# On the computation of structured singular values and pseudospectra

Michael Karow

*Institut für Mathematik, TU Berlin*

Effrosyni Kokiopoulou, Daniel Kressner

*Seminar for Applied Mathematics, ETH Zurich*

**Abstract**

Structured singular values and pseudospectra play an important role in assessing the properties of a linear system under structured perturbations. This paper discusses computational aspects of structured pseudospectra for structures that admit an eigenvalue minimization characterization, including the classes of real, skew-symmetric, Hermitian, and Hamiltonian perturbations. For all these structures we develop algorithms that require $O(n^2)$ operations per grid point, combining the Schur decomposition with a Lanczos method. These algorithms form the basis of a graphical MATLAB interface for plotting structured pseudospectra.

*Key words:* Structured pseudospectrum, structured singular value, real perturbations, skew-symmetric perturbations, Hermitian perturbations, Hamiltonian perturbations.

## 1. Introduction

Structured singular values and pseudospectra are useful tools for analyzing the spectral behavior of matrices and dynamical systems under uncertainties.

Given a class of perturbations $\mathbf{\Delta} \subseteq \mathbb{C}^{n\times n}$, the *structured singular value* of a matrix $B \in \mathbb{C}^{n\times n}$ is defined as

$$\mu_{\mathbf{\Delta}}(B) = [\inf\{\|\triangle\| : \triangle \in \mathbf{\Delta} \text{ and } \det(I - \triangle B) = 0\}]^{-1}. \quad (1)$$

Here and in the following, $\|\cdot\|$ denotes the 2-norm (also called spectral norm) of a matrix. If $B$ is invertible, $\mu_{\mathbf{\Delta}}(B)^{-1}$ is the minimal norm among all perturbations that make $B^{-1}$ singular. In the unstructured case, $\mathbf{\Delta} = \mathbb{C}^{n\times n}$ and $\mu_{\mathbf{\Delta}}(B) = \|B\|$ coincides with the largest singular value of $B$.

The structured pseudospectrum of a matrix $A \in \mathbb{C}^{n\times n}$ with respect to $\mathbf{\Delta}$ and a threshold $\epsilon > 0$ is defined as

$$\Lambda_{\mathbf{\Delta}}(A;\epsilon) := \bigcup_{\triangle\in\mathbf{\Delta},\|\triangle\|<\epsilon} \Lambda(A+\triangle), \quad (2)$$

where $\Lambda(\cdot)$ denotes the spectrum of a matrix. Structured pseudospectra are closely tied to structured singular values, as seen from the relation

$$\Lambda_{\mathbf{\Delta}}(A;\epsilon) = \{\lambda \in \mathbb{C} : \mu_{\mathbf{\Delta}}((A-\lambda I)^{-1}) > 1/\epsilon\}. \quad (3)$$

Note that in the unstructured case $\mathbf{\Delta} = \mathbb{C}^{n\times n}$, we have $\mu_{\mathbf{\Delta}}((A-\lambda I)^{-1}) = \|(A-\lambda I)^{-1}\| = 1/\sigma_{\min}(A-\lambda I)$, where $\sigma_{\min}(\cdot)$ denotes the smallest singular value of a matrix.

*Email addresses:* karow@math.tu-berlin.de (Michael Karow), effrosyni.kokiopoulou@sam.math.ethz.ch (Effrosyni Kokiopoulou), daniel.kressner@sam.math.ethz.ch (Daniel Kressner)

The focus of this paper is on how to actually compute (3) or, equivalently, how to evaluate $\mu_{\mathbf{\Delta}}((A-\lambda I)^{-1})$ for a fixed matrix $A$ and many different values of $\lambda \in \mathbb{C}$. For $\mathbf{\Delta} = \mathbb{C}^{n\times n}$, the software package EigTool (Wright 2002) is routinely used for plotting unstructured pseudospectra. The computational cost of EigTool benefits from two tricks, described for example in the book by Trefethen and Embree (2005, Chapter 39). First, prior to any pseudospectral computation, $A$ is replaced by its Schur form, reducing the cost of multiplying $(A-\lambda I)^{-1}$ with a vector from $O(n^3)$ to $O(n^2)$. It is crucial to note that this trick works because the unstructured pseudospectrum is invariant under unitary similarity transformations. (This property does *not* hold in general for structured pseudospectra.) Second, a Lanczos method (Bai et al. 2000; Golub and Van Loan 1996) is employed to estimate $\|(A-\lambda I)^{-1}\|$ based on matrix-vector products with $(A-\lambda I)^{-1}$ and its conjugate transpose.

