

MATHICSE Technical Report

Nr. 35 .2016

September 2016



A reduced basis approach to large-scale pseudospectra computation

Petar Sirković

A reduced basis approach to large-scale pseudospectra computation ^{*}

Petar Sirković[†]

September 16, 2016

Abstract

For studying spectral properties of a non-normal matrix $A \in \mathbb{C}^{n \times n}$, information about its spectrum $\sigma(A)$ alone is usually not enough. Effects of perturbations on $\sigma(A)$ can be studied by computing ε -pseudospectra, that is the level-sets of the resolvent norm function $g(z) = \|(zI - A)^{-1}\|_2$. The computation of ε -pseudospectra requires determining the smallest singular values $\sigma_{\min}(zI - A)$ on for all z on a portion of the complex plane. In this work, we propose a reduced basis approach to pseudospectra computation that provides highly accurate estimates of pseudospectra in the region of interest. It incorporates the sampled singular vectors of $zI - A$ for different values of z and implicitly exploits their smoothness properties. It provides rigorous upper and lower bounds for the pseudospectra in the region of interest. We also present a comparison of our approach to several existing approaches on a number of numerical examples, showing that our approach provides significant improvement in terms of computational time.

1 Introduction

Let $A \in \mathbb{C}^{n \times n}$ be a non-normal matrix and $\varepsilon > 0$. Effects of perturbations on the spectrum of A

$$\sigma(A) = \{z \in \mathbb{C} : \|(zI - A)^{-1}\|_2 = \infty\}$$

can be studied by computing the so-called ε -pseudospectra:

$$\sigma_\varepsilon(A) := \{z \in \mathbb{C} : \|(zI - A)^{-1}\|_2 > \varepsilon^{-1}\},$$

which can also be seen as lower level sets of the function

$$g(z) = \|(zI - A)^{-1}\|_2 = \sigma_{\min}(zI - A). \quad (1)$$

The evaluation of $g(z)$ in a domain of interest $D \subset \mathbb{C}$ provides $\sigma_\varepsilon(A) \cap D$ for all $\varepsilon > 0$. We consider a large-scale setting, where evaluating $g(z)$ exactly using the standard techniques is computationally feasible only for a few values of $z \in \mathbb{C}$. The aim is to compute an approximation $\tilde{g}(z) \approx g(z)$ on the whole domain D using only few exact computations of $\sigma_{\min}(zI - A)$.

^{*}Supported by the SNF research module *A Reduced Basis Approach to Large-Scale Pseudospectra Computations* within the SNF ProDoc *Efficient Numerical Methods for Partial Differential Equations*.

[†]ANCHP, MATHICSE, EPF Lausanne, Switzerland. petar.sirkovic@epfl.ch

An example of how pseudospectral images look like can be seen in Figure 1. Using a coarse grid, as in Figure 1a, usually does not capture the full variation of $\|(zI - A)^{-1}\|_2$, making the use of a finer grid necessary as in Figure 1b. However, as the exact computation of the presented resolvent norms on the fine grid takes approximately 10 hours, there is a clear need for a computationally more efficient way to compute ε -pseudospectra.

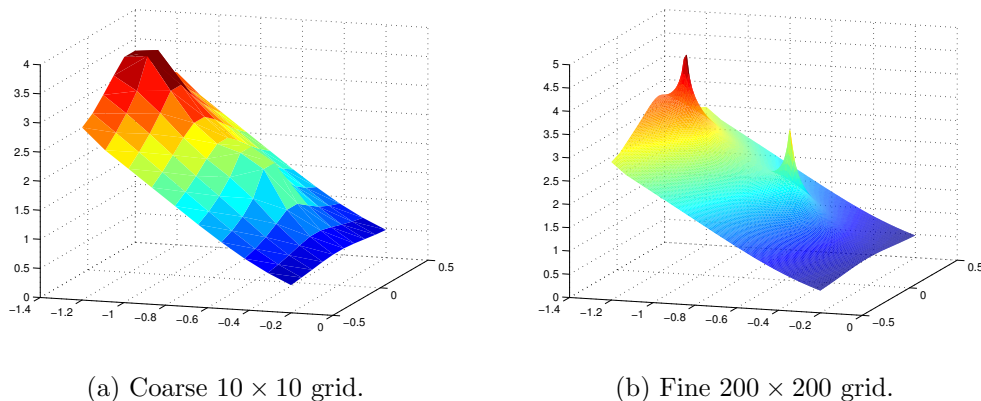


Figure 1: Resolvent norms $\log_{10} \|(zI - A)^{-1}\|_2$, for Example 4.3 with $A \in \mathbb{R}^{9512 \times 9512}$, evaluated on $D = [-1.2, -0.2] + [-0.5, 0.5]i$ using the grid-based approach on a rough grid (left) and on a fine grid (right).

Applications of pseudospectra include linearized stability analysis in fluid mechanics [38], the convergence analysis and design of iterative methods [4, 43], the asymptotic behavior of matrix functions [24, 43] and operator theory [6, 11, 12, 22, 44]. By definition, pseudospectra can also be used to quantify the effects of perturbations and uncertainties on computed eigenvalues and eigenvectors. A more detailed overview of pseudospectra applications can be found in [43].

EigTool [45], the most commonly used software for pseudospectra computation, uses a grid-based approach, where $g(z)$ is computed for a finitely many points z_1, \dots, z_m , typically arranged on a uniformly spaced rectangular grid, requiring $\mathcal{O}(mN^3)$ operations. The approach can be made more efficient if a Schur decomposition of the matrix A is available, resulting in $N^3 + mN^2$ operations [33, 42]. However, for large values of N , this still remains computationally too expensive. For large sparse matrices A , a grid-based approach can be made more efficient by using a sparse LU decomposition [7, 13], while in cases when $\sigma_\varepsilon(A)$ is computed only for fixed value of ε , path following techniques may be used, typically requiring fewer evaluations of $g(z)$ than a grid based approach [5, 8, 35].

For large matrices A , the projection-based approaches have been proposed, where given an orthonormal basis $U \in \mathbb{R}^{N \times k}$ for a subspace $\mathcal{U} \subset \mathbb{C}^N$, we have the following inclusion

$$\sigma_\varepsilon(A) \subset \sigma_\varepsilon(U, AU),$$

with $\sigma_\varepsilon(G, H) = \{z \in \mathbb{C} : \sigma_{\min}(zG - H) < \varepsilon\}$, for $G, H \in \mathbb{C}^{N \times k}$. Existing choices for the subspace \mathcal{U} include Krylov subspaces [41, 46, 39] or an invariant subspace containing eigenvectors belonging to a few eigenvalues in or close to the region of interest [36, 16]. As we will see later, in Section 4, both approaches often suffer from slow convergence and lack of means to quantify the obtained accuracy. Nevertheless, the projection-based

approaches have been successfully applied to computation of pseudospectral quantities [29, 34], providing a significant improvement over the previous work [10, 19].

In addition to the mentioned computationally oriented approaches, the asymptotic behavior of ε -pseudospectra has been recently studied in [17], while *a priori* estimates for pseudospectra using first-order approximations have been derived in [21].

In this paper we propose a new projection-based approach inspired by the greedy sampling method used in the reduced basis methods [37]. It is primarily designed to provide highly accurate approximation to ε -pseudospectra in isolated parts of the spectrum, that is, regions in the complex plane containing only few eigenvalues of A . As seen in the previous paragraph, for a carefully chosen orthonormal matrix $V \in \mathbb{R}^{N \times k}$, $k \ll N$, $\sigma_{\min}(zV - AV)$ can be used to reconstruct $g(z)$. After a preprocessing step, $\sigma_{\min}(zV - AV)$ can be computed in $\mathcal{O}(k^3)$ operations for any $z \in \mathbb{C}$. If $v(z) \in V$, where $v(z)$ is the smallest right singular vector of $zI - A$, then $\sigma_{\min}(zI - A) = \sigma_{\min}(zV - AV)$ and the reconstruction is exact. Clearly, one can not expect that $v(z) \in \text{span}(V)$ for all $z \in D$, but the goal is to find such V which contains good approximations to $v(z)$ for all $z \in D$. Such V can be efficiently obtained using a greedy sampling driven by error estimates, similarly as for parameter-dependent Hermitian eigenvalue problems [40]. Even though $\sigma_{\min}((x + iy)I - A)$ depends on just two real parameters x and y , approximating it is still very challenging due to the need for high absolute accuracy, in order to get a reasonably good relative accuracy in the vicinity of the eigenvalues. Moreover, the particular structure of the problem allows for additional improvements, such as incorporating the invariant subspace approach for obtaining a good *a priori* basis, and an optimized computation of the lower bounds. Additionally, we use the so-called "saturation assumption" proposed in [23] to reduce the number of error estimates computations. Finally, we also propose an adaptive version of the approach by incorporating a domain splitting technique, which proves to be essential for efficiently solving some of the more challenging numerical examples.

This paper is organized as follows. First, we present our new approach to pseudospectra computation in Section 2, with a detailed discussion on how the pseudospectra bounds are computed, and the sampling strategy and error estimates in Sections 2.1 and 2.2, respectively. In Section 2.3, we present some interpolation results and *a priori* error estimates for the proposed approach, as well as a comparison to linear interpolation of the sampled values. In Section 3, we present an efficient implementation of our approach, with discussions on the implementational details, and the overall complexity in Sections 3.1 and 3.2, respectively. Additionally, in Section 3.3 we present a modified version of our approach which uses a domain splitting procedure. Finally, in Section 4, we present several numerical experiments showing the performance of the proposed approach in comparison to some of the existing methods for the pseudospectra computation, as well as the benefits of incorporating the domain splitting into our approach.

2 Subspace acceleration for pseudospectra computation

In this section, our new projection-based approach for approximating (1) on a domain $D \subset \mathbb{C}$ is presented. Without loss of generality, we assume that D is a rectangle $D = [a, b] + [c, d]i \subset \mathbb{C}$ in the complex plane. Since assessing the resolvent norms on the whole continuous domain D is computationally infeasible, we follow standard practice

in pseudospectra computation [43] and substitute D by a finite, but rather fine, uniformly spaced grid $\Xi \subset D$.

For $z = x + iy \in \mathbb{C}$, the computation of $g(z)$ can be viewed as a Hermitian eigenvalue problem depending on the two real parameters x and y :

$$\begin{aligned} g(x + iy)^2 &= \lambda_{\min}(((x + yi)I - A)^*((x + yi)I - A)) \\ &= \lambda_{\min}(A^*A - x(A + A^*) - yi(A^* - A) + (x^2 + y^2)I) \\ &= \lambda_{\min}(\widehat{A}(x, y)) + x^2 + y^2, \end{aligned} \quad (2)$$

where $\widehat{A}(x, y) = A^*A - x(A + A^*) - yi(A^* - A)$.

Given finite $\mathcal{S} \subset D$, by sampling the ℓ smallest eigenpairs of $\widehat{A}(x, y)$ for each $(x, y) \in \mathcal{S}$, it is possible to compute both an upper bound $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and a lower bound $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ for $\lambda_{\min}(\widehat{A}(x, y))$ for each $(x, y) \in \Xi$.

Given $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$, we can bound $\sigma_{\min}(zI - A)$ using (2) in the following way:

$$\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2 \leq \sigma_{\min}^2(zI - A) \leq \lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2.$$

By taking the square root, the upper bound $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ for $\sigma_{\min}(zI - A)$ can now be defined as

$$\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell) := \sqrt{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2}, \quad (3)$$

while the lower bound $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ is defined by

$$\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell) = \sqrt{\max(\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2, 0)}, \quad (4)$$

keeping in mind the non-negativity of the singular values.

Computation of the bounds $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$, and the choice of appropriate error estimates for driving the sampling procedure is explained in more detail in following Sections 2.1 and 2.2, respectively.

2.1 Computation of $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$

The smallest eigenvalues of a parameter-dependent Hermitian matrix $\widehat{A}(x, y)$ can be efficiently approximated by sampling its smallest eigenpairs for carefully chosen parameter values, as explained in [40]. Here, we present a modified version of the approach exploiting the particular structure of the given problem.

Given a set of samples $\mathcal{S} = \{(x_1, y_1), \dots, (x_M, y_M)\} \subset D$, suppose that for each sample $(x_i, y_i) \in \mathcal{S}$ we have computed the $\ell \geq 1$ smallest eigenvalues

$$\lambda_i = \lambda_i^{(1)} \leq \lambda_i^{(2)} \leq \dots \leq \lambda_i^{(\ell)}$$

of $\widehat{A}(x_i, y_i)$ along with an orthonormal basis of associated eigenvectors $v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(\ell)} \in \mathbb{C}^n$. These eigenvectors are collected into a subspace

$$\mathcal{V}(\mathcal{S}, \ell) := \text{colspan}(\mathcal{V}),$$

where $\mathcal{V} = [\{v_1^{(1)}, \dots, v_1^{(\ell)}, v_2^{(1)}, \dots, v_2^{(\ell)}, \dots, v_M^{(1)}, \dots, v_M^{(\ell)}\}] \in \mathbb{R}^{n \times M\ell}$.

