Algorithm 2. simply switches in every third step of the greedy loop from targeting the largest growth for λ_p to targeting the largest growth for λ_{p-1} .¹ Hence, we make use of (4.1) for $l = l_t$, where either $l_t = 0$ or $l_t = 1$.

The idea behind Algorithm 3. is the following. When the relative distance between d_{p-1} and d_p is comparably large, we interpret the situation as offering enough room for directly improving the error bound. Hence, we directly target the smallest eigenvalue by minimizing (4.1) for $l_t = 0$, which is the same as (3.7). Otherwise, we consider the upper bound d_{p-1} as too constraining for the future growth of λ_p and we address the penultimate eigenvalue λ_{p-1} via minimizing the error bound (4.1) for $l_t = 1$. If the interval (d_{p-2}, d_{p-1}) also happens to be comparably small, we target the next biggest eigenvalue λ_{p-2} via $l_t = 2$, and so forth. The threshold above which an interval (d_{p-k+1}, d_{p-k}) is considered as “large” is set by a parameter $\tau \in (0, 1)$. If the length of none of the bounding intervals (d_{p-k+1}, d_{p-k}) , $k = 1, \dots, p-1$, exceeds the threshold, the algorithm remains at $l_t = 0$, i.e., targeting the smallest λ_p . Therefore, choosing τ close to one means that the if-statement in Algorithm 3. will typically

¹Switching the target every third step represents a best-practice choice. The higher the modulo-number chosen, the more closely the performance of Algorithm 2. will resemble the standard greedy approach.

Algorithm 3. Fast greedy MPE with target switching based on the growth potential.

```

1:  $\bar{J}_s = \{1, \dots, n\} \setminus J_s$ 
2: while  $|J_s| \leq \text{maxpoints}$  do
3:    $l_t = 0$ 
4:   for  $k = 1 : p - 1$  do
5:     if  $(d_{p-k} - d_{p-k+1})/d_{p-k} > \tau$  then
6:        $l_t = k - 1$ 
7:       break
8:     end if
9:   end for
10:  determine  $j_{opt}$  by sorting  $\{v^j, j \in \bar{J}_s\}$  according to (4.1) for  $l = l_t$ .
11:  update:  $J_{s+1} = J_s \cup \{j_{opt}\}$ ,  $\bar{J}_{s+1} = \bar{J}_s \setminus \{j_{opt}\}$ 
12:   $s = s + 1$ 
13: end while

```

return “false”; a very small value of τ means that the if-statement will typically return “true”. In both cases, Algorithm 3. is designed to essentially fall back to the standard greedy choice of always targeting the smallest index.

5. Numerical experiments. The illustrations given in section 4 consider the scenario that we encounter in a single iteration of the greedy point selection. In this section, we assess the performance of the proposed approximation to the rank-one eigenvalue problem given by (3.7), when applied sequentially in the greedy loop.

5.1. Basic performance test. First, we conduct an experiment for assessing the computational efficiency of the proposed approach. To this end, we consider a column-orthogonal basis matrix $U \in \mathbb{R}^{n \times p}$ with $n = 10,727$, $p = 23$.² The objective is to compute a subset of indices out of the set $\{1, \dots, n\}$ such that the norm of the corresponding masked projector $\|U(U^T P P^T U)^{-1} U^T P P^T\| = 1/\sigma_{\min}(P^T U)$ is minimized. We determine $p = 23$ initial points via the DEIM point selection [6, Algorithm 1]. Starting from this set of points, we add another 477 points to the index set and the corresponding unit vector columns to the mask matrix P by using four approaches: (i) the exhaustive greedy search Algorithm 1., (ii) the greedy method based on the estimate (3.7), (iii) the enhanced greedy scheme Algorithm 2., and (iv) the enhanced greedy scheme Algorithm 3. with the user parameter τ set to $\tau = 0.05$. This choice means that when the lower bound d_{p-k+1} for the modified eigenvalue λ_{p-k+1} is larger than 95% of the associated upper bound d_{p-k} , then the growth potential is rated as poor and the next index is targeted.

The associated computation times are 1173.5s for the exhaustive greedy method and 2.5s for the various surrogates.³

Figure 5 shows the growth of the smallest five singular values of $P_s^T U \in \mathbb{R}^{s \times p}$ when iteratively adding new unit vector columns to the mask matrix $P_s \in \{0, 1\}^{n \times s}$, $s = 6, \dots, 477$. The nonzero entry of the added unit column appears at the optimal point index determined via either the exhaustive greedy search Algorithm 1. (Figure 5, left) or the enhanced greedy schemes Algorithms 2. and 3. (Figure 5, middle and right, resp.). As explained in section 4.2 the standard greedy method suffers from its

²In this example, the matrix U stems from a POD of aerodynamic flow snapshots.