The difficulty of computing structured singular values and pseudospectra heavily depends on the nature of $\mathbf{\Delta}$ and ranges from trivial to NP-hard. In the following we provide a brief survey of available results in the literature.

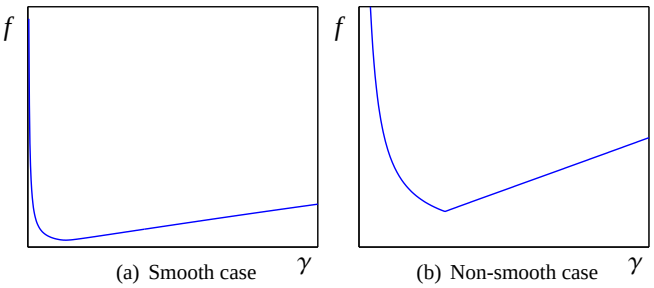
*Computationally trivial classes* $\mathbf{\Delta}$. For a surprisingly wide range of linear structures $\mathbf{\Delta}$, it can be shown that $\mu_{\mathbf{\Delta}}((A-\lambda I)^{-1}) \equiv \|(A-\lambda I)^{-1}\|$ for $A \in \mathbf{\Delta}$ and any $\lambda \in \mathbb{C}$. In these cases there is consequently no difference between structured and unstructured pseudospectra. This has been shown by Rump (2006) for the classes of symmetric, persymmetric, Toeplitz, symmetric Toeplitz, Hankel, persymmetric Hankel, and circulant matrices. The basic idea of the proof is to construct a *structured mapping* $\triangle \in \mathbf{\Delta}$ such that $\triangle u = v$ for $v = \alpha(A-\lambda I)u$, $|\alpha| = 1$, and some vector $u \in \mathbb{C}^n$ with

$\|u\| = 1$ and $\|(A - \lambda I)u\| = \sigma_{\min}(A - \lambda I)$. Then the relation $\mu_{\mathbf{\Delta}}((A - \lambda I)^{-1}) = \|(A - \lambda I)^{-1}\|$ readily follows from the definition (1). A variety of such structured mappings can be found in (Mackey et al. 2007; Rump 2003). Their connections to structured pseudospectra/eigenvalue condition numbers have also been investigated in (Graillat 2006; Karow 2007; Karow et al. 2006a).

*Computationally tractable classes* $\mathbf{\Delta}$. Many structures admit structured mappings $\triangle u = v$ with $\triangle \in \mathbf{\Delta}$ only under certain constraints on the vectors $u, v$. For example, a skew-symmetric mapping $\triangle$ is possible if and only if $u, v$ are orthogonal to each other. In such cases, the evaluation of $\mu_{\mathbf{\Delta}}((A-\lambda I)^{-1})$ amounts to the solution of a constrained optimization problem. For certain structures, the latter can be rewritten as an unconstrained eigenvalue/singular value optimization problem that is computationally tractable in the sense that the global optimum can be reliably computed/approximated in polynomial time. This holds not only for the well-known case $\mathbf{\Delta} = \mathbb{R}^{n\times n}$ as shown by Qiu et al. (1995), but also for complex skew-symmetric, Hermitian and other linear structures induced by bilinear and sesquilinear forms (Karow 2007). Solely concerned with such structures, this paper focuses on the efficient solution of the eigenvalue optimization problems associated with $\mu_{\mathbf{\Delta}}((A - \lambda I)^{-1})$, particularly aiming at the computation of structured pseudospectra. Instances of computationally tractable classes $\mathbf{\Delta}$ not considered in this paper include complex diagonal block structures with less than 4 blocks (Doyle 1982), complex off-diagonal block structures (Karow et al. 2006b), and structures related to higher order systems (Pappas and Hinrichsen 1993; Soh et al. 1985; Tisseur and Higham 2001).