By solving the following $M\ell \times M\ell$ eigenvalue problem

$$V^* \hat{A}(x, y) V w = \lambda_{\mathcal{V}} w, \quad (5)$$

where V denotes an orthonormal basis for $\mathcal{V}(\mathcal{S}, \ell)$, we obtain the smallest $r \leq M\ell$ eigenvalues

$$\lambda_{\mathcal{V}}^{(1)} \leq \lambda_{\mathcal{V}}^{(2)} \leq \dots \leq \lambda_{\mathcal{V}}^{(r)}$$

and the corresponding eigenvectors $w_1, \dots, w_r \in \mathbb{C}^{M\ell}$. By the eigenvalue interlacing property we have

$$\lambda_{\min}(\hat{A}(x, y)) \leq \lambda_{\mathcal{V}}^{(1)},$$

which allows us to define the subspace upper bound for $\lambda_{\min}(\hat{A}(x, y))$ as:

$$\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) := \lambda_{\mathcal{V}}^{(1)}.$$

Following [26], a lower bound $\lambda_{\text{LB}}(x, y; \mathcal{S})$ for $\lambda_{\min}(\hat{A}(x, y))$ can be easily computed as a solution of the following linear program

$$\begin{aligned} \lambda_{\text{LB}}(x, y; \mathcal{S}) &:= \min_{d \in \mathcal{B}} [1, x, y]^T d \\ \text{s.t.} \quad &[1, x_i, y_i]^T d \geq \lambda_i^{(1)}, \quad i = 1, \dots, M, \end{aligned} \quad (6)$$

where $\mathcal{B} \subset \mathbb{R}^3$ is the bounding box for the feasible set given as:

$$\begin{aligned} \mathcal{B} &= [\lambda_{\min}(A^* A), \lambda_{\max}(A^* A)] \times [\lambda_{\min}(A + A^*), \lambda_{\max}(A + A^*)] \\ &\quad \times [\lambda_{\max}(i(A^* - A)), \lambda_{\max}(i(A^* - A))]. \end{aligned} \quad (7)$$

However, in practice, this lower bound is not always a very accurate approximation to $\lambda_{\min}(\hat{A}(x, y))$. By additionally exploiting the structure in U and gaps among the sampled smallest eigenpairs of $\hat{A}(x, y)$ in \mathcal{S} , we can calculate a lower bound $\eta(x, y)$ for Ritz values of $\hat{A}(x, y)$ on U_{\perp}

$$\eta(x, y) \leq \lambda_{\min}(U_{\perp}^* \hat{A}(x, y) U_{\perp}),$$

where $U, U_{\perp} \in \mathbb{C}^{n \times r}$ are orthonormal bases for $\{w_1, \dots, w_r\}$ and its orthogonal complement, respectively. As shown in [40], $\eta(x, y)$ can be computed by simply solving a linear program similar to (6) with updated right-hand side of the constraints, which, in this case, given $\lambda_{\text{LB}}(x, y; \mathcal{S})$ requires solving just one 3×3 linear system.

Combining the information about the Ritz values of $\hat{A}(x, y)$ on U and on U_{\perp} , and using the quadratic residual perturbation bounds [32], allows us to define the subspace lower bound for $\lambda_{\min}(\hat{A}(x, y))$:

$$\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) := \min(\lambda_{\mathcal{V}}^{(1)}, \eta(x, y)) - \frac{2\rho^2}{|\lambda_{\mathcal{V}}^{(1)} - \eta(x, y)| + \sqrt{|\lambda_{\mathcal{V}}^{(1)} - \eta(x, y)|^2 + 4\rho^2}}, \quad (8)$$

with the residual norm $\rho = \|U_{\perp}^* \hat{A}(x, y) U\|_2 = \|\hat{A}(x, y) U - U(U^* \hat{A}(x, y) U)\|_2$.

Remark 2.1. First, it is worth noting that the smallest eigenvectors of $\hat{A}(x, y)$ coincide with the right singular vectors corresponding to $\sigma_{\min}(zI - A)$. Secondly, our subspace-accelerated approach for computing upper bounds $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ can be seen as a special

case of the general projection-based approach for the choice of $\mathcal{U} = \mathcal{V}(\mathcal{S}, \ell)$:

$$\begin{aligned} \min_{u \in \mathcal{V}(\mathcal{S}, \ell)} \|(zI - A)u\|_2 &= \sigma_{\min}((zI - A)V) \\ &= \sqrt{\lambda_{\min}(V^*(zI - A)^*(zI - A)V)} \\ &= \sqrt{\lambda_{\min}(V^*\hat{A}(x, y)V) + x^2 + y^2} = \sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell), \end{aligned}$$

with $z = x + iy$. In fact, in the invariant subspace approach we construct the subspace by sampling the right singular vector corresponding to $\sigma_{\min}(zI - A)$ for $z \in \lambda(A)$ (they coincide with the eigenvectors for the corresponding $z \in \lambda(A)$), while in our approach we generalize this idea by allowing, both, sampling of the smallest singular vectors for $z \notin \lambda(A)$ as well as sampling of more than one smallest singular vector per sampling point.

2.1.1 Bounding box

As explained above, to compute $\eta(x, y)$, we first need to solve (6) and compute $\lambda_{\text{LB}}(x, y; \mathcal{S})$. The role of \mathcal{B} in (6) is to ensure that the solution is finite for all optimization directions in $\{1\} \times [a, b] \times [c, d]$ by bounding the feasible set. However, for the examples considered in Section 4, matrices A^*A , $A + A^*$ and $i(A^* - A)$ often have very small relative gaps between the extremal eigenvalues and the rest of the spectrum, making the eigenproblems in (7) very hard to solve.

Yet, in this specific application, we can avoid the computation of \mathcal{B} . Alternatively, for the linear program (6) to always have finite solutions, it is sufficient to ensure that all possible optimization directions in $\{1\} \times [a, b] \times [c, d]$ can be written as nonnegative linear combination of the constraint directions $(1, x_1, y_1), \dots, (1, x_M, y_M)$. Luckily, this condition can be easily fulfilled by inserting the vertices of D

$$(a, c), (a, d), (b, c), (b, d),$$

a priori into \mathcal{S} , since each point inside a polygon can be written as a convex combination of its vertices. Computationally, this modification reduces the number of full-size eigenvalue problems that need to be solved to four (computing \mathcal{B} required six)

$$\lambda_{\min}(\hat{A}(a, c)), \lambda_{\min}(\hat{A}(a, d)), \lambda_{\min}(\hat{A}(b, c)), \lambda_{\min}(\hat{A}(b, d)).$$

In practice, we also observe that this modification often improves the accuracy of the computed SCM lower bounds $\lambda_{\text{LB}}(x, y; \mathcal{S})$.

2.2 Error estimates and sampling

As described in the previous section, by sampling the smallest eigenpairs of $\lambda_{\min}(\hat{A}(x, y))$ on a set of samples \mathcal{S} , we can compute an upper and a lower bound for $\lambda_{\min}(\hat{A}(x, y))$ on the whole domain D . In our approach, we use a greedy sampling strategy, adding in each iteration to \mathcal{S} a point from Ξ with the largest error estimate. Similarly as in [26, 40], for $z = x + iy \in D$, given $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$, we define the error estimate $\Delta(x, y; \mathcal{S}, \ell)$ for the Hermitian eigenvalue problem $\lambda_{\min}(\hat{A}(x, y) + (x^2 + y^2)I)$ in the

following way:

$$\begin{aligned}\Delta(x, y; \mathcal{S}, \ell) &:= \frac{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2 - \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) - x^2 - y^2}{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2} \\ &= \frac{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)}{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2}\end{aligned}\quad (9)$$

In each iteration, we compute $\Delta(x, y; \mathcal{S}, \ell)$ for all $(x, y) \in \Xi$, and select the one having the largest error estimate as the next parameter sample point.

2.3 Interpolation properties

As shown in [40], the subspace eigenvalue bounds $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ interpolate the exact values of $\lambda_{\min}(\hat{A}(x, y))$:

$$\lambda_{\min}(\hat{A}(x, y)) = \lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) = \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) \quad \forall (x, y) \in \mathcal{S}. \quad (10)$$

Additionally, if $\lambda_{\min}(\hat{A}(x, y))$ is a simple eigenvalue, the subspace bounds also capture the derivatives

$$\nabla \lambda_{\min}(\hat{A}(x, y)) = \nabla \lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) = \nabla \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) \quad \forall (x, y) \in \mathcal{S}, \quad (11)$$

with the gradient ∇ with respect to (x, y) . These interpolation results easily extend to the singular value bounds $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ as can be seen from the following theorem.

Theorem 2.2. *For $z = x + iy \in \mathcal{S}$, the singular value bounds $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$, defined in (3) and (4), respectively, satisfy*

$$\sigma_{\min}(zI - A) = \sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell) = \sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell).$$

Additionally, if $\sigma_{\min}(zI - A)$ is simple and positive, then

$$\nabla \sigma_{\min}(zI - A) = \nabla \sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell) = \nabla \sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell).$$

Proof. The first equality follows directly from (10) by taking the square root. Since $\sigma_{\min}(zI - A) > 0$, by differentiating (2) we get

$$\nabla \sigma_{\min}(zI - A) = \frac{1}{2\sigma_{\min}(zI - A)} \left(\nabla \lambda_{\min}(\hat{A}(x, y)) + \begin{bmatrix} 2x \\ 2y \end{bmatrix} \right).$$

Simplicity of $\sigma_{\min}(zI - A)$ implies (11), which together with the first equality, gives the second equality. \square

Using Theorem 2.2, we obtain *a priori* error estimates for $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$.

Theorem 2.3. *Let $z_{\mathcal{S}} = x_{\mathcal{S}} + iy_{\mathcal{S}}$ such that $\sigma_{\min}(z_{\mathcal{S}}I - A)$ is simple and positive and let $h > 0$ such that $\sigma_{\min}(zI - A)$, $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ are twice differentiable on $B(z_{\mathcal{S}}, h)$. Then there exist constants $C_1, C_2 > 0$ such that*

$$\begin{aligned}|\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \sigma_{\min}(zI - A)| &< C_1 h^2 \\ |\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell) - \sigma_{\min}(zI - A)| &< C_2 h^2,\end{aligned}$$

for all $z = x + iy \in B(z_{\mathcal{S}}, h)$.

Proof. Let $z = x + iy \in B(z_S, h)$. Expanding $\sigma_{\min}(zI - A)$ and $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ around z_S using a second-order Taylor polynomial expansion and using the results of Theorem 2.3, we obtain

$$\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \sigma_{\min}(zI - A) = \frac{(z - z_S)^2}{2} (\nabla^2 \sigma_{\min}(z_S I - A) - \nabla^2 \sigma_{\text{SUB}}(x_S, y_S; \mathcal{S}, \ell)).$$

for $z_1, z_2 = x_2 + iy_2 \in [z_S, z]$. The first inequality now holds for

$$C_1 = \max_{\tilde{z} \in B(z, h)} \|\nabla^2 \sigma_{\min}(\tilde{z}I - A)\|_2 + \max_{\tilde{z} = \tilde{x} + i\tilde{y} \in B(z, h)} \|\nabla^2 \sigma_{\text{SUB}}(\tilde{x}, \tilde{y}; \mathcal{S}, \ell)\|_2.$$

The second inequality can be shown in the same way. \square

Remark 2.4. To ensure the differentiability conditions on $\sigma_{\min}(zI - A)$ and $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ needed in the assumptions of Theorem 2.3, it is sufficient that the smallest singular values $\sigma_{\min}(zI - A)$ and $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ stay simple and positive on $B(z, h)$, see [27]. A simple criterion for differentiability of $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ is not available, since (8) involves $\eta(x, y)$, which depends on the solution of the linear program $\lambda_{\text{LB}}(x, y; \mathcal{S})$, which is not necessarily smooth around (x_i, y_i) .

Remark 2.5. The requirement for positivity of σ_{\min} in Theorems 2.2 and 2.3 is artificial and can be fixed by using "signed" singular values as in the case of the analytic SVD [9].

In practice, since $\hat{A}(x, y)$ is an analytic function in x and y , we can expect much faster convergence than the one guaranteed by Theorem 2.3, see [40, Section 3.4.1] and [2, Section 2.3.2]. Numerical experiments shown in Section 4 support this.

Additionally, in the following theorem, we show that using the subspace lower bounds $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ for approximating $\lambda_{\min}(\hat{A}(x, y))$ is always at least as good as linearly interpolating the computed values of $\lambda_{\min}(\hat{A}(x, y))$.