³All computations were carried out on a 64-bit Dell laptop computer endowed with four Intel Core i7-2637M CPU @ 1.70-GHz processors and 3.2 GB memory.

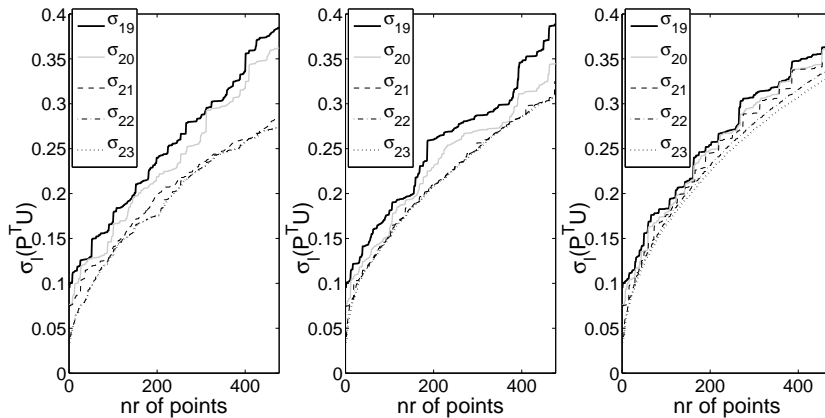


FIG. 5. Missing point index selection for a basis matrix $U \in \mathbb{R}^{n \times p}$ with $n = 10,727$, $p = 23$. The initial p points are determined by the DEIM point selection [6, Alg. 1]. An additional 477 points are selected based on the exhaustive greedy method, Algorithm 1. (left), based on Algorithms 2. (middle) and 3. (right), respectively. The graphs show the growth behavior of the five smallest singular values of $P^T U \in \{0,1\}^{s \times p}$ when adding columns to P . The final values of σ_p are 0.2736 (left), 0.3074 (middle), 0.3315 (right).

inherent growth restriction. This is confirmed by the left-most image of the figure. The lines associated with the modified singular values σ_{p-1} and σ_p almost coincide. The plot in the middle of Figure 5 is associated with Algorithm 2., which switches between targeting the smallest and second-to-smallest singular value at every third step of the greedy iteration. The figure shows that the algorithm uses the full growth potential of the two smallest singular values. However, the third-smallest singular value now acts as a constraint. Finally, the right plot in Figure 5 shows the performance of Algorithm 3. for $\tau = 0.05$. In the case at hand, Algorithm 3. has targeted 330 times the growth of σ_p , 112 times σ_{p-1} , 20 times σ_{p-2} , 8 times σ_{p-3} , 6 times σ_{p-4} , and 1 time σ_{p-5} , while selecting the 477 additional indices for the mask matrix. The growth lines of the last five eigenvalues remain close to each other but are well separated. The largest end value of σ_p is achieved by this approach.

Figure 6 shows the decrease of the error bound $1/\sigma_{\min}(P^T U)$ versus the number of points added to the index set J , respectively, columns added to the mask matrix. As expected from the results of section 4, the (3.7)-criterion closely mimics the behavior of the exhaustive search, yet it is about 470 times faster. The enhanced methods Algorithms 2. and 3. are just as fast but achieve an even lower error bound. The left plot in Figure 6 shows the behavior when adding the first 30 points/columns. The points selected by the exhaustive greedy method and the (3.7)-approximation are exactly the same. Moreover, in this detailed view, the 3-cycle of Algorithm 2. is clearly visible. The more sophisticated target selection of Algorithm 3. pays off in the longer run.

5.2. Comparison with the true combinatorial optimum. We compare the various greedy MPE methods to the true combinatorial optimum for a sufficiently small academic test case. We consider a random orthogonal matrix $U \in \mathbb{R}^{n \times p}$ with $n = 60$, $p = 5$, and we use DEIM to construct an initial set of $p = 5$ point indices. We then use the various greedy MPE methods to add another five point indices to