*Computationally intractable classes* $\mathbf{\Delta}$. For many structures computational methods for evaluating $\mu_{\mathbf{\Delta}}$ are either prohibitively expensive or not even known. In fact, for a number of structures the evaluation of $\mu_{\mathbf{\Delta}}$ has been proven to be NP-hard, most prominently for diagonal (Demmel 1992) and real block diagonal structures (Poljak and Rohn 1993). Despite these pessimistic results, reliable and somewhat efficient techniques for obtaining usually tight lower and upper bounds on $\mu_{\mathbf{\Delta}}$ for mixed complex/real diagonal structures are available and form the basis of MATLAB's $\mu$-Analysis and Synthesis Toolbox (Balas et al. 1993). However, a summary of these techniques is beyond the scope of this paper; we refer to (Packard and Doyle 1993) for a survey of early results in this direction.

*Outline of the paper.* The rest of this paper is organized as follows. Sections 2 and 3 are concerned with the computation of real structured singular values and real structured pseudospectra, respectively. In particular, an algorithm is developed that computes the real pseudospectrum at a cost of $O(n^2)$ operations per grid point. These results are extended to skew-symmetric, Hermitian, and Hamiltonian pseudospectra in Sections 4, 5, and 6, respectively. The algorithms presented in this paper form the basis of a MATLAB graphical interface. This interface is briefly described in Section 7, which also contains a few experiments concerning computational efficiency.




(a) Smooth case (b) Non-smooth case

Figure 1: Pictorial plots illustrating the shape of $f_{\text{real}}(\gamma)$.

## 2. Computing real structured singular values

In the case of real perturbations $\mathbf{\Delta} = \mathbb{R}^{n\times n}$, a well-known result (Qiu et al. 1995) shows that the corresponding structured singular value $\mu_{\text{real}}$ defined in (1) satisfies

$$\mu_{\text{real}}(B) = \inf_{\gamma\in(0,1]} \sigma_2\left(\begin{bmatrix} \mathcal{R}B & -\gamma^{-1}\mathcal{I}B \\ \gamma\mathcal{I}B & \mathcal{R}B \end{bmatrix}\right),$$

where $\mathcal{R}B$ and $\mathcal{I}B$ denote the real and imaginary parts of $B \in \mathbb{C}^{n\times n}$, respectively, and $\sigma_2(\cdot)$ denotes the second largest singular value of a matrix. Hence, computing $\mu_{\text{real}}(B)$ involves a singular value optimization with respect to $\gamma \in (0, 1]$. To discuss this optimization problem in more detail, let us define

$$C(\gamma) = \begin{bmatrix} \mathcal{R}B & -\gamma^{-1}\mathcal{I}B \\ \gamma\mathcal{I}B & \mathcal{R}B \end{bmatrix}, \tag{4}$$

with which

$$\mu_{\text{real}}(B) = \inf_{\gamma\in(0,1]} \sigma_2(C(\gamma)). \tag{5}$$

The objective function $f_{\text{real}}(\gamma) = \sigma_2(C(\gamma))$ in (5) is *quasi-convex* (Qiu et al. 1995). A function $f : \mathcal{X} \to \mathbb{R}$ is called quasi-convex if for each $x, y \in \mathcal{X}$ and $\xi \in [0, 1]$, it holds that

$$f(\xi x + (1 - \xi)y) \le \max\{f(x), f(y)\}. \tag{6}$$

Any quasi-convex function has convex level sets. Quasiconvexity also implies that a local minimizer will be a global minimizer. Figure 1(a) illustrates what the objective function $f_{\text{real}}(\gamma)$ typically looks like.

### 2.1. Solving the optimization problem

We now discuss the solution of the optimization problem (5). The optimal solution can be obtained by localizing the point $\gamma^*$ where the derivative $\dot{f}_{\text{real}}(\gamma)$ changes sign. In order to compute $\gamma^*$ we have chosen to use *bisection* as a simple and robust root bracketing method. At iteration $k$, bisection employs an interval $[a_k, b_k]$ that contains the optimum, and divides this interval into two sub-intervals of equal length. In particular, it picks the midpoint point $\gamma_{k+1} = \frac{a_k+b_k}{2}$ and then selects one of $[a_k, \gamma_{k+1}]$ or $[\gamma_{k+1}, b_k]$, depending on the sign of $\dot{f}_{\text{real}}(\gamma_{k+1})$. The method is stopped when the length of the interval $|b_k - a_k|$ becomes smaller than a given tolerance.