Theorem 2.6. Suppose we are given \mathcal{S} , defined as above, and $(x, y) \in \text{conv}(\mathcal{S})$. Let $1 \leq i < j < k \leq M$ such that $(x, y) \in \text{conv}\{(x_i, y_i), (x_j, y_j), (x_k, y_k)\}$. We define $l_{ijk}(x, y) : \mathbb{R}^2 \rightarrow \mathbb{R}$ to be the linear function interpolating $\lambda_{\min}(A(x, y))$ in (x_i, y_i) , (x_j, y_j) and (x_k, y_k) . Then we have

$$l_{ijk}(x, y) \leq \lambda_{\text{LB}}(x, y; \mathcal{S}),$$

where $\lambda_{\text{LB}}(x, y; \mathcal{S})$ is defined as in (6). Additionally, there exist $1 \leq \tilde{i} < \tilde{j} < \tilde{k} \leq M$ such that $(x, y) \in \text{conv}\{(x_{\tilde{i}}, y_{\tilde{i}}), (x_{\tilde{j}}, y_{\tilde{j}}), (x_{\tilde{k}}, y_{\tilde{k}})\}$ and $l_{\tilde{i}\tilde{j}\tilde{k}}(x, y) = \lambda_{\text{LB}}(x, y; \mathcal{S})$.

Proof. The dual problem of (6) is:

$$\begin{aligned} \lambda_{\text{LB}}(x, y; \mathcal{S}) &= \max_{\substack{z^T A = [1, x, y]^T, \\ z \geq 0}} [\lambda_1, \dots, \lambda_M]^T z, \end{aligned} \quad (12)$$

We can interpret (12) as an optimization problem over all possible representations of (x, y) as a convex combination of the points in \mathcal{S} . Barycentric coordinates of (x, y) on the triangle spanned by (x_i, y_i) , (x_j, y_j) and (x_k, y_k) clearly provide an admissible point of (12), immediately proving $l_{ijk}(x, y) \leq \lambda_{\text{LB}}(x, y; \mathcal{S})$.

Moreover, there is an optimal point z for (12) such that z has only three non-zero coordinates, since each non-zero coordinate of z corresponds to one the active constraints in the optimal solution of the primal problem. This immediately gives that there exist $1 \leq \tilde{i} < \tilde{j} < \tilde{k} \leq M$ such that $l_{\tilde{i}\tilde{j}\tilde{k}}(x, y) = \lambda_{\text{LB}}(x, y; \mathcal{S})$. \square

As shown in [40], we have $\lambda_{\text{LB}}(x, y; \mathcal{S}) \leq \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) \leq \lambda_{\min}(\hat{A}(x, y))$. Combining this with the results of Theorem 2.6, we get

$$l_{ijk}(x, y) \leq \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) \leq \lambda_{\min}(\hat{A}(x, y)),$$

for all $(x, y) \in \text{conv}(\mathcal{S})$ and (i, j, k) such that (x, y) is inside the triangle spanned by (x_i, y_i) , (x_j, y_j) and (x_k, y_k) . This shows that using the subspace lower bounds $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ for approximating $\lambda_{\min}(\hat{A}(x, y))$ is always at least as good as linearly *interpolating* the computed values of $\lambda_{\min}(\hat{A}(x, y))$.

3 Algorithm and computational details

In this section we present a summary of our subspace-accelerated approach for pseudospectra computation, in form of Algorithm 1, introduced in Section 2 and discuss its implementation and computational complexity in Sections 3.1 and 3.2, respectively. Finally, we present an adaptive version of the approach in Section 3.3.

3.1 Implementation details

The efficient implementation of our proposed approach for computing upper and lower bounds for $\sigma_{\min}(zI - A)$ requires care in order to avoid unnecessary computations. Some implementation details are discussed in the following.

Initialization of the sample \mathcal{S} . As explained in Section 2.1, we initialize \mathcal{S} to contain the vertices of the domain D :

$$\mathcal{S} = \{(a, c), (a, d), (b, c), (b, d)\}.$$

For certain problems, it makes sense to *a priori* add additional points from D to \mathcal{S} . To make the error estimates (9) sufficiently small, we require high absolute accuracy in regions around the eigenvalues of A . In numerical experiments we observe that \mathcal{S} eventually contains many points very close to the exact eigenvalues of A . We use this observation, and combine our approach with the invariant subspace approach, to "warm start" the algorithm by inserting eigenvalues of A inside D into the initial sample. In practice, this is usually enough to ensure high absolute accuracy in the proximity of the eigenvalues of A . Such eigenvalues of A can be efficiently computed by simply computing the eigenvalues closest to the centre of D . However, in order not to make the sample \mathcal{S} too large, we limit the number of the exact eigenvalues included in \mathcal{S} to 20 closest to the center of D , unless stated otherwise.

Choice of ℓ . Using a larger value of ℓ , number of the smallest eigenvectors included in $\mathcal{V}(\mathcal{S}, \ell)$ per sample point, leads to better bounds, but on the other hand, it increases the computational cost. Intuitively, given eigenvalue gaps between few smallest eigenvalues, ℓ should be chosen to maximize the eigenvalue gap $\lambda_i^{(\ell+1)} - \lambda_i^{(\ell)}$. However, in the absence of a priori information on eigenvalue gaps, this is not possible. In our implementation, we have used $\ell = 6$ for all $(x, y) \in D$, since this usually ensured that the eigenvalue gap $\lambda_i^{(\ell+1)} - \lambda_i^{(1)}$ is sufficiently large for our approach to provide satisfying convergence.

Computation of $\lambda_{\min}(\hat{A}(x, y))$. As can be seen in (2), computing the smallest eigenpairs of $\hat{A}(x, y)$ is equivalent to computing the smallest singular values and associated singular vectors of the matrix $zI - A$. However, numerically this is not equivalent. When computing $\lambda_{\min}(\hat{A}(x, y))$ directly, we are working with a matrix of squared condition number. To avoid that, we solve the singular value problem instead, by computing the smallest eigenpairs of the extended matrix $\begin{bmatrix} 0 & zI - A \\ (zI - A)^* & 0 \end{bmatrix}$ using the inverse Lanczos method. For a dense matrix A , this can be made more efficient by first computing the Schur decomposition of $A = QTQ^T$, see [33, 43], since $\sigma_{\min}(Q(zI - T)Q^T) = \sigma_{\min}(zI - T)$. In this case, each iteration of the inverse Lanczos method requires solving just two triangular linear systems. For a large-scale sparse matrix A , the inverse Lanczos method can be made more computationally efficient by first computing the sparse LU factorization of $zI - A$. We assume this method to be accurate and efficient for all $(x, y) \in \Xi$.

Computation of $\lambda_{\min}(\hat{A}(x, y))$ for $x + iy \in \sigma(A)$. As mentioned above, we "warm start" our approach by initializing \mathcal{S} to contain the exact eigenvalues of A inside D . However, for $z = x + iy \in \sigma(A)$, the inverse Lanczos method can not be directly applied since $(zI - A)^{-1}$ is not defined. Knowing that the smallest singular value is 0, it is possible to extract the non-singular part of $zI - A$, by deflating the directions of the smallest singular vectors, and compute the subsequent singular values and vectors.

Without loss of generality, we can assume that $x = y = 0$. Furthermore, we assume that zero is a simple eigenvalue of A . Suppose v_1 and u_1 are the left and the right singular vectors of A corresponding to the singular value zero:

$$v_1^* A = 0, \quad A u_1 = 0. \quad (13)$$

We know that the Lanczos method will converge to the second largest eigenvector, if the starting vector is orthogonal to the dominant eigenvector. Thus, when computing the subsequent left (right) smallest singular vectors using the inverse Lanczos method, we need to choose an initial starting vector which is orthogonal to v_1 (u_1). However, in order to successfully apply the inverse Lanczos method to this setting, we need to be able to efficiently solve the following linear systems

$$\text{for a given } v \in \{v_1\}^\perp \quad \text{find } u \in \{u_1\}^\perp \quad \text{s.t.} \quad A u = v, \quad (14)$$

$$\text{for a given } u \in \{u_1\}^\perp \quad \text{find } v \in \{v_1\}^\perp \quad \text{s.t.} \quad A^* v = u, \quad (15)$$

which are similar to the correction equation in Jacobi-Davidson SVD [25].

We use the idea of bordered linear systems (see e.g. [28, 18]) for solving (14) and (15), and solve the following linear systems instead:

$$\text{for a given } v \in \{v_1\}^\perp \quad \text{find } u \in \{u_1\}^\perp \quad \text{s.t.} \quad \begin{bmatrix} A^T & v_1 \\ u_1^T & 0 \end{bmatrix} \begin{bmatrix} u \\ \mu \end{bmatrix} = \begin{bmatrix} v \\ 0 \end{bmatrix}, \quad (16)$$

$$\text{for a given } u \in \{u_1\}^\perp \quad \text{find } v \in \{v_1\}^\perp \quad \text{s.t.} \quad \begin{bmatrix} A & u_1 \\ v_1^T & 0 \end{bmatrix} \begin{bmatrix} v \\ \lambda \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix}. \quad (17)$$

It can be easily shown that, under these conditions, the bordered matrices are nonsingular and that solving the bordered linear systems yields $\lambda = \mu = 0$ and, thus, the same solution as (14) and (15). Moreover, it is important to note that bordering A , as in (16) and (17), does not "destroy" its matrix envelope and thus the sparse LU factors of the bordered matrices have approximately the same number of nonzero elements as those of A .

Computation of $V^*\hat{A}(x, y)V$. By the affine linear decomposition of $\hat{A}(x, y)$ (2), we have

$$V^*\hat{A}(x, y)V = V^*(A^*A)V - xV^*(A + A^*)V - yV^*i(A^* - A)V.$$

Following a standard technique in reduced basis methods [37], we precompute and store matrices $V^*(A^*A)V$, $V^*(A + A^*)V$, $V^*i(A^* - A)V$, and update them as new columns are added to V . This allows computation of $V^*\hat{A}(x, y)V$ at a negligible cost as long as $M\ell \ll N$.

Choice of r . The subspace lower bounds $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ clearly depend on the choice of r , number of the smallest Ritz vectors used in the construction of the subspace U . As explained in [40, Remark 3.3], r is chosen adaptively for each $(x, y) \in \Xi$ by taking the maximal value of $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ among a few small values of $r = 0, 1, 2, \dots$.

Computation of the residual norm ρ . Efficient and accurate computation of the residual ρ is a very important for the accuracy of the lower bounds. The application of the technique used in [40] requires precomputation of matrices in the affine linear expansion of $V^*\hat{A}(x, y)^*\hat{A}(x, y)V$, one of which is $V^*(A^*A)^*A^*AV$. We can expect $V^*(A^*A)^*A^*AV$ to be extremely ill-conditioned even for moderate $\kappa(A)$. To avoid this, we pay a slightly higher price and compute in each iteration the QR decomposition of the following $n \times 4M\ell$ matrix

$$QR = [A^*AV, (A + A^*)V, i(A^* - A)V, V].$$

For any $(x, y) \in D$, this allows computation ρ by solving the following small $4M\ell \times r$ singular value problem

$$\begin{aligned} \rho &= \|U_\perp^* A(x, y)U\|_2 = \|A(x, y)U - U(U^* A(x, y)U)\|_2 \\ &= \|A(x, y)VW - VW\Lambda\|_2 \\ &= \|[A^*AV, (A + A^*)V, i(A^* - A)V, V][W^T, -xW^T, -yW^T, -\Lambda W^T]^T\|_2 \\ &= \|R[W^T, -xW^T, -yW^T, -\Lambda W^T]^T\|_2, \end{aligned} \tag{18}$$

where $W \in \mathbb{R}^{M\ell \times r}$ such that $U = VW$ and $\Lambda = \text{diag}(\lambda_{\mathcal{V}}^{(1)}, \lambda_{\mathcal{V}}^{(2)}, \dots, \lambda_{\mathcal{V}}^{(r)})$.

Updating of $\lambda_{\text{LB}}(A(x, y))$. Computationally the most expensive part of computing $\lambda_{\text{SLB}}(A(x, y))$ is solving (6). In [40], the interior point method is proposed for solving (6). However, for this specific application, the simplex method proves to be far superior, since the linear program (6) has just three variables. Additionally, as we incrementally build (6), the simplex method, unlike the interior-point method, allows us to take advantage of previously computed solutions and just slightly update them to compute the new ones. For example, if the newly added

constraints do not cut off the previously optimal vertex, it will stop immediately. In practice, we observe that this modification significantly reduces the computational time.

Computation of the new sample point (x_{M+1}, y_{M+1}) . The next parameter sample μ_{M+1} is computed as the maximizer of the error estimate (9) on Ξ . In every iteration, this requires recomputing of $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ on whole Ξ , which can become computationally quite expensive. Instead, as explained in [23], we can optimize the search for (x_{M+1}, y_{M+1}) by using the error estimates from the previous iteration. As $M \rightarrow \infty$, the error estimates (9) converge to 0. Even though the convergence is not monotonic, it is reasonable to assume what is known as the saturation assumption, which in the current setting takes the following form: there exists $C_{\text{sat}} > 0$ such that

$$\Delta(x, y; \mathcal{S}^*, \ell) < C_{\text{sat}} \Delta(x, y; \mathcal{S}, \ell), \quad \forall \mathcal{S}^* \supset \mathcal{S}, \quad \forall (x, y) \in \Xi.$$

We assume that the elements in Ξ are sorted descendingly according to the error estimates (9) from the previous iteration, and look for (x_{M+1}, y_{M+1}) by iterating over Ξ . We sequentially recompute the bounds $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ and keep track of the current maximum error estimate Δ_{max} as well as the point $(x_{\text{max}}, y_{\text{max}}) \in \Xi$ where it was attained. Reaching a point $(x, y) \in \Xi$ such that $C_{\text{sat}} \Delta(x, y; \mathcal{S}, \ell) < \Delta_{\text{max}}$, allows us to skip all the remaining elements of Ξ , since the saturation assumption ensures that their error estimates will be smaller than Δ_{max} .