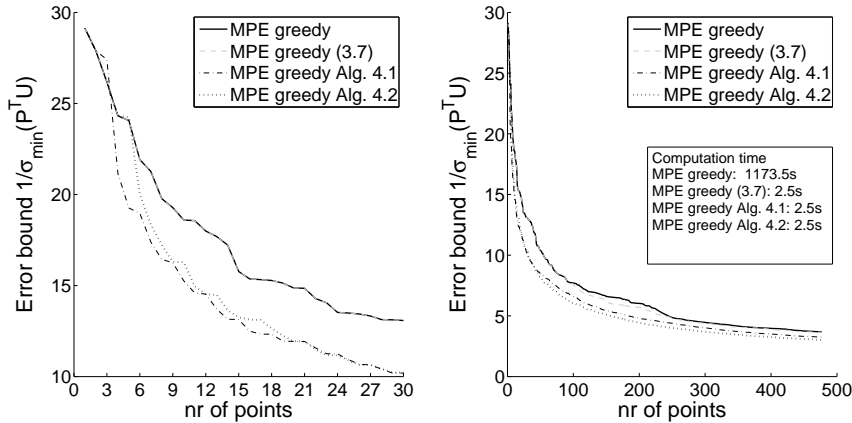


FIG. 6. Missing point index selection for a basis matrix $U \in \mathbb{R}^{n \times p}$ with $n = 10,727$, $p = 23$. The initial p points are determined by the DEIM point selection [6, Alg. 1]. An additional amount of 477 points is selected based on the exhaustive greedy method Algorithm 1., based on the (3.7)-criterion and based on Algorithms. 2. and 3., respectively. Left: Decrease of the error bound after adding the first 30 additional points.

TABLE 1

Comparison of the error bound (2.2) with the exact value after adding five point indices to an initial index set of $p \in \{5, 20\}$ DEIM points. Here, $E[\frac{\|y - \Pi(y)\|_2}{\|y - UU^T y\|_2}]$ denotes the mean averaged over 100,000 random vectors y .

$U \in \mathbb{R}^{60 \times 5}$	$E[\frac{\ y - \Pi(y)\ _2}{\ y - UU^T y\ _2}]$	$\ \Pi\ _2$	Comp. time
MPE greedy	1.181	3.068	$\approx 0.01s$
MPE greedy (3.7)	1.181	3.068	$< 0.005s$
MPE greedy Alg. 4.1	1.175	2.914	$< 0.005s$
MPE greedy Alg. 4.2	1.157	2.919	$< 0.005s$
Combinatorial optimum	1.139	2.346	2.26h
$U \in \mathbb{R}^{100 \times 20}$	$E[\frac{\ y - \Pi(y)\ _2}{\ y - UU^T y\ _2}]$	$\ \Pi\ _2$	Comp. time
MPE greedy	1.626	5.674	$\approx 0.05s$
MPE greedy (3.7)	1.626	5.674	$< 0.01s$
MPE greedy Alg. 4.1	1.556	4.362	$< 0.01s$
MPE greedy Alg. 4.2	1.556	4.362	$< 0.01s$
Combinatorial optimum	1.507	3.981	2 days, 3.32h

the DEIM point selection, which yields in each case a mask matrix $P \in \mathbb{R}^{n \times (p+5)}$ and an associated masked projector $\Pi = U(U^T P P^T U)^{-1} U^T P P^T$. Subsequently, we form a random vector $y \in \mathbb{R}^n$ and compare the exact relative distance $\frac{\|y - \Pi(y)\|_2}{\|y - UU^T y\|_2}$ to the error bound $\|\Pi\|_2$ from (2.2). Moreover, we compute the actual best five additional points by a brute force trial of all possible combinations and compute the quantities $\frac{\|y - \Pi(y)\|_2}{\|y - UU^T y\|_2}$, $\|\Pi\|_2$ for the resulting missing point index set.

We repeat the exercise for $n = 100$, $p = 20$. In this case, the exhaustive combinatorial optimization takes more than 2 days of computation time, even though this academic test case is smaller than the dimensions to be expected in realistic problems. Table 1 displays the results.

For the $n = 100, p = 20$ test case the DEIM point selection is

$$J_{DEIM} = \{3, 6, 7, 20, 22, 23, 28, 30, 33, 34, 36, 37, 40, 41, 50, 58, 65, 72, 79, 99\}.$$

The five additional indices determined by the various MPE options are

$$\begin{aligned} J_{MPEgreedy} &= \{90, 16, 26, 76, 97\} = J_{MPE(3.7)}, \\ J_{MPE(Alg.4.1)} &= \{90, 16, 2, 66, 8\} = J_{MPE(Alg.4.2)}, \\ J_{comb.opt.} &= \{9, 14, 77, 24, 43\}. \end{aligned}$$

5.3. An engineering example. This section features an application of the accelerated missing point selection to a computational fluid dynamics model reduction problem. More precisely, the accelerated MPE is applied within the context of the linear frequency domain (LFD) approach to unsteady fluid dynamics.