**Algorithm 1**: The Lanczos algorithm

1: **Input**: Matrix $\tilde{C}$, residual tolerance tol
starting vector $r$.
2: **Output**: $Q_k$: Lanczos basis, coefficients $\alpha_k$ and $\beta_k$
3: **Initialization**: $\beta_0 = \|r\|_2$.
4: **for** $k = 1, 2, \ldots$ **do**
5: $q_k = r/\beta_{k-1}$.
6: $r = \text{matvec}(\tilde{C}, q_k)$.
7: $r = r - q_{k-1}\beta_{k-1}$.
8: $\alpha_k = \langle q_k, r\rangle$.
9: {Full reorthogonalization}
10: **for** $l = 1, \ldots, k-1$ **do**
11: $r = r - \langle r, q_l\rangle q_l$
12: **end for**
13: $\beta_k = \|r\|_2$.
14: {Check for convergence}
15: Compute the eigenpair of interest $T_k w = \theta w$.
16: Compute the residual $\text{res}(k) = |\beta_k w(k)|$.
17: **if** $\text{res}(k) \le \text{tol}$ **then**
18: Set $Q_k = [q_1, \ldots, q_k]$ and break.
19: **end if**
20: **end for**

At each step of bisection, we need to evaluate the derivative of $f_{\text{real}}(\gamma) = \sigma_2(C(\gamma))$. If the second largest singular value is simple then $\sigma_2(\cdot)$ is differentiable and

$$\dot{\sigma}_2(C(\gamma)) = Re\left[u_2^* \dot{C}(\gamma) v_2\right], \tag{7}$$

where $u_2$ and $v_2$ are left and right singular vectors of $C(\gamma)$ corresponding to $\sigma_2(C(\gamma))$, see (Sun 1988). With $C(\gamma)$ defined as in (4),

$$\dot{C}(\gamma) = \begin{bmatrix} \mathbf{0} & 1/\gamma^2 \mathcal{I} B \\ \mathcal{I} B & \mathbf{0} \end{bmatrix}.$$

If the second largest singular value fails to be simple then (7) yields an element of the subdifferential, see Lewis and Sendov (2005), which can equally be used in the bisection method. We conclude that at each step $k$ of bisection we need to compute a singular triplet (singular value, right and left singular vectors) to determine $\dot{f}_{\text{real}}(\gamma_k)$. However, it is important to stress that this computation is not required to be very accurate except for the last few iterations, since only the sign of $\dot{\sigma}_2(C(\gamma))$ is of interest. In the next section, when we discuss the computation of real structured pseudospectra, it will be shown how to take advantage of this observation.

We have also experimented with a regula falsi method, combining bisection with the secant method. This somewhat decreases the number of iterations but requires higher accuracy for $\dot{\sigma}_2(C(\gamma))$. In effect, the improvement of the overall execution time was modest at best.

## 3. Computing real structured pseudospectra

By (3), the real structured pseudospectra of $A$ are the level sets of $\mu_{\text{real}}((A - \lambda I)^{-1})$, $\lambda \in \mathbb{C}$. For plotting the real structured pseudospectrum of $A$, we introduce a two-dimensional grid $\mathcal{G}$ on the complex plane and evaluate $\mu_{\text{real}}((A - \lambda I)^{-1})$ at each grid point $\lambda \in \mathcal{G}$. In light of our discussion in the previous section, observe that we need to apply bisection in order to solve an eigenvalue optimization problem for each grid point $\lambda \in \mathcal{G}$. Each step of bisection in turn involves the computation of a singular triplet of the matrix $C(\gamma)$ defined in (4) with $B = (A - \lambda I)^{-1}$. Since computing a singular value decomposition for each iteration becomes rather expensive for larger $n$, we propose below an efficient approach for computing real structured pseudospectra, based on the Lanczos method combined with the Schur decomposition of $A$.

### 3.1. The Lanczos method

Let us recall that we only need to determine the second largest singular value $\sigma_2$ and singular vectors $u_2, v_2$ of $C(\gamma)$, see (7). Equivalently, $\sigma_2^2$ is the second largest eigenvalue of the Hermitian matrix $\tilde{C} = C(\gamma)^* C(\gamma)$ and $v_2$ is a corresponding eigenvector. Moreover, $u_2 = \frac{1}{\sigma_2} C(\gamma) v_2$.