Stopping criterion. Given the prescribed tolerance $\varepsilon_{\text{tol}} > 0$, we stop the execution of Algorithm 1 when

$$\max_{(x, y) \in \Xi} \Delta(x, y; \mathcal{S}, \ell) = \max_{(x, y) \in \Xi} \frac{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)}{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2} < \varepsilon_{\text{tol}}. \quad (19)$$

However, for $(x, y) \in \Xi$ close to an eigenvalue of A , fulfilling (19) requires the absolute error $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ to be very small which can not always be attained due to inexact computation of $\lambda_{\text{min}}(\tilde{A}(x, y))$. To circumvent this issue, we additionally prescribe an absolute tolerance $\varepsilon_{\text{abs}} > 0$ and for points $(x, y) \in \Xi$ satisfying either

$$\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) < \varepsilon_{\text{abs}}$$

or

$$\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2 < \varepsilon_{\text{abs}},$$

and assuming that $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ already is a very good approximation to $\sigma_{\text{min}}((x + iy)I - A)$, we set $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ to the value of $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$.

3.2 Algorithm and computational complexity.

Algorithm 1 summarizes our proposed approach explained in Section 2, taking into account implementational details from Section 3.1. The algorithm requires solution of M singular value problems of size $n \times n$ for computing the exact smallest singular values

and vectors of $zI - A$, one for each $z \in \mathcal{S}$. Computing $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ and $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ for all $(x, y) \in \Xi$ in every iteration amounts to solving at most $M|\Xi|$ LP problems with 3 variables and up to M constraints, as well as at most $M|\Xi|$ eigenproblems of size at most $M\ell \times M\ell$. As long as $M\ell \ll n$, these parts will be negligible, and the computational cost of Algorithms 1 will be dominated by the cost of computing the exact singular values and vectors. Moreover, as explained in Section 3.1, by assuming the saturation assumption, we do not have to recompute the bounds $\lambda_{\text{SUB}}(\mu; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(\mu; \mathcal{S}, \ell)$ for all $(x, y) \in \Xi$ in every iteration. In practice, the bounds for specific $(x, y) \in \Xi$ are recomputed only a few times throughout the iterations.

Algorithm 1 Reduced Basis approach for pseudospectra computation

Input: $A \in \mathbb{C}^{n \times n}$, uniformly spaced grid Ξ on $D = [a, b] + [c, d]i \subset \mathbb{C}$, number of sampled eigenpairs per sample point ℓ , relative error tolerance ε_{tol} .

Output: Sample set $\mathcal{S} \subset D$ with the corresponding eigenvalues $\lambda^{(j)}_i$ and an eigenvector basis V for $\mathcal{V}(\mathcal{S}, \ell)$ such that $\frac{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)}{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) + x^2 + y^2} < \varepsilon_{\text{tol}}$ for every $(x, y) \in \Xi$.

- 1: Initialize the sample set $\mathcal{S} = \{(a, c), (a, d), (b, c), (b, d)\} \cup \sigma(A) \cap D$.
 - 2: Compute the ℓ smallest eigenpairs of $\hat{A}(x, y)$, for all $(x, y) \in \mathcal{S}$.
 - 3: Compute an orthonormal basis V for $\mathcal{V}(\mathcal{S}, \ell)$, matrices in the affine linear expansion of $V^* \hat{A}(x, y) V$ and R .
 - 4: Compute $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ for all $(x, y) \in \Xi$.
 - 5: $(x_{\text{max}}, y_{\text{max}}) \leftarrow \arg \max_{(x, y) \in \Xi} \Delta(x, y; \mathcal{S}, \ell)$.
 - 6: **while** $\Delta(x_{\text{max}}, y_{\text{max}}; \mathcal{S}, \ell) > \varepsilon_{\text{tol}}$ **do**
 - 7: $\mathcal{S} \leftarrow \mathcal{S} \cup \{(x_{\text{max}}, y_{\text{max}})\}$.
 - 8: Compute the ℓ smallest eigenpairs of $\hat{A}(x_{\text{max}}, y_{\text{max}})$.
 - 9: Update the orthonormal basis V for $\mathcal{V}(\mathcal{S}, \ell)$, matrices in the affine linear expansion of $V^* \hat{A}(x, y) V$ and recompute R .
 - 10: **for** $(x, y) \in \Xi$ **do**
 - 11: **if** $C_{\text{sat}} \Delta(x, y; \mathcal{S}, \ell) < \Delta_{\text{max}}$ **then**
 - 12: Exit the for loop.
 - 13: **end if**
 - 14: Recompute $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) = \lambda_{\min}(V^* \hat{A}(x, y) V)$.
 - 15: Recompute the residual norm ρ according to (18).
 - 16: Recompute $\lambda_{\text{LB}}(x, y; \mathcal{S})$ according to (6).
 - 17: Recompute $\eta(x, y)$.
 - 18: Recompute $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ according to (8).
 - 19: Compute $\Delta(x, y; \mathcal{S}, \ell)$ according to (9) and update Δ_{max} and $(x_{\text{max}}, y_{\text{max}})$.
 - 20: **end for**
 - 21: **end while**
 - 22: Compute $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\sigma_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ for all $(x, y) \in \Xi$.
-

3.3 Domain partitioning

As explained in the previous section, as long as $M\ell \ll n$, the computational cost of Algorithms 1 will be dominated by the cost of computing the exact singular values and vectors. However, for some of the examples considered in Section 4, this was not the case and, as M increased, the computational time needed for recomputing the bounds

$\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ became prevalent. Similar problems have already appeared in the reduced basis framework for linear system, where they have been addressed using domain splitting techniques, see e.g. [14, 20]. In the following, we discuss how a similar idea can be incorporated into Algorithm 1 to make it more adaptive, and present a domain splitting procedure for it in the form of Algorithm 2.

While executing Algorithm 1, we can perform a domain splitting whenever $M\ell$, the number of columns in V , becomes too large, say larger than a certain constant `maxsize`. In other words, we can split our problem into two subproblems, on the subdomains D_1 and D_2 as well as their corresponding subgrids Ξ_1 and Ξ_2 , respectively, and start to run Algorithm 1 on each of the subdomains separately in parallel. When performing the domain splitting, ideally, we would like to reuse the already sampled smallest eigenvectors of $\hat{A}(x, y)$ on D for solving the subproblems on D_1 and D_2 , which is why the samples \mathcal{S}_1 and \mathcal{S}_2 are initialized with elements from \mathcal{S} . We can further recursively apply the same splitting rule on each of the subproblems in case their orthonormal bases V_1 and V_2 also become too large.

Some aspects of an efficient implementation of such domain splitting are discussed in the following description, while the choice of the constant `maxsize` is discussed in more detail in Remark 3.1.

Splitting of D and Ξ . The rectangular domain D is split into half-rectangles D_1 and D_2 , so that the longer side of (the rectangle) D is split in two. The grid Ξ is split into the subgrids Ξ_1 and Ξ_2 accordingly.

Splitting of \mathcal{S} . As discussed above, we would like to reuse the already sampled smallest eigenpairs of $\hat{A}(x, y)$ on D , which is why the samples \mathcal{S}_1 and \mathcal{S}_2 are initialized as

$$\mathcal{S}_1 = \{(x, y) \in \mathcal{S} : \text{dist}((x, y), D_1) < \varepsilon_{\text{domain}}\} \quad (20)$$

$$\mathcal{S}_2 = \{(x, y) \in \mathcal{S} : \text{dist}((x, y), D_2) < \varepsilon_{\text{domain}}\} \quad (21)$$

Additionally, in order to make the approach consistent with Algorithm 1 on each of the subdomains, we insert in both \mathcal{S}_1 and \mathcal{S}_2 the eigenvalues of A on D_1 and D_2 , respectively, which might have been missed when computing $\sigma(A) \cap D$ due to the set maximal number of eigenvalues. Furthermore, as discussed in Section 2.1, in order to ensure the existence of the solution for the linear program (6) on each of the subdomains, we insert in both \mathcal{S}_1 and \mathcal{S}_2 the common vertices of the rectangles D_1 and D_2 .

Computation of $\sigma(A) \cap D_1$ and $\sigma(A) \cap D_2$. Some of the eigenvalues of A inside D_1 and D_2 have already been computed when computing $\sigma(A) \cap D$, which is why their eigenspaces can be deflated when computing $\sigma(A) \cap D_1$ and $\sigma(A) \cap D_2$, resulting in an accelerated convergence of the eigensolver.

Computation of V_1 , V_2 , $V_1^* \hat{A}(x, y) V_1$ and $V_2^* \hat{A}(x, y) V_2$. If orthonormal bases V_1 and V_2 , for $\mathcal{V}(\mathcal{S}_1, \ell)$ and $\mathcal{V}(\mathcal{S}_2, \ell)$, respectively, were computed from scratch, then computing $V_1^* \hat{A}(x, y) V_1$ and $V_2^* \hat{A}(x, y) V_2$ would require multiplication of $2Q$ $n \times n$ matrices with $\mathcal{O}(M\ell)$ vectors, which can become quite costly as M increases. Instead, this can be made efficient by exploiting the already computed quantities. Let $\mathcal{V} = VR$ be a QR decomposition, with $R \in \mathbb{R}^{M\ell \times M\ell}$ and let $P_1 \in \mathbb{R}^{M\ell \times M\ell}$ be

a block-diagonal matrix defined as

$$P_1 = \text{diag}(c_1 I_\ell, \dots, c_M I_\ell),$$

where $c_i = 1$ if $(x_i, y_i) \in \mathcal{S}_1$, and zero otherwise, for all $i = 1, \dots, M$. Thus, we have

$$\mathcal{V}_1 = \mathcal{V}P_1 = V(RP_1),$$

which allows us to compute V_1 by first computing a QR decomposition of the matrix $RP_1 = Q_{P_1} R_{P_1}$, and then multiplying V with Q_{P_1} . Furthermore, the matrices in the expansion of $V_1^* \hat{A}(x, y) V_1$ can also be efficiently computed in a complexity independent of n , since $V_1^* \hat{A}(x, y) V_1 = Q_{P_1}^T (V^* \hat{A}(x, y) V) Q_{P_1}$.

Remark 3.1. *It is not a priori clear what is the critical value of M (or more precisely $M\ell$) when the subspace-accelerated approach becomes more expensive than a direct grid-based approach, where for each $(x, y) \in \Xi$ the eigenvalue problem would be solved exactly. This depends on a number of different factors such as:*

- *the ratio between the computational times needed for computing $\lambda_{\min}(\hat{A}(x, y))$ and solving a small dense eigenvalue problem $\lambda_{\min}(V^* \hat{A}(x, y) V)$ for a single value of μ ,*
- *size of the training set Ξ ,*
- *computational savings due to the saturation assumption (3.1).*

In our implementation we have used `maxsize` = 150.

4 Numerical experiments

In this section, we report on the performance of our proposed approaches for a number of large-scale examples available in the literature. Algorithms 1 and 2 have been implemented in MATLAB Version 7.14.0.739 (R2012a) and all experiments have been performed on an Intel Xeon CPU E31225 with 4 cores, 3.1 GHz, and 8 GB RAM.

When implementing and testing Algorithms 1 and 2, we have made the following choices. Unless stated otherwise, we set the error tolerance ε_{tol} to 0.1, the absolute tolerance ε_{abs} to 10^{-8} , the maximum number of sampled points to $M_{\text{max}} = 100$ and Ξ to be 100×100 uniformly spaced grid on D . The smallest singular values and the corresponding singular vectors of $zI - A$ have been computed, as explained in Section 3.1, using the MATLAB built-in function `eigs`, which is based on ARPACK [31], with the tolerance set to 10^{-10} . For solving the linear program (6), we have used MOSEK 7 Matlab toolbox [3] implementation of the simplex method with updating. In all experiments, we have used Algorithm 1 with the number of smallest eigenpairs included in $\mathcal{V}(\mathcal{S}, \ell)$ set to $\ell = 6$. In the first three iterations we have worked with the saturation constant set to $C_{\text{sat}} = +\infty$ and $C_{\text{sat}} = 1$ in the following iterations. For choosing r in Section 2, we have tested all values $r = 0, 1, \dots, 3\ell$, as explained in Section 3.1.

Algorithm 2 Domain splitting procedure for Algorithm 2

Input: Rectangular domain $D \subset \mathbb{C}$, a uniformly spaced grid Ξ on D , the sample set \mathcal{S} , sampled smallest eigenpairs at \mathcal{S} the orthonormal basis $V \in \mathbb{R}^{n \times M\ell}$ $\mathcal{V}(\mathcal{S}, \ell)$, and the projected matrices in the expansion of $V^* \hat{A}(x, y) V$.