The LFD approach applies to time-accurate flows governed by the Navier–Stokes equations under small, approximately periodic perturbations, the key feature being a transition to the frequency domain via a Fourier transformation. The resulting equations are linearized in the frequency domain by a truncation after the first harmonic terms. Even briefly sketching the LFD method is beyond the scope of this paper and the interested reader is referred to [16] and [19] for details. For our purpose, it is sufficient to note that the LFD approach leads to the problem of solving a large-scale sparse complex-valued parametric linear equation system associated with the unstructured spatial discretization of a set of partial differential equations:

$$(5.1) \quad A(M, \kappa)W = b(M, \kappa), \quad A(M, \kappa) \in \mathbb{C}^{N \times N}, b(M, \kappa) \in \mathbb{C}^N.$$

The parameters of interest are the free-stream *Mach number*, M , which is the far field flow velocity divided by the speed of sound, and the *reduced frequency*, κ , which is the frequency of the periodic perturbations normalized by a reference length and the free-stream velocity. The state vector $W = W(M, \kappa) \in \mathbb{C}^N$ contains the discretized flow variables transformed to the frequency domain, i.e., the Fourier transformations of the density, the velocity components, the total energy, and the turbulent viscosity. Hence, the dimension of the discretized state vector for the flow around a two-dimensional airfoil associated with a computational grid of degree n is $N = 5n$.

Following [19], we replace the full model (5.1) by a weighted least-squares problem restricted to a low-dimensional POD subspace. The reduced model is based on the ansatz of decoupling the Mach number and the reduced frequency dependency via $W_r(M, \kappa) = U_M \alpha(\kappa)$, where $U_M \in \mathbb{C}^{N \times p}$, $U_M^H U_M = I_p$, represents a suitable unitary basis that spans the space of approximate solutions at a certain Mach number M . Given U_M , the coefficient vector $\alpha(\kappa) \in \mathbb{C}^p$ is determined by the minimum residual condition in the so-called nondescriptor L_2 -metric

$$(5.2) \quad \min_{\alpha(\kappa) \in \mathbb{C}^p} \|A(M, \kappa)U_M \alpha(\kappa) - b(M, \kappa)\|_S^2.$$

Here, $S \in \mathbb{R}^{N \times N}$ is a diagonal matrix of positive weights inducing the inner product $\langle v, w \rangle_S := v^H S w$ on \mathbb{C}^N associated with the spatial discretization. The proper weights are the reciprocal values of the volumes of the corresponding grid cells $S_{jj} = \Omega_j^{-1}$; see [19] for details. Computationwise, the dominant operation is the (sparse) matrix product $A(M, \kappa)U_M$, which has an $\mathcal{O}(Nrp)$ -FLOP count, where r is the average number of nonzero entries per row in $A(M, \kappa)$. This product has to be recomputed

for each value of κ .⁴ Note that usually $N \gg p$, so that the system (5.2) is massively overdetermined.

MPE-reduced least-squares system. As explained in section 2, we use a mask matrix $P \in \{0, 1\}^{n \times s}$ to reduce the costs associated with the least-squares system (5.3). In this way, the full matrix product $A(M, \kappa)U_M$ is omitted. The non-Euclidean norm is transferred to the complex 2-norm via $\|v\|_S = \|\sqrt{S}v\|_2$. The resulting masked, weighted least-squares problem restricted to U_M reads

$$(5.3) \quad \min_{\alpha(\kappa) \in \mathbb{C}^p} \|P^T \sqrt{S}A(M, \kappa)U_M \alpha(\kappa) - P^T \sqrt{S}b\|_2^2.$$

The matrix products are evaluated as follows: the product $P^T \sqrt{S}A(M, \kappa)$ is performed by picking the rows out of the matrix $A(M, \kappa)$ that are indicated by the mask matrix and by multiplying each selected row with the associated weight in \sqrt{S} . This results in a sparse ($s \times N$)-matrix. Hence, neither P nor S needs to be formed as actual matrices. Then, the right-multiplication with U_M is performed. Due to the sparsity of the rows of $A(M, \kappa)$, the FLOP count is $\mathcal{O}(rsp)$ and is thus completely independent of the scale N of the spatial discretization. After having constructed the reduced operator, the remaining operations for solving (5.3) are also independent of N . (For dense matrices, we arrive at a FLOP count of $\mathcal{O}(Nsp)$ versus $\mathcal{O}(N^2p)$ for the original system (5.2).)

We refer to the flow approximations $W_r(M, \kappa) = U_M \alpha(\kappa)$ obtained by solving (5.3) for $\alpha \in \mathbb{C}^p$ as *MPE-ROM solutions*. The flow approximations $W_r(M, \kappa) = U_M \alpha(\kappa)$ obtained by solving the full least-squares system (5.2) for $\alpha \in \mathbb{C}^p$ will be called *minres-ROM solutions*.

For the numerical experiments, we conduct reduced predictions at a fixed transonic Mach number of $M_0 = 0.802$ for the flow around the NACA 64A010 airfoil on a grid of degree $n = 10,727$, so that $N = 53,635$. A POD of $p = 5$ state vector snapshots $\{W(M_0, \kappa) \in \mathbb{C}^N, \kappa = 0.1, 0.3, 0.5, 0.7, 0.9\}$ is used as the reduced space of approximate solutions, i.e.,

$$U_{M_0} \stackrel{POD}{=} (W(M_0, \kappa_1), W(M_0, \kappa_2), W(M_0, \kappa_3), W(M_0, \kappa_4), W(M_0, \kappa_5)) \in \mathbb{C}^{N \times p}.$$

The state snapshots are computed with DLR's TAU-LFD solver [16].