To approximate the eigenvalues of $\tilde{C}$ we apply the Lanczos method to $\tilde{C}$ (Bai et al. 2000; Golub and Van Loan 1996), see also Algorithm 1. Each step of the Lanczos method involves one matrix-vector multiplication with $C(\gamma)$ followed by a matrix-vector multiplication with $C(\gamma)^*$. The version shown in Algorithm 1 applies full reorthogonalization to retain numerical accuracy. After $k$ steps of Lanczos, we obtain a decomposition of the form

$$\tilde{C} Q_k = Q_k T_k + r e_k^*, \tag{8}$$

where $Q_k^* r = 0$ and $T_k$ is a tridiagonal symmetric real matrix

$$T_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & \beta_{k-1} & \alpha_k \end{bmatrix} \tag{9}$$

composed of the coefficients generated in the course of the method. Well-known convergence results (Bai et al. 2000) imply that the second largest eigenvalue of $T_k$ provides a good approximation to $\sigma_2^2$ for sufficiently large $k$. If $w$ denotes the corresponding eigenvector of $T_k$ then $Q_k w$ provides an approximation to $u_2$.

It is important to observe that the main computational kernel of the Lanczos method is the matrix-vector multiplication of $\tilde{C}$ with $q_k$ (Line 6). Hence, we only need the action of the matrix $\tilde{C}$ on a vector and can avoid the explicit formation of $\tilde{C}$.

### 3.2. Efficient computation of matrix-vector products

To multiply $\tilde{C}$ with a vector $x$ we first need to multiply with $C(\gamma)$ and then with $C^*(\gamma)$. Let us consider the multiplication with $C(\gamma)$:

$$\begin{aligned} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} \mathcal{R} B & -\gamma^{-1} \mathcal{I} B \\ \gamma \mathcal{I} B & \mathcal{R} B \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{R} B x_1 - \gamma^{-1} \mathcal{I} B x_2 \\ \gamma \mathcal{I} B x_1 + \mathcal{R} B x_2 \end{bmatrix}. \end{aligned} \tag{10}$$

Observe that

$$(y_1 + i\gamma^{-1}y_2) = (\mathcal{R}B + i\mathcal{I}B)(x_1 + i\gamma^{-1}x_2), \tag{11}$$

which implies that $y_1, y_2$ can be computed by performing one complex matrix-vector multiplication with $B = \mathcal{R}B + i\mathcal{I}B$ and scaling the imaginary parts. A similar statement holds for the matrix-vector multiplication with $C(\gamma)^*$:

$$\begin{aligned}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} &= \begin{bmatrix} \mathcal{R}B^\top & \gamma\mathcal{I}B^\top \\ -\gamma^{-1}\mathcal{I}B^\top & \mathcal{R}B^\top \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= \begin{bmatrix} \mathcal{R}B^\top x_1 + \gamma\mathcal{I}B^\top x_2 \\ -\gamma^{-1}\mathcal{I}B^\top x_1 + \mathcal{R}B^\top x_2 \end{bmatrix},\end{aligned} \tag{12}$$

which can be written as

$$(y_1 + i\gamma y_2) = (\mathcal{R}B^\top - i\mathcal{I}B^\top)(x_1 + i\gamma x_2), \tag{13}$$

reducing the computation to a matrix-vector multiplication with $B^*$.

In summary, one step of Lanczos applied to $C(\gamma)^*C(\gamma)$ requires two matrix-vector multiplications with $B$ and $B^*$, respectively. In order to make this computation more efficient, we propose to perform a Schur decomposition of $A$ a priori, before the pseudospectrum computation starts. Then, using the fact that $A = URU^*$, where $U$ is unitary and $R$ is upper triangular, it holds that

$$B = (A - \lambda I)^{-1} = (URU^* - \lambda I)^{-1} = U(R - \lambda I)^{-1}U^*. \tag{14}$$

A matrix-vector multiplication with $B$ (or $B^*$) now requires a matrix-vector multiplication with $U^*$, followed by a linear system solution with an upper (lower) triangular matrix $R$ ($R^*$) and, finally, a matrix-vector multiplication with $U$. This not only avoids the explicit formation of $B$ but also reduces the cost of matrix-vector multiplications with $B$ or $B^*$ to $O(n^2)$.