Output: Two subdomains D_1 and D_2 , s.t. $D = D_1 \cup D_2$ and $D_1 \cap D_2 = \emptyset$, the corresponding grids Ξ_1 and Ξ_2 , the samples \mathcal{S}_1 and \mathcal{S}_2 , the orthonormal basis V_1 and V_2 , and the matrices in the expansion of $V_1^* \hat{A}(x, y) V_1$ and $V_2^* \hat{A}(x, y) V_2$.

- 1: Split the domain D into D_1 and D_2 by halving the longer side of the rectangle.
 - 2: Split Ξ into Ξ_1 and Ξ_2 accordingly.
 - 3: Split the sample \mathcal{S} into \mathcal{S}_1 and \mathcal{S}_2 according to (20) and (21).
 - 4: Compute the orthonormal basis for subdomains V_1 and V_2 from V .
 - 5: Compute matrices in the expansion of $V_1^* \hat{A}(x, y) V_1$ and $V_2^* \hat{A}(x, y) V_2$.
 - 6: **for** each shared vertex (\tilde{x}, \tilde{y}) between D_1 and D_2 **do**
 - 7: Compute the ℓ smallest eigenpairs of $\hat{A}(\tilde{x}, \tilde{y})$.
 - 8: $\mathcal{S}_1 \leftarrow \mathcal{S}_1 \cup (\tilde{x}, \tilde{y})$ and $\mathcal{S}_2 \leftarrow \mathcal{S}_2 \cup (\tilde{x}, \tilde{y})$
 - 9: Update both orthonormal basis V_1 and V_2 as well as the matrices in the affine linear expansion of $V_1^* \hat{A}(x, y) V_1$ and $V_2^* \hat{A}(x, y) V_2$.
 - 10: **end for**
 - 11: $\mathcal{S}_1 \leftarrow \mathcal{S}_1 \cup (\sigma(A) \cap D_1)$ and $\mathcal{S}_2 \leftarrow \mathcal{S}_2 \cup (\sigma(A) \cap D_2)$.
 - 12: Compute the ℓ smallest eigenpairs of $\hat{A}(x, y)$, for all new (x, y) in \mathcal{S}_1 and \mathcal{S}_2 .
 - 13: Update both orthonormal basis V_1 and V_2 as well as the matrices in the affine linear expansion of $V_1^* \hat{A}(x, y) V_1$ and $V_2^* \hat{A}(x, y) V_2$.
-

4.1 Comparison with other approaches

As can be seen in Examples 4.1 – 4.6, in terms of the computational time, Algorithm 1 is significantly faster than the grid-based approach, while providing satisfying accuracy.

Additionally, we compare performance of Algorithm 1 against two other projection-based approaches, namely the Krylov subspace approach and the invariant subspace approach. On a smaller 30×30 uniformly spaced grid $\tilde{\Xi}$, we compute the exact smallest singular values $\sigma_{\min}(zI - A)$, as well as the approximations $\sigma_{\text{kry}}(x, y; k)$ and $\sigma_{\text{inv}}(x, y; k)$ for few values of $k \in \mathbb{N}$, where

$$\begin{aligned} \sigma_{\text{kry}}(x, y; k) &= \sigma_{\min}(zU_k^{\text{kry}} - AU_k^{\text{kry}}) \\ \sigma_{\text{inv}}(x, y; k) &= \sigma_{\min}(zU_k^{\text{inv}} - AU_k^{\text{inv}}), \end{aligned}$$

where U_k^{arn} and U_k^{inv} denote the k -dimensional Krylov subspace of matrix A and the k -dimensional invariant subspace spanned by the eigenvectors corresponding to eigenvalues closest to D , respectively. In Figures 2c – 7c, we present the convergence rates towards the exact values of $\sigma_{\min}((x + iy)I - A)$:

$$\max_{(x, y) \in \tilde{\Xi}} \frac{\sigma_{\text{kry}}(x, y; k)^2 - \sigma_{\min}((x + iy)I - A)^2}{\sigma_{\min}((x + iy)I - A)^2} \quad (22)$$

$$\max_{(x, y) \in \tilde{\Xi}} \frac{\sigma_{\text{inv}}(x, y; k)^2 - \sigma_{\min}((x + iy)I - A)^2}{\sigma_{\min}((x + iy)I - A)^2} \quad (23)$$

w.r.t. the subspace size k and compare them to the corresponding convergence rates for

the computed subspace bounds $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ and $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$:

$$\max_{(x, y) \in \tilde{\Xi}} \frac{\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell) - \sigma_{\min}((x + iy)I - A)^2}{\sigma_{\min}((x + iy)I - A)^2} \quad (24)$$

$$\max_{(x, y) \in \tilde{\Xi}} \frac{\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell) - \sigma_{\min}((x + iy)I - A)^2}{\sigma_{\min}((x + iy)I - A)^2} \quad (25)$$

w.r.t. the dimensionality of the subspace $\mathcal{V}(\mathcal{S}, \ell)$. We observe that the convergence of $\sigma_{\text{kry}}(x, y; k)^2, \sigma_{\text{inv}}(x, y; k)^2$ usually flattens after first few iterations, while the subspace upper bounds $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ provide a very accurate approximation to $\sigma_{\min}((x + iy)I - A)^2$ after only a few iterations. The corresponding relative error (24) is often very small already at the beginning due to the "warm starting" and this very fast convergence to the exact values could be used as a motivation for deriving a heuristic version of our approach. Eventually, when Algorithm 1 finishes, we usually observe that even the subspace lower bounds $\lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell)$ provide a more accurate approximation than the other two approaches.

4.1.1 Dense matrices

We first consider two moderately sized dense matrices A ($n \leq 5000$), such that it is still possible to compute their Schur decomposition $A = QTQ^*$. We compute approximate pseudospectra $\sigma_{\varepsilon}(T)$, and compare the results obtained using Algorithm 1 and other approaches for pseudospectra computation. For more details, see Examples 4.1 and 4.2.

Example 4.1. We consider the example `random_demo.m` from *EigTool* [45], where $A \in \mathbb{R}^{n \times n}$ is a random matrix whose entries are drawn from the normal distribution with mean 0 and variance $1/N$. As $N \rightarrow \infty$, spectral abscissa of A converges to 1. We choose $N = 2000$ and set $D = [0.95, 1.05] + [-0.05, 0.05]i$ to be a region in the complex plane around the right-most part of the spectrum. The observed matrix A has four eigenvalues inside D . The spectrum of A (blue dots) in the region around D (red square) is shown in Figure 2a, whereas in Figure 2b we can see the convergence of the maximum relative error estimate (9) in Algorithm 1 w.r.t. iteration. The Algorithm 1 reaches the desired tolerance in 26 iterations with the computational time of 1209 seconds, while the exact computation using a grid-based approach would take around 22000 seconds. In Figure 2e we see the computed ε -pseudospectra for $\varepsilon = 10^{-1}, 10^{-2}$, while the surface plot of $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ is presented in Figure 2d. We see that with the prescribed relative error tolerance $\varepsilon_{\text{tol}} = 0.1$, the upper and the lower bounds for ε -pseudospectra almost completely overlap. The convergence of the maximum relative error for $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell), \lambda_{\text{SLB}}(x, y; \mathcal{S}, \ell), \sigma_{\text{kry}}(x, y; k)^2, \sigma_{\text{inv}}(x, y; k)^2$ w.r.t. the subspace size is shown in Figure 2c, where we can see, what often happens in practice, that $\sigma_{\text{kry}}(x, y; k), \sigma_{\text{inv}}(x, y; k)$ often converge very slowly to the exact values of $\sigma_{\min}(zI - A)$.

Example 4.2. We consider the example `landau_demo.m` from *EigTool* [45], with matrix A representing an integral equation from laser theory [30]. We choose $N = 2000$ and $D = [-0.8, 1.2] + [-0.2, 0.2]i$, a region in the complex plane around the right-most part of the spectrum. There are five eigenvalues of A inside D which we initially include in \mathcal{S} . The spectrum of A (blue dots) in the region around D (red square) is shown in Figure 3a, whereas in Figure 3b we can see the convergence of the maximum error

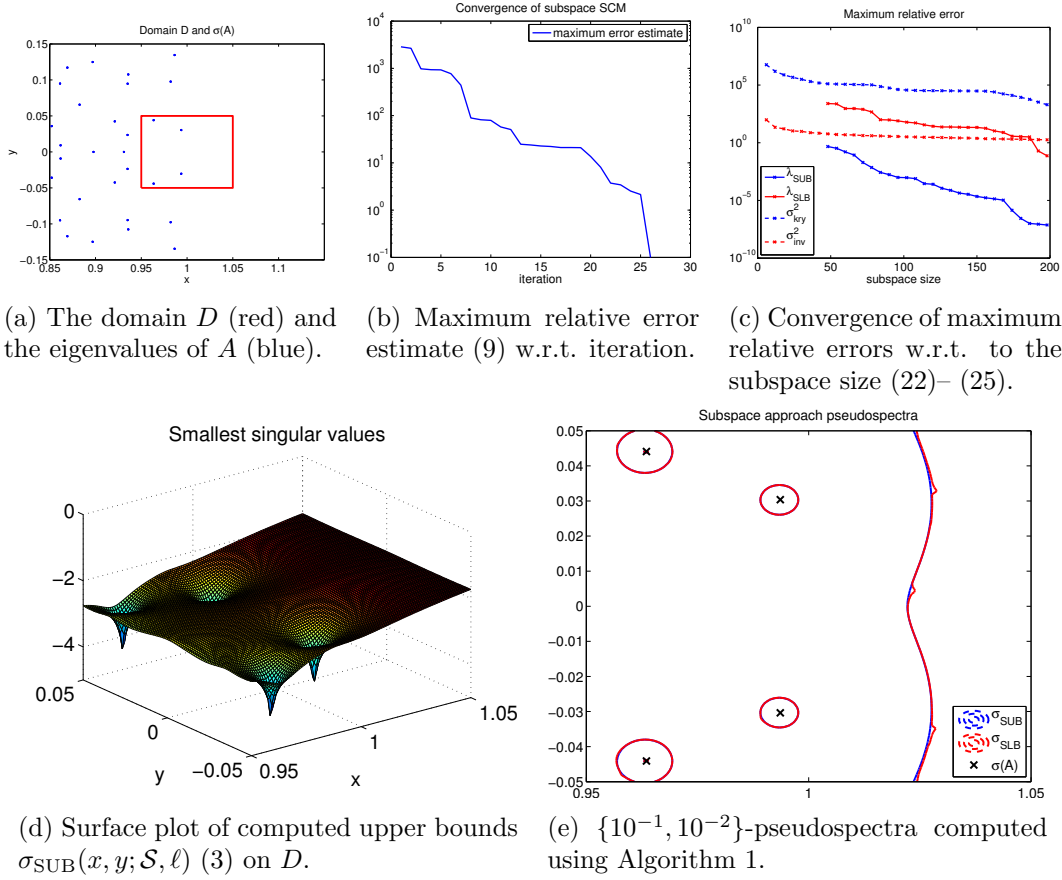


Figure 2: Application of Algorithm 1 to Example 4.1.

estimate in Algorithm 1 w.r.t. to iteration. The results are similar to those observed in Example 4.1, with Algorithm 1 reaching the desired tolerance in only 3 iterations with the computational time of 216 seconds, while the exact computation using a grid-based approach would take around 16200 seconds. Again, we see in Figure 3e that using $\varepsilon_{\text{tol}} = 0.1$ leads to very accurate approximations to $\{10^{-1}, 10^{-2}\}$ -pseudospectra, and observe in Figure 3c that the subspace estimates observe faster convergence w.r.t. the subspace size.

4.1.2 Sparse matrices

For a large-scale sparse matrix A , computing the Schur decomposition of A is rarely possible and almost never justified. We consider four large sparse matrices A and compute approximate pseudospectra $\sigma_\varepsilon(A)$, and compare the results obtained using Algorithm 1 with other approaches for pseudospectra computation. As explained in Section 3.1, we use the sparse LU decomposition of A to speed up the computation of $\lambda_{\min}(\hat{A}(x, y))$. For more details, see Examples 4.3 – 4.6.