To begin with, we repeat the performance test of subsection 5.1. As before, the first p point indices are determined via [6, Alg. 1]. In this case, this algorithm is applied to the matrix $Q \in \mathbb{C}^{N \times p}$ obtained from an SVD of $A(M, \kappa)U_M \stackrel{SVD}{=} QSV^T$. Starting from this set of points, we add 495 columns to the mask matrix P by using the exhaustive greedy search Algorithm 1. and the greedy methods based on the (3.7)-criterion and Algorithms 2. and 3. with the user parameter τ set to $\tau = 0.05$. The computation time for the former is 2434.0s, while the latter ones take 7.0s each.

Figure 7 shows the decrease of the error bound $1/\sigma_{\min}(P^T Q)$ versus the number of points added to the missing point index set. The essential observations of section 5.1 are confirmed by this figure. The accelerated point selection procedures now exhibit a speed-up factor larger than 347 when compared with the exhaustive greedy method. This is slightly less than the speed-up factor observed in the example of section 5.1, because in this case, the modified eigenvalue problem to be

⁴In the special case at hand, $A(M, \kappa)$ actually exhibits an affine dependency in κ that alleviates this [19, sect. 4.2]. Yet, for a proof of concept, here, we treat the parametric dependency as completely arbitrary.

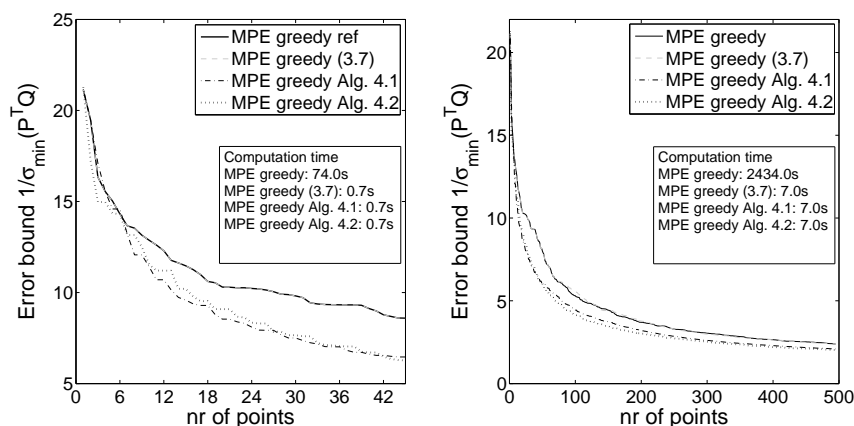


FIG. 7. Missing point index selection for a complex basis matrix $Q \in \mathbb{C}^{n \times p}$ with $n = 53,635$, $p = 5$. The initial p points are determined by the DEIM point selection [6, Alg. 1]. An additional 495 points are selected based on the exhaustive greedy method Algorithm 1., based on the (3.7)-criterion and based on Algorithms 2. and 3.. Left: Decrease of the error bound after adding the first 45 additional points.

solved within the exhaustive greedy loop is only $p = 5$ -dimensional and thus quite small. The difference in computational time between the exhaustive greedy method and the (3.7)-criterion will become larger when the number of columns of the input matrix, and thus also the dimension of the modified eigenvalue problem, increases. As anticipated from section 4.3, Algorithms 2. and 3. outperform the standard approach.

Next, we compare the accuracy of the minres-ROM solutions with the accuracy of MPE-ROM solutions based on the fast approximations. Approximate flow solutions are computed at intermediate reduced frequencies $\kappa \in \{0.2, 0.4, 0.6, 0.8\}$.

Figure 8 displays the relative errors of the minres-ROM solutions, as well as the MPE-ROM solutions following the projection-based point selection scheme PSS1 and the residual-based point selection scheme PSS2. For each scheme, the first five columns of the mask matrix P are computed via [6, Alg. 1] using U and Q as the input matrix, respectively. Afterward, an additional 45 columns are added to the mask matrix P . For the exhaustive greedy method, this takes 74s. Conducting the accelerated enhanced greedy selections takes 0.7s each. As mentioned in section 2.3, the error bound associated with the residual-based point selection strategy PSS2 formally only applies to a certain parameter condition. Yet, the figure shows that reusing the set of filter points obtained via Algorithm 3. at, say, $\kappa = 0.4$ for the predictions at the other reduced frequencies leads to comparable results and outperforms in all but one cases the projection-based point-selection strategy PSS1. (Recycling the mask matrix from $\kappa = 0.4$ is an arbitrary choice; the point filters specific to $\kappa = 0.2, 0.4, 0.8$ do equally well.)