### 3.3. Stopping criterion and accuracy

As we have already mentioned in Section 2, each step of bisection only requires the sign of $\dot{\sigma}_2(C(\gamma))$. This suggests that one can significantly relax the residual tolerance used in Lanczos (see Algorithm 1, Line 17) and still be able to estimate the sign correctly. Numerical experience suggests that a tolerance of $10^{-5}$ suffices. This limits the number of matrix-vector multiplications required by the Lanczos method and therefore the total computational cost of bisection itself. Nevertheless, it is important to stress that once the optimal solution $\gamma^*$ has been localized, we require high accuracy in the final approximation of $\mu_{\text{real}}((A - \lambda I)^{-1}) = \sigma_2(C(\gamma^*))$, so we set the tolerance to $10^{-15}$ in the Lanczos method (only) at this final step. Algorithm 2 summarizes the main steps of bisection for computing $\mu_{\text{real}}((A-\lambda I)^{-1})$. The matrix-vector multiplication needed in the Lanczos method called in lines 6 and 15 (with the two different tolerances) is performed in $O(n^2)$ operations with the trick described in Section 3.2.

**Algorithm 2**: The bisection algorithm

1: **Input**: $URU^*$: Schur decomposition of $A$, $\lambda$: shift, $\mathsf{tol}_h$, $\mathsf{tol}_l$: high and low tolerance, $[a_0, b_0]$: initial interval.
2: **Output**: $\gamma^*$, $\mu_{\text{real}}((A - \lambda I)^{-1})$.
3: $k = 0$
4: **while** $|b_k - a_k| > \mathsf{tol}_h$ **do**
5: $\quad\gamma_{k+1} = \frac{a_k+b_k}{2}$
6: $\quad$Approximate $\dot{f}_{\text{real}}(\gamma_{k+1})$ using Lanczos($\mathsf{tol}_l$).
7: $\quad$**if** $\dot{f}_{\text{real}}(\gamma_{k+1}) > 0$ **then**
8: $\quad\quad a_{k+1} = a_k$, $b_{k+1} = \gamma_{k+1}$
9: $\quad$**else**
10: $\quad\quad a_{k+1} = \gamma_{k+1}$, $b_{k+1} = b_k$
11: $\quad$**end if**
12: $\quad k = k + 1$
13: **end while**
14: $\gamma^* = \gamma_{k+1}$
15: Compute $\mu_{\text{real}}((A - \lambda I)^{-1}) = \sigma_2(C(\gamma^*))$ (see (5)) using Lanczos($\mathsf{tol}_h$).

### 3.4. Numerical example

Let us illustrate the real structured pseudospectrum for the following matrix by Demmel (1987):

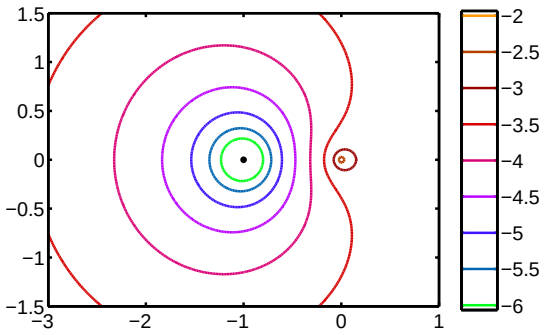
$$A = \begin{bmatrix} -1 & -100 & -10000 \\ 0 & -1 & -100 \\ 0 & 0 & -1 \end{bmatrix}. \tag{15}$$

We used a grid of 100 points in each direction. Figures 2(a) and 2(b) show the unstructured and real structured pseudospectrum, respectively, and reveal striking differences between both. On the first sight, the eigenvalue $-1$ appears to become much less sensitive if the perturbations are restricted to be real. This would contradict first-order sensitivity results in (Kressner et al. 2009), which state that the Hölder condition number of an eigenvalue remains almost the same under real perturbations. Indeed, a closer inspection of the real pseudospectrum reveals that for smaller $\epsilon$ the eigenvalue $-1$ is still moved to almost the same extent but its movement is mainly confined to the six spikes at $-1$. This observation agrees well with the fact that the shape of real pseudospectra for Jordan blocks converges to spikes as $\varepsilon \to 0$ (Karow 2008; Chaitin-Chatelin and Frayssé 1996).