Example 4.3. This example concerns an example from fluid dynamics, a model of a flow over obstacle, with $Re = 100$, linearized around steady state, using Q2-Q1 mixed finite

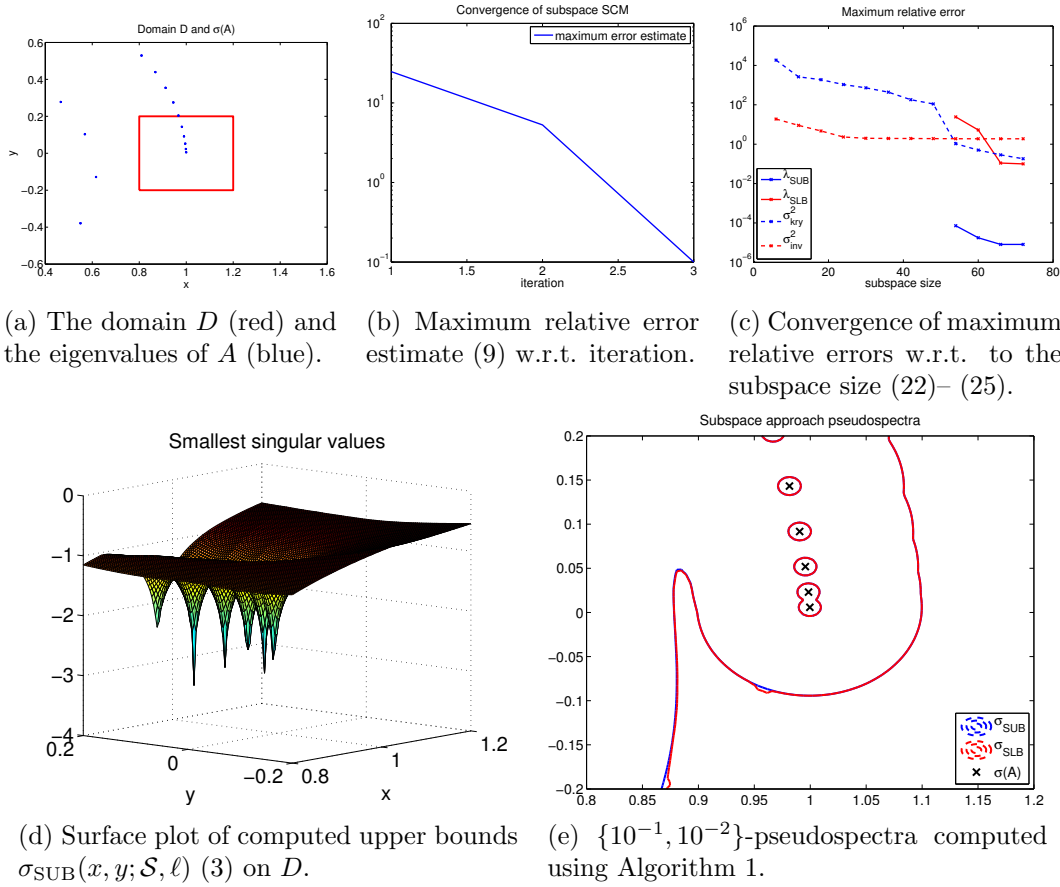


Figure 3: Application of Algorithm 1 to Example 4.2.

elements using *IFISS* [15]. We are given matrices A and M of size $N = 9512$ representing finite elements discretizations of the operator and the mass matrix, respectively. We compute pseudospectra of the matrix $M^{-1}A$ in $D = [-1.2, -0.2] + [-0.5, 0.5]i$, a region in the complex plane around the right-most part of the spectrum. There are three eigenvalues of A inside D which we initially include in \mathcal{S} . The spectrum of A (blue dots) in the region around D (red square) is shown in Figure 4a, whereas in Figure 4b we can see the convergence of the maximum error estimate in Algorithm 1 w.r.t. to iteration. Similarly as in previous examples, Algorithm 1 reaches the desired tolerance in 36 iterations with the computational time of 2006 seconds, while the exact computation using a grid-based approach would take around 11000 seconds, and yields accurate approximations to $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$ -pseudospectra as shown in Figure 4e.

Example 4.4. We consider the *MATPDE* example from the *Matrix Market* [1] collection of non-Hermitian eigenvalue problems, where the matrix A is a five-point central finite difference discretization of the two-dimensional variable-coefficient linear elliptic equation. Size of the matrix A is $N = 2961$ and we choose $D = [0, 0.1] + [-0.05, 0.05]i$, region in the complex plane around the left-most part of the spectrum. In this region there are six eigenvalues of A which we initially include in \mathcal{S} . The spectrum of A (blue dots) in the region around D (red square) is shown in Figure 5a, whereas in Figure 5b

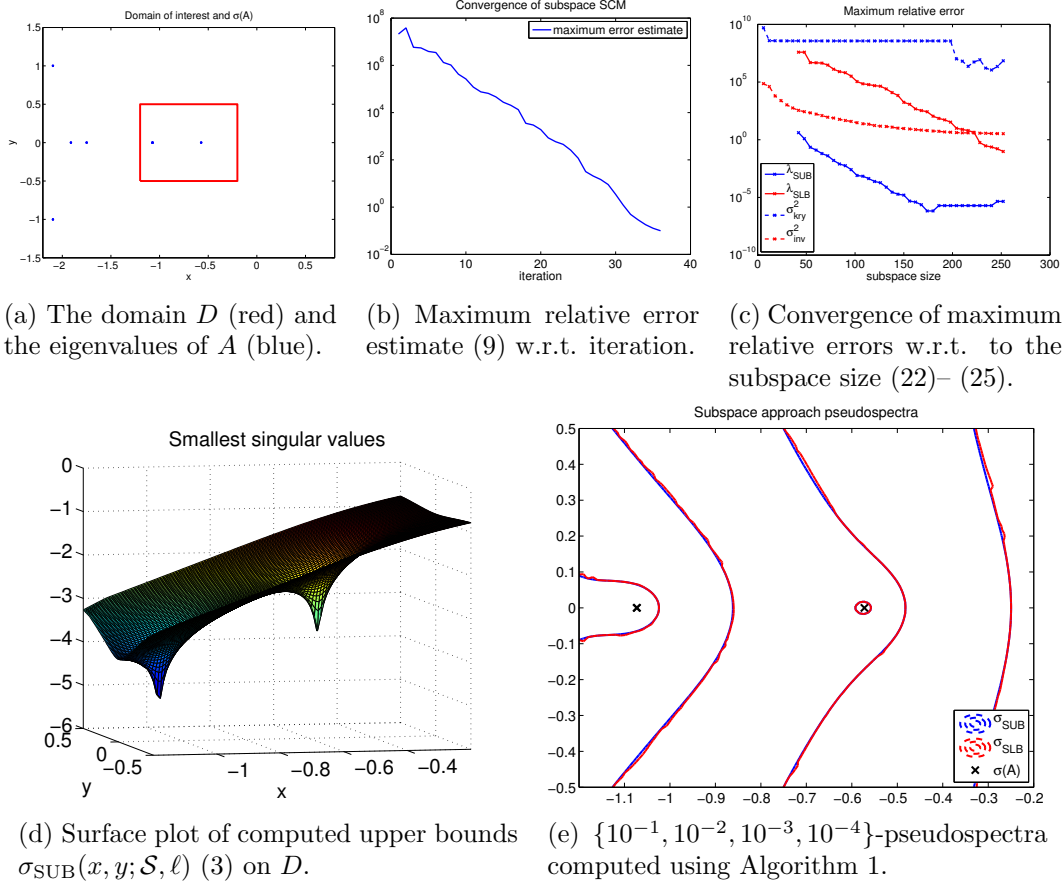


Figure 4: Application of Algorithm 1 to Example 4.3.

we can see the convergence of the maximum error estimate in Algorithm 1 w.r.t. to iteration. Again, we observe fast convergence of Algorithm 1, reaching the desired tolerance in 10 iterations with the computational time of 381 seconds, while the exact computation using a grid-based approach would take around 877 seconds, and providing accurate approximations to $\{10^{-2}, 10^{-3}, 10^{-4}\}$ -pseudospectra shown in Figure 5e.

Example 4.5. We consider the BRUSSEL example from the Matrix Market [1] collection of non-Hermitian eigenvalue problems, where the matrix A arises in chemical engineering as a discretization of a 2D reaction-diffusion model. Size of the matrix A is $N = 3200$ and we choose $D = [-0.5, 0.5] + [1.5, 2.5]i$, region in the complex plane around the right-most part of the spectrum. In this region there are three eigenvalues of A which we initially include in \mathcal{S} . The spectrum of A (blue dots) in the region around D (red square) is shown in Figure 6a, whereas in Figure 6b we can see the convergence of the maximum error estimate in Algorithm 1 w.r.t. to iteration. Algorithm 1 reaches the desired tolerance in 26 iterations with the computational time of 639 seconds, whereas the exact computation using a grid-based approach would take around 1550 seconds, and provides accurate approximations to $\{10^{-1}, 10^{-2}\}$ -pseudospectra.

Example 4.6. We consider the H2plus example from the Matrix Market [1] collection of non-Hermitian eigenvalue problems, where the matrix A arises in quantum chemistry

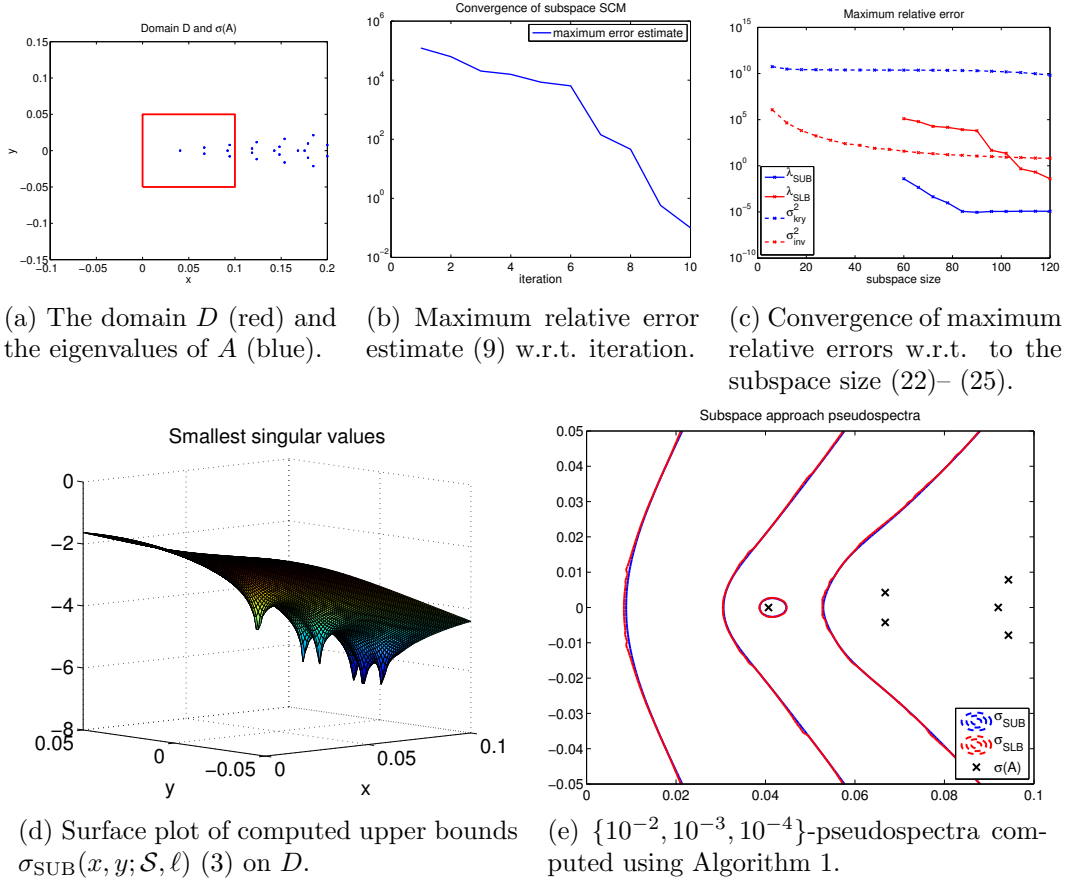


Figure 5: Application of Algorithm 1 to Example 4.4.

as a discretization of a model for H_2^+ in an electromagnetic field. Size of the matrix A is $N = 2534$ and we choose $D = [2.5, 3.5] + [-0.5, 0.5]i$, region in the complex plane around the right-most part of the spectrum. In this region there are six eigenvalues of A which we initially include in \mathcal{S} . The spectrum of A (blue dots) in the region around D (red square) is shown in Figure 7a, whereas in Figure 7b we can see the convergence of the maximum error estimate in Algorithm 1 w.r.t. to iteration. The "ward starting" proves to be very beneficial in this example, as Algorithm 1 requires only 2 iterations to reach the desired tolerance i the computational time of 130 seconds, while the exact computation using a grid-based approach would take around 8200 seconds, and still produces accurate approximations to $\{3 \cdot 10^{-1}, 10^{-1}, 3 \cdot 10^{-2}, 10^{-2}\}$ -pseudospectra. As can be seen in Figure 7c, the results are similar to those presented in the previous examples, with $\lambda_{\text{SUB}}(x, y; \mathcal{S}, \ell)$ providing a significantly better approximation than the other two approaches. The relative error for $\sigma_{\text{inv}}(x, y; k)^2$ increases for larger values of k due to the fact that not all eigenvectors included in the invariant subspace have converged.