Figure 9 shows the surface distributions of the frequency domain energy of the full minres-ROM approximation, the exhaustive greedy MPE-ROM approximation, and the MPE-ROM based on PSS2 combined with Algorithm 3. compared with the reference TAU-LFD solution. At this Mach number, the flow field exhibits sonic shocks which produce the peaks of the plots. The figure also displays the shape of

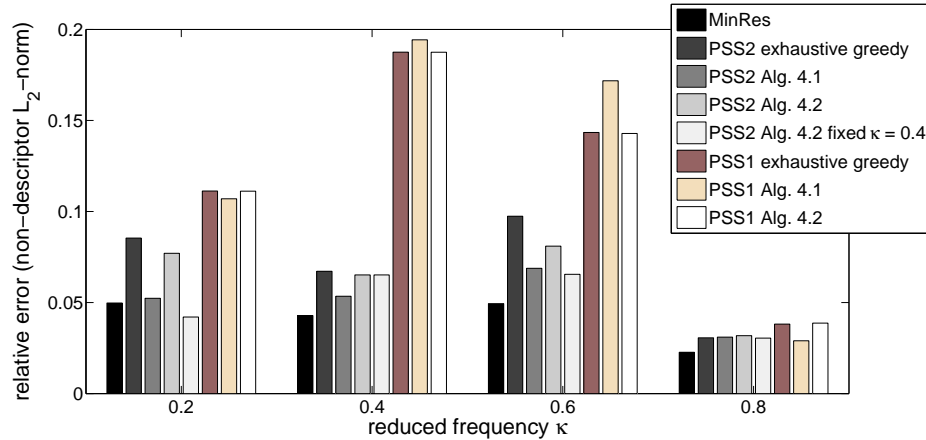


FIG. 8. Relative errors for the flow approximations associated with the full minimum residual model (5.2) and the MPE model (5.3) based on using only 50 rows of the complete set of $N = 53,656$ rows. The mask matrix is computed via the exhaustive greedy method following PSS1, PSS2, and the enhanced greedy Algorithms 2. and 3.. In addition, we show the error bars corresponding to the PSS2-MPE model, which uses the mask matrix computed for $\kappa = 0.4$ with Algorithm 3. throughout at the remaining parameter locations.

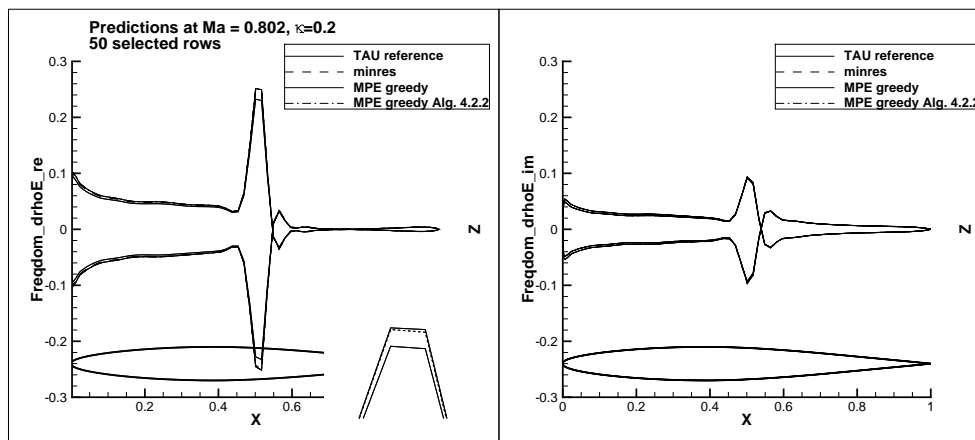


FIG. 9. Distributions of the frequency domain energy at the surface of the NACA 64A010 airfoil computed with the full minres-ROM, the exhaustive greedy MPE-ROM, and the Algorithm 3.-based MPE-ROM, in comparison to the reference LFD solution of DLR's TAU code. Left: Real part of frequency domain energy. Right: Imaginary part of frequency domain energy.

the NACA 64A010 airfoil in order to relate the shock location to the airfoil. For more details on the engineering aspects of this example, see [16].