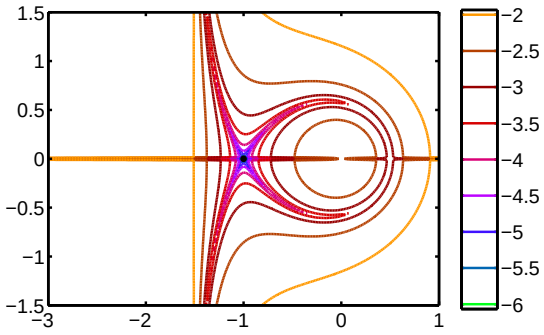
## 4. Skew-symmetric pseudospectra

The techniques described above for accelerating the computation of real structured pseudospectra can be extended to other structures. We first illustrate this for the class $\mathbf{\Delta}$ of (complex) skew-symmetric perturbations. In this case, it can be shown (Karow 2007) that the corresponding structured singular value $\mu_{\text{skew}}(B)$ satisfies the following eigenvalue optimization problem:

$$\mu_{\text{skew}}(B) = \sqrt{\inf_{\gamma\geq 0} \lambda_2\left(\begin{bmatrix} B^*B & \gamma(\overline{B} + B^*) \\ \gamma(B + B^\top) & B^\top\overline{B} \end{bmatrix}\right)}, \tag{16}$$




(a) Unstructured pseudospectrum




(b) Real pseudospectrum

Figure 2: Unstructured (top) and real pseudospectrum (bottom) of the Demmel matrix (15)

where $\lambda_2$ denotes the second largest eigenvalue of a Hermitian matrix. By definition, the skew-symmetric pseudospectrum is the level set of $\mu_{\text{skew}}(B)$, with $B = (A - \lambda I)^{-1}$ as usual. Observe that

$$\begin{bmatrix} B^*B & \gamma(\bar{B} + B^*) \\ \gamma(B + B^\top) & B^\top\bar{B} \end{bmatrix} = \begin{bmatrix} B & \gamma I \\ \gamma I & \bar{B} \end{bmatrix}^* \begin{bmatrix} B & \gamma I \\ \gamma I & \bar{B} \end{bmatrix} - \gamma^2 I, \tag{17}$$

which implies that the objective function is identical to

$$f_{\text{skew}}(\gamma) = \sigma_2^2\left(\begin{bmatrix} B & \gamma I \\ \gamma I & \bar{B} \end{bmatrix}\right) - \gamma^2. \tag{18}$$

Hence, the optimization problem becomes

$$\mu_{\text{skew}}(B) = \sqrt{\inf_{\gamma \geq 0} f_{\text{skew}}(\gamma)}. \tag{19}$$

Again we employ bisection for solving this optimization problem. Recall that each step of bisection requires the sign of the derivative of $f_{\text{skew}}(\gamma)$ with respect to $\gamma$. Differentiating (18) we obtain

$$\dot{f}_{\text{skew}}(\gamma) = 2\sigma_2\left(\begin{bmatrix} B & \gamma I \\ \gamma I & \bar{B} \end{bmatrix}\right)\dot{\sigma}_2\left(\begin{bmatrix} B & \gamma I \\ \gamma I & \bar{B} \end{bmatrix}\right) - 2\gamma, \tag{20}$$

where

$$\dot{\sigma}_2\left(\begin{bmatrix} B & \gamma I \\ \gamma I & \bar{B} \end{bmatrix}\right) = Re\left[u_2^*\begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix}v_2\right], \tag{21}$$

with $u_2, v_2$ the left and right singular vectors belonging to the second largest singular value of $C(\gamma) = \begin{bmatrix} B & \gamma I \\ \gamma I & \bar{B} \end{bmatrix}$.

At each step of bisection we therefore need to compute the second singular triplet of $C(\gamma)$. For this purpose, we apply the Lanczos algorithm to $C(\gamma)^*C(\gamma)$, which requires one matrix-vector multiplication with $C(\gamma)$ followed by another one with $C(\gamma)^*$. This in turn requires matrix-vector multiplications with $B$ or $\bar{B}$. Similarly to the real structured pseudospectrum, we propose to use the Schur decomposition of $A$ (see (14)), which reduces the cost of matrix-vector multiplication with $B$ (or $\bar{B}$) to $O(n^2)$ without the need of actually forming or factorizing $B = (A - \lambda I)^{-1}$.

### 4.1. Numerical example

As an illustrative example, we use the complex skew-symmetric matrix

$$A = \begin{bmatrix} 0 & 1-\phi & 0 \\ -1+\phi & 0 & i \\ 0 & -i & 0 \end{bmatrix}, \quad \phi = 0.01, \tag{22}$$

from (Rump 2006). Figures 3(a) and 3(b) show the unstructured and