4.2 Single domain vs multidomain

As discussed in Section 3.3, if $M\ell$, the number of columns in V , becomes too large for a certain example, then the computational cost of recomputing $\Delta(x, y; \mathcal{S}, \ell)$ becomes

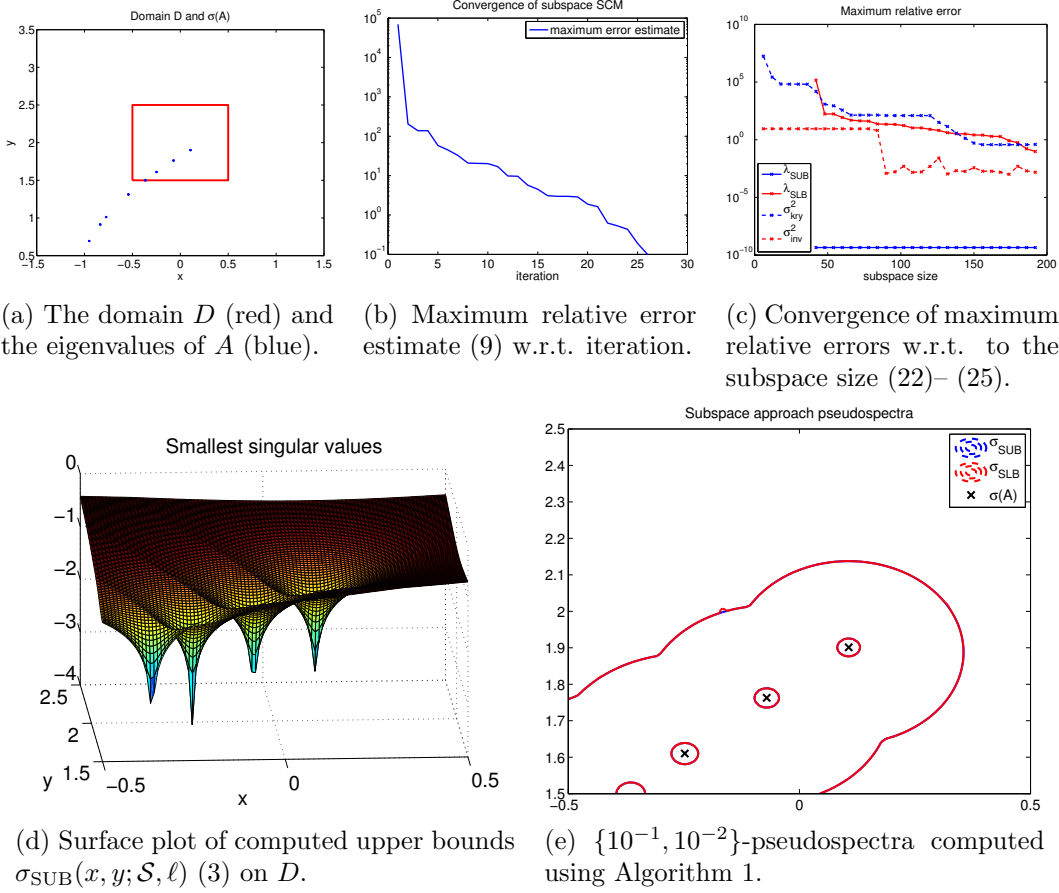


Figure 6: Application of Algorithm 1 to Example 4.5.

dominant. This is due to $\mathcal{O}((M\ell)^3)$ complexity for solving the "small" projected eigenvalue problem (5), resulting in the total time which may be even larger than that of a grid-based approach.

In the following, we present two numerical examples for which such an explosion in the computational time happens and show that it can be avoided using an adaptive approach with the domain splitting procedure presented in Algorithm 2. Similar speedup has been observed also for other examples where M becomes very large.

Example 4.7. We consider the same matrix as in Example 4.1 and compute ε -pseudospectra on the domain $D = [0.9, 1.1] + [-0.1, 0.1]i$. The observed matrix A has thirteen eigenvalues inside D . We compare the performance of Algorithm 1 without domain splitting ($\text{maxsize} = +\infty$) and with domain splitting ($\text{maxsize} = 150$). In Figures 8a–8c, we present the convergence of the maximum error estimate w.r.t. iteration, computational time and the number of solutions of large-scale singular value problems, respectively. We can see that after approximately 30 iterations using the domain splitting procedure accelerates the convergence despite the larger number of solved large-scale singular value problems. This can be explained using Figure 8d, which shows that in the single domain approach the computation of the error estimates becomes predominant after around 50 iterations.

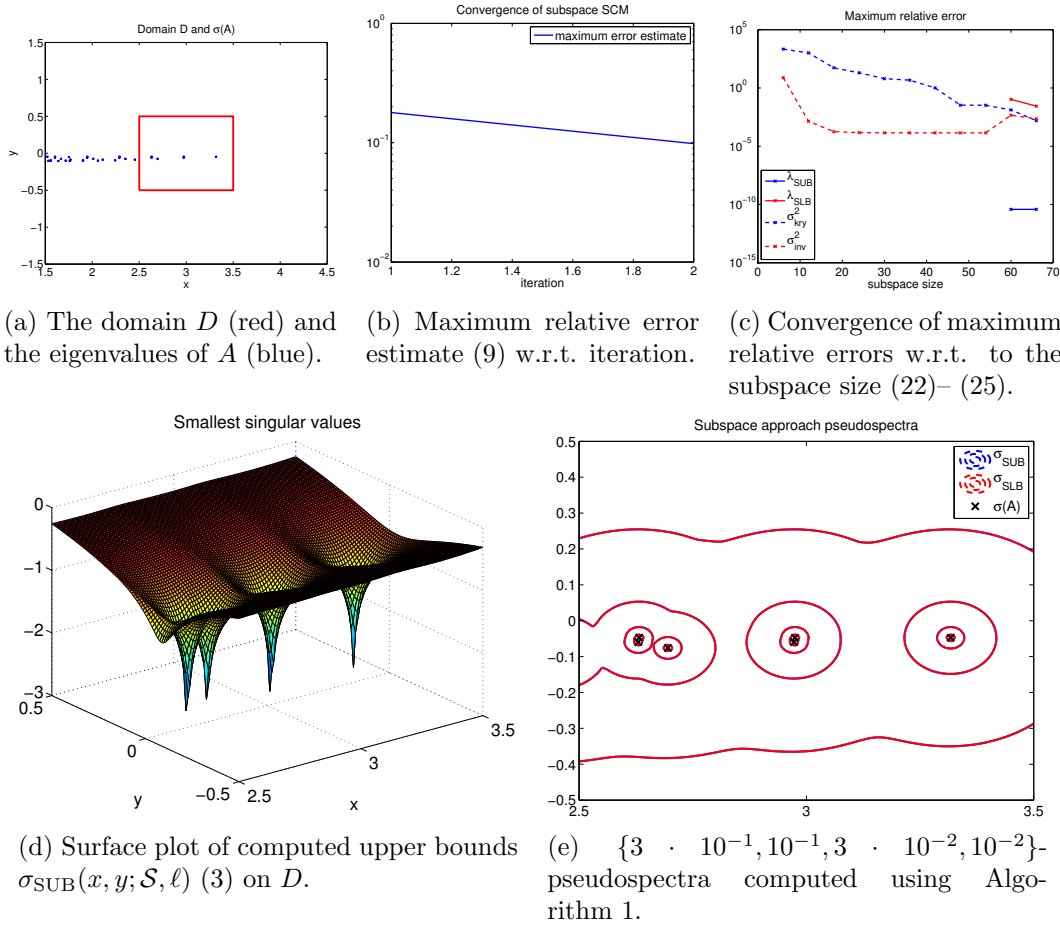


Figure 7: Application of Algorithm 1 to Example 4.6.

Example 4.8. We consider the same matrix as in Example 4.5 and compute ε -pseudospectra on the domain $D = [-1, 0.25] + [1, 2.25]i$. The matrix A has 10 eigenvalues inside D . The performance of Algorithm 1 without domain splitting ($\text{maxsize} = +\infty$) and with domain splitting ($\text{maxsize} = 150$) is shown in Figures 8a–8d. We can see that the results are similar to those in Example 4.7, with the adaptive approach again showing better results than the single domain approach after around 30 iterations.

5 Conclusions

For a large-scale matrix A , computing ε -pseudospectra using a grid based approach is usually computationally too expensive. Few projection-based approaches have been proposed as a potential solution. However, they usually suffer from slow convergence and lack of means to quantify obtained accuracy. We have proposed a new subspace approach, given in Algorithm 1, inspired by the greedy sampling strategies commonly used in reduced basis methods [37]. Our approach is primarily designed to provide highly accurate approximations of ε -pseudospectra in isolated parts of the spectrum, containing only few eigenvalues of A .

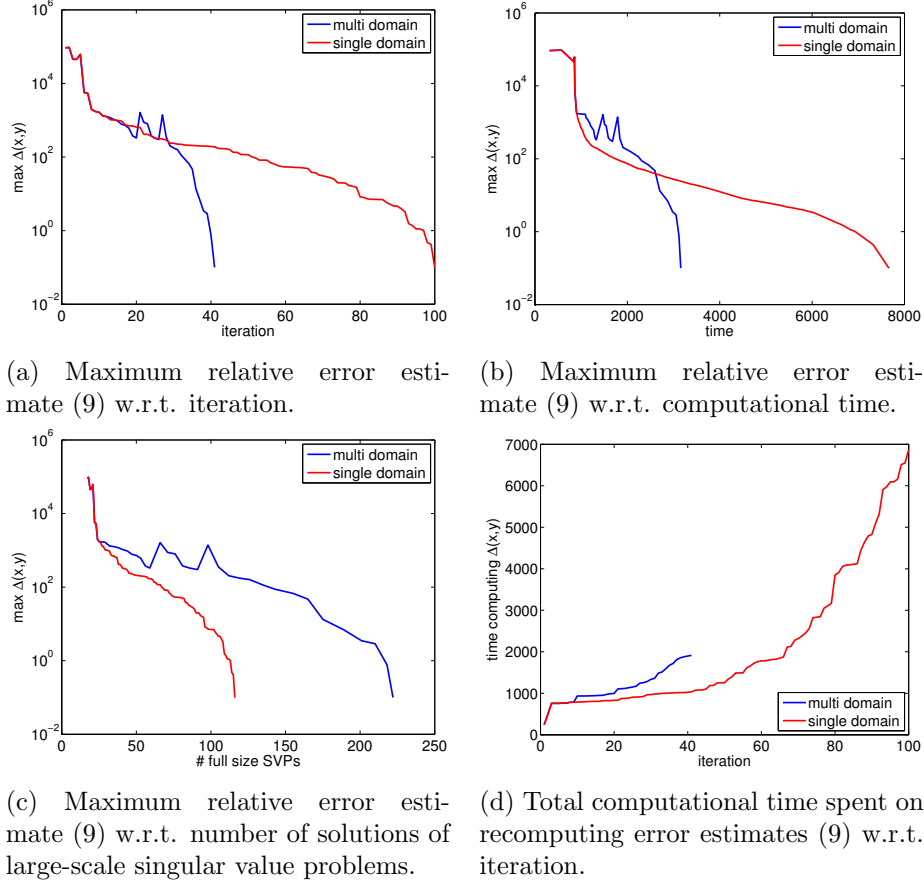


Figure 8: Performance of Algorithm 1 for Example 4.1 with and without domain splitting (Algorithm 2).

We have shown that the singular value problem $\sigma_{\min}(zI - A)$ can be recasted into a Hermitian eigenvalue problem linearly depending on two real parameters, and presented our new approach which builds upon an existing approach for approximating the smallest eigenvalues of a parameter-dependent Hermitian matrix [40]. Taking into account the particular problem structure and demands for high absolute accuracy led to the following modifications:

- simplified procedure for the computation of the lower bounds,
- accelerated solving of linear program (6) by using the simplex method with updating,
- more robust computation of the residual norm,
- optimization of the sampling strategy using the ideas from [23],
- "warm starting" of the approach using the eigenvalues of A inside the D .

It is these improvements that make our approach computationally competitive. Additionally, we have extended the interpolation results from [40] to the proposed singular value bounds, allowing us to provide *a priori* error estimates. Finally, we have also

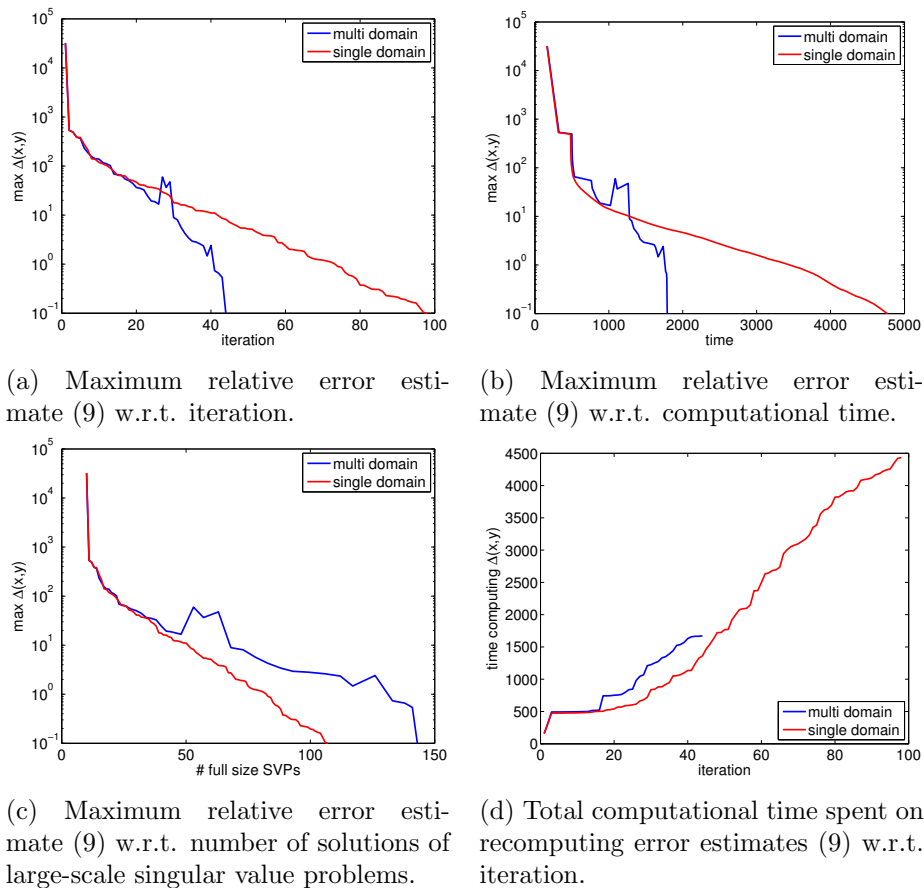


Figure 9: Performance of Algorithm 1 for Example 4.5 with and without domain splitting (Algorithm 2).

developed an adaptive version of Algorithm 1 by using a domain splitting procedure, presented in Algorithm 2, which makes the approach more robust to some more challenging examples.