6. Conclusion. We have exposed that each iteration of the exhaustive greedy missing point estimation necessitates solving a series of rank-one modified eigenvalue problems, where the objective is to find the rank-one modification that leads to the largest growth in the smallest eigenvalue of the reference matrix. The number of vec-

tors considered for inducing the rank-one modifications depends on the dimension n of the underlying (discretized) full model. By theoretical analysis, we identified the dominating factors that control the growth behavior of the modified eigenvalues. Based on this information, we introduced a fast surrogate that sorts the set of candidate vectors according to an approximate solution of the associated eigenvalue problem. If p denotes the dimension of the subspace associated with the reduced model, then the computational costs per iteration for the surrogate are $\mathcal{O}(np^2)$, compared to $> \mathcal{O}(np^3)$ for the exhaustive greedy search. For the examples considered, the surrogate is highly accurate in the sense that it selects in most cases the same vector as the exhaustive greedy search, which is of higher priority than the actual accuracy of the eigenvalue approximation. Moreover, we have exposed a limitation that is inherent to the greedy approach and introduced a variation that alleviates this problem.

The method works only for adding *more columns to the mask matrix* than there are basis vectors in the subspace associated with the reduced model, and thus may be used as a complement to the DEIM algorithm, which provides just as many selected point indices (and, consequently, columns in P) as there are basis vectors. In our numerical experiments, the accelerated improved greedy search was two orders of magnitude faster than its standard counterpart. In addition, it arrived at a lower objective function value in all cases considered.

Appendix A. Data used for producing Figures 1 and 2. We have employed $M = D + vv^T \in \mathbb{R}^{10}$, where $D = \text{diag}(d_1, \dots, d_{10})$ and

$$\begin{pmatrix} d_1 \\ \vdots \\ d_{10} \end{pmatrix} = \begin{pmatrix} 27.610194395368403 \\ 2.761881284022761 \\ 1.923333064320836 \\ 1.361924156790169 \\ 1.006972407162030 \\ 0.508704827545440 \\ 0.260909161469627 \\ 0.220879298701363 \\ 0.041124946138949 \\ 0.000287864608653 \end{pmatrix}, \quad v = \begin{pmatrix} 0.272480414329213 \\ -0.049940997962530 \\ 0.165293429904062 \\ -0.969295197028605 \\ 0.177632055984951 \\ 0.460800409235046 \\ 0.723363805017810 \\ -0.491873666872008 \\ -0.561513294667482 \\ 0.755446933707137 \end{pmatrix}.$$

Appendix B. Efficient evaluation of the eigenvalue approximation. It is a well-known fact that high-level scripting languages like SciPy [11] and MATLAB [12] are slow when it comes to entrywise matrix operations within larger loops. A pseudocode that avoids such operations is given in Algorithm 4., where we orientate ourselves by the MATLAB syntax. The code solves all quadratic equations required for evaluating the (3.7)-criterion efficiently in a blockwise way. In Algorithm 4., the symbol $\mathbf{1}$ denotes the vector with all entries equal to one of suitable dimension. The symbol $U[\bar{J}_s, :]$ means to take all rows corresponding to the index set \bar{J}_s . Likewise, $V_{sq}[:, p]$ means to take the p th column of the matrix V_{sq} .

Note that Algorithm 4. describes a single greedy iteration for adding a new index j_{opt} to the previous-stage index set J_s . By Algorithm 4., step 2, every such iteration requires the SVD of an $s \times p$ matrix. According to (3.1), this is in fact an update problem where a new row is added to an existing SVD. There are methods that achieve this more efficiently than recomputing the SVD from scratch [4] and it is exactly one such method that is utilized in approximate form within the greedy loop in our accelerated algorithm. Yet, because the subspace dimension p and the maximum number of missing point indices allowed in practical applications are expected to be comparably small, the savings of employing such an SVD update technique once more *after* the new index has been determined might be negligible.

Algorithm 4. Efficient greedy step based on (3.7).

Input: $U \in \mathbb{R}^{n \times p}$, $J_s = \{j_1, \dots, j_s\} \subset \{1, \dots, n\}$, $P_s = (e_{j_1}, \dots, e_{j_s}) \in \mathbb{R}^{n \times s}$, where