We compared the performance of our approach to few other existing approaches on a number of examples discussed in the literature. Our approach presents to be significantly faster than the grid based approach, while providing satisfying accuracy together with rigorous error estimates. In comparison to the other projection-based approaches, our approach provides higher relative accuracy w.r.t. to the subspace size, especially in the proposed upper bounds $\sigma_{\text{SUB}}(x, y; \mathcal{S}, \ell)$, as well as the error estimates. Also, it is the first approach, to our knowledge, that provides certified upper bounds for ε -pseudospectra, enabling localization of eigenvalues. Finally, we have presented two numerical examples for which a larger sample set is required to attain satisfactory accuracy and for which benefits of using an adaptive version of our approach are evident.

Acknowledgements

We thank Daniel Kressner for constructive discussions and very helpful feedback, both on development of the approach as well as writing the paper. We thank Mark Embree

for providing us insight into pseudospectra computation as well as various interesting examples. We also thank Jonas Ballani, Michael Steinlechner, and Ana Šušnjara for discussing various ideas and approaches with us.

References

- [1] Matrix Market, 2007. Available at <http://math.nist.gov/MatrixMarket/>.
- [2] R. Andreev and C. Schwab. Sparse tensor approximation of parametric eigenvalue problems. In *Numerical analysis of multiscale problems*, volume 83 of *Lect. Notes Comput. Sci. Eng.*, pages 203–241. Springer, Heidelberg, 2012.
- [3] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28)*., 2015.
- [4] C. A. Beattie, M. Embree, and D. C. Sorensen. Convergence of polynomial restart Krylov methods for eigenvalue computations. *SIAM Rev.*, 47(3):492–515, 2005.
- [5] C. Bekas and E. Gallopoulos. Cobra: parallel path following for computing the matrix pseudospectrum. *Parallel Comput.*, 27(14):1879–1896, 2001.
- [6] A. Böttcher and S. M. Grudsky. *Spectral properties of banded Toeplitz matrices*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005.
- [7] T. Braconnier and N. J. Higham. Computing the field of values and pseudospectra using the Lanczos method with continuation. *BIT*, 36(3):422–440, 1996. International Linear Algebra Year (Toulouse, 1995).
- [8] M. Brühl. A curve tracing algorithm for computing the pseudospectrum. *BIT*, 36(3):441–454, 1996. International Linear Algebra Year (Toulouse, 1995).
- [9] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. K. Nichols. Numerical computation of an analytic singular value decomposition of a matrix valued function. *Numer. Math.*, 60(1):1–39, 1991.
- [10] J. V. Burke, A. S. Lewis, and M. L. Overton. Robust stability and a criss-cross algorithm for pseudospectra. *IMA J. Numer. Anal.*, 23(3):359–375, 2003.
- [11] E. B. Davies. *Linear operators and their spectra*, volume 106 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2007.
- [12] E. B. Davies and M. Plum. Spectral pollution. *IMA J. Numer. Anal.*, 24(3):417–438, 2004.
- [13] T. A. Davis. *Direct methods for sparse linear systems*, volume 2 of *Fundamentals of Algorithms*. SIAM, Philadelphia, PA, 2006.
- [14] J. L. Eftang, A. T. Patera, and E. M. Rønquist. An “hp” certified reduced basis method for parametrized elliptic partial differential equations. *SIAM J. Sci. Comput.*, 32(6):3170–3200, 2010.

- [15] H. C. Elman, A. Ramage, and D. J. Silvester. Algorithm 886: IFISS, a Matlab toolbox for modelling incompressible flow. *ACM Trans. Math. Software*, 33(2):Art. 14, 18, 2007.
- [16] S. K. Godunov and M. Sadkane. Computation of pseudospectra via spectral projectors. *Linear Algebra Appl.*, 279(1-3):163–175, 1998.
- [17] F. Gong, O. Meyerson, J. Meza, M. Stoiciu, and A. Ward. Explicit bounds for the pseudospectra of various classes of matrices and operators. *arXiv preprint arXiv:1505.05931*, 2015.
- [18] W. Govaerts. Computation of singularities in large nonlinear systems. *SIAM J. Numer. Anal.*, 34(3):867–880, 1997.
- [19] N. Guglielmi and M. L. Overton. Fast algorithms for the approximation of the pseudospectral abscissa and pseudospectral radius of a matrix. *SIAM J. Matrix Anal. Appl.*, 32(4):1166–1192, 2011.
- [20] B. Haasdonk, M. Dohlmann, and M. Ohlberger. A training set and multiple bases generation approach for parameterized model reduction based on adaptive grids in parameter space. *Math. Comput. Model. Dyn. Syst.*, 17(4):423–442, 2011.
- [21] A. Hannukainen. Convergence analysis of GMRES for the Helmholtz equation via pseudospectrum. *arXiv preprint arXiv:1505.08072*, 2015.
- [22] A. C. Hansen. On the approximation of spectra of linear operators on Hilbert spaces. *J. Funct. Anal.*, 254(8):2092–2126, 2008.
- [23] J. S. Hesthaven, B. Stamm, and S. Zhang. Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods. *ESAIM Math. Model. Numer. Anal.*, 48(1):259–283, 2014.
- [24] N. J. Higham. *Functions of matrices*. SIAM, Philadelphia, PA, 2008.
- [25] M. E. Hochstenbach. A Jacobi-Davidson type SVD method. *SIAM J. Sci. Comput.*, 23(2):606–628, 2001. Copper Mountain Conference (2000).
- [26] D. B. P. Huynh, G. Rozza, S. Sen, and A. T. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf-sup stability constants. *C. R. Math. Acad. Sci. Paris*, 345(8):473–478, 2007.
- [27] F. Kangal, K. Meerbergen, E. Mengi, and W. Michiels. A subspace method for large scale eigenvalue optimization. *arXiv preprint arXiv:1508.04214*, 2015.
- [28] Herbert B. Keller. Constructive methods for bifurcation and nonlinear eigenvalue problems. In *Computing methods in applied sciences and engineering (Proc. Third Internat. Sympos., Versailles, 1977)*, I, volume 704 of *Lecture Notes in Math.*, pages 241–251. Springer, Berlin, 1979.
- [29] D. Kressner and B. Vandereycken. Subspace methods for computing the pseudospectral abscissa and the stability radius. *SIAM J. Matrix Anal. Appl.*, 35(1):292–313, 2014.

- [30] H. J. Landau. The notion of approximate eigenvalues applied to an integral equation of laser theory. *Quart. Appl. Math.*, 35(1):165–172, 1977/78.
- [31] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK users’ guide*. SIAM, Philadelphia, PA, 1998. Solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods.
- [32] C.-K. Li and R.-C. Li. A note on eigenvalues of perturbed Hermitian matrices. *Linear Algebra Appl.*, 395:183–190, 2005.
- [33] S. H. Lui. Computation of pseudospectra by continuation. *SIAM J. Sci. Comput.*, 18(2):565–573, 1997.
- [34] K. Meerbergen, E. Mengi, W. Michiels, and R. Van Beeumen. Computation of pseudospectral abscissa for large scale nonlinear eigenvalue problems. Technical report, 2015.
- [35] D. Mezher and B. Philippe. PAT—a reliable path-following algorithm. *Numer. Algorithms*, 29(1-3):131–152, 2002. Matrix iterative analysis and biorthogonality (Luminy, 2000).
- [36] S. C. Reddy, P. J. Schmid, and D. S. Henningson. Pseudospectra of the Orr-Sommerfeld operator. *SIAM J. Appl. Math.*, 53(1):15–47, 1993.
- [37] G. Rozza, D.B.P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics. *Arch. Comput. Methods Eng.*, 15(3):229–275, 2008.
- [38] P. J. Schmid. Nonmodal stability theory. In *Annual review of fluid mechanics. Vol. 39*, volume 39 of *Annu. Rev. Fluid Mech.*, pages 129–162. Annual Reviews, Palo Alto, CA, 2007.
- [39] V. Simoncini and E. Gallopoulos. Transfer functions and resolvent norm approximation of large matrices. *Electron. Trans. Numer. Anal.*, 7:190–201, 1998. Large scale eigenvalue problems (Argonne, IL, 1997).
- [40] P. Sirković and D. Kressner. Subspace acceleration to parameter-dependent hermitian eigenproblems. *To appear in SIAM J Matrix Anal. Appl.*, 2016.
- [41] K.-C. Toh and L. N. Trefethen. Calculation of pseudospectra by the Arnoldi iteration. *SIAM J. Sci. Comput.*, 17(1):1–15, 1996. Special issue on iterative methods in numerical linear algebra (Breckenridge, CO, 1994).
- [42] L. N. Trefethen. Computation of pseudospectra. In *Acta numerica, 1999*, volume 8 of *Acta Numer.*, pages 247–295. Cambridge Univ. Press, Cambridge, 1999.
- [43] L. N. Trefethen and M. Embree. *Spectra and pseudospectra*. Princeton University Press, Princeton, NJ, 2005.
- [44] C. Tretter. *Spectral theory of block operator matrices and applications*. Imperial College Press, London, 2008.

- [45] T. G. Wright. EigTool, 2002. Available at <http://www.comlab.ox.ac.uk/pseudospectra/eigtool/>.
- [46] T. G. Wright and L. N. Trefethen. Large-scale computation of pseudospectra using ARPACK and eigs. *SIAM J. Sci. Comput.*, 23(2):591–605, 2001. Copper Mountain Conference (2000).

Recent publications:

MATHEMATICS INSTITUTE OF COMPUTATIONAL SCIENCE AND ENGINEERING
Section of Mathematics
Ecole Polytechnique Fédérale (EPFL)
CH-1015 Lausanne

- 22.2016** JONAS BALLANI, DANIEL KRESSNER, MICHAEL PETERS:
Multilevel tensor approximation of PDEs with random data
- 23.2016** DANIEL KRESSNER, ROBERT LUCE:
Fast computation of the matrix exponential for a Toeplitz matrix
- 24.2016** FRANCESCA BONIZZONI, FABIO NOBILE, ILARIA PERUGIA:
Convergence analysis of Padé approximations for Helmholtz frequency response problems
- 25.2016** MICHELE PISARONI, FABIO NOBILE, PÉNÉLOPE LEYLAND:
A continuation multi level Monte Carlo (C-MLMC) method for uncertainty quantification in compressible aerodynamics
- 26.2016** ASSYR ABDULLE, ONDREJ BUDÁČ:
Multiscale model reduction methods for flow in heterogeneous porous media
- 27.2016** ASSYR ABDULLE, PATRICK HENNING:
Multiscale methods for wave problems in heterogeneous media
- 28.2016** ASSYR ABDULLE, ORANE JECKER:
On heterogeneous coupling of multiscale methods for problems with and without scale separation
- 29.2016** WOLFGANG HACKBUSCH, DANIEL KRESSNER, ANDRÉ USCHMAJEW:
Perturbation of higher-order singular values
- 30.2016** ASSYR ABDULLE, ONDREJ BUDÁČ, ANTOINE IMBODEN:
Multiscale methods and model order reduction for flow problems in three-scale porous media
- 31.2016** DANIEL KRESSNER, ANA ŠUŠNJARA:
Fast computation of spectral projectors of banded matrices
- 32.2016** NICCOLO DAL SANTO, SIMONE DEPARIS, ANDREA MANZONI, ALFIO QUARTERONI:
Multi space reduced basis preconditioners for large-scale parametrized PDEs
- 33.2016** ASSYR ABDULLE, MARTIN HUBER, SIMON LEMAIRE:
An optimization-based numerical method for diffusion problems with sign-changing coefficients
- 34.2016** ASSYR ABDULLE, ANDREA DI BLASIO:
Numerical homogenization and model order reduction for multiscale inverse problems
- 35.2016** PETAR SIRKOVIĆ:
A reduced basis approach to large-scale pseudospectra computation