- $s \geq p$.
- 1: $\bar{J}_s = \{1, \dots, n\} \setminus J_s$ # complementary index set
 - 2: $\Psi \Sigma \Phi^T \stackrel{\text{SVD}}{=} P_s^T U$, $d = \text{diag}(\Sigma^2)$
 - 3: $V = U[\bar{J}_s, :] \cdot \Phi$ # standard matrix-vector mult.
 - 4: $c_{up,vec} = 1 ./ (d - d[p]) \begin{pmatrix} \mathbf{1} \\ 0 \end{pmatrix}$ # elementwise division, $\mathbf{1} \in \mathbb{R}^{p-1}$
 - 5: $c_{up,vec}[p-1] = 0.0$, $c_{up,vec}[p] = 0.0$
 - 6: $V_{sq} = V * V$ # elementwise row-by-row mult.
 - 7: $c_{up,array} = \mathbf{1} + V_{sq} \cdot c_{up,vec}$ # standard matrix-vector mult.
 - 8: $g_1 = d[p-1] + d[p]$, $g_2 = d[p-1] \cdot d[p]$
 - 9: $\alpha_{1,array} = g_1 \mathbf{1} + (V_{sq}[:, p-1] + V_{sq}[:, p]) ./ c_{up,array}$
 - 10: $\alpha_{2,array} = g_2 \mathbf{1} + (d_p V_{sq}[:, p-1] + d_{p-1} V_{sq}[:, p]) ./ c_{up,array}$
 - 11: $\text{screenvec} = d_{p-1} \mathbf{1} - 0.5 \alpha_{1,array} + \sqrt{0.25 \alpha_{1,array} \cdot \alpha_{1,array} - \alpha_{2,array}}$
 - 12: # square-root applies elementwise
 - 13: $[val, j_{opt}] = \text{sort}(\text{screenvec}, \text{'ascend'})$ # return value and index of smallest entry
- Output:** $J_{s+1} = J_s \cup \{j_{opt}\}$, $P_{s+1} = [P, e_{j_{opt}}]$, $\bar{J}_{s+1} = \bar{J}_s \setminus \{j_{opt}\}$
-

REFERENCES

- [1] P. ASTRID, S. WEILAND, K. WILLCOX, AND T. BACKX, *Missing points estimation in models described by proper orthogonal decomposition*, IEEE Trans. Automat. Control, 53 (2008), pp. 2237–2251.
- [2] 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*, C. R. Math. Acad. Sci. Paris I, 339 (2004), pp. 667–672.
- [3] P. BENNER, S. GUGERCIN, AND K. WILLCOX, *A Survey of Model Reduction Methods for Parametric Systems*, Tech. report MPIMD/13-14, Max Planck Institute, Magdeburg, 2013.
- [4] J. R. BUNCH AND C. P. NIELSEN, *Updating the singular value decomposition*, Numer. Math., 31 (1978), pp. 111–129.
- [5] J. R. BUNCH, C. P. NIELSEN, AND D. C. SORENSEN, *Rank-one modifications of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.
- [6] S. CHATURANTABUT AND D. SORENSEN, *Nonlinear model reduction via discrete empirical interpolation*, SIAM J. Sci. Comput., 32 (2010), pp. 2737–2764.
- [7] Z. DRMAC AND S. GUGERCIN, *A New Selection Operator for the Discrete Empirical Interpolation Method — Improved A Priori Error Bound and Extensions*, CoRR, arXiv:1505.00370, 2015.
- [8] D. GALBALLY, K. FIDKOWSKI, K. WILLCOX, AND O. GHATTAS, *Non-linear model reduction for uncertainty quantification in large-scale inverse problems*, Internat. J. Numer. Methods Engrg., 81 (2010), pp. 1581–1608.
- [9] G. H. GOLUB, *Some modified matrix eigenvalue problems*, SIAM Rev., 15 (1973), pp. 318–334.
- [10] M. GU AND S. C. EISENSTAT, *A stable and efficient algorithm for the rank-one modification of the symmetric eigenproblem*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1266–1276.
- [11] E. JONES, T. OLIPHANT, P. PETERSON, ET AL., *SciPy: Open Source Scientific Tools for Python*, <http://www.scipy.org/> (2001).
- [12] *MATLAB, Version 7.10.0 (R2010a)*, The MathWorks, Natick, MA, 2010.
- [13] B. PEHERSTORFER AND K. WILLCOX, *Online Adaptive Model Reduction for Nonlinear Systems*, Tech. report ACDL TR-14-1, Massachusetts Institute of Technology, Cambridge, MA, 2014.
- [14] S. SARGSYAN, S. L. BRUNTON, AND N. KUTZ, *Nonlinear Model Reduction for Complex Systems Using Sparse Optimal Sensor Locations From Learned Nonlinear Libraries*, arXiv:1501.04590v1 [nlin.PS], 2015.
- [15] R. C. THOMPSON, *The behavior of eigenvalues and singular values under perturbations of restricted rank*, Linear Algebra Appl., 13 (1976), pp. 69–78.

- [16] R. THORMANN AND M. WIDHALM, *Linearised frequency domain predictions of dynamic response data for viscous transonic flows*, AIAA J., 51 (2013), pp. 2540–2557.
- [17] J. H. WILKINSON, *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford, UK, 1965.
- [18] K. WILLCOX, *Unsteady Flow Sensing and Estimation Via the Gappy Proper Orthogonal Decomposition*, in proceedings of the 34th AIAA Fluid Dynamics Conference and Exhibit, 2004.
- [19] R. ZIMMERMANN, *A locally parametrized reduced order model for the linear frequency domain approach to time-accurate computational fluid dynamics*, SIAM J. Sci. Comput., 36 (2014), pp. B508–B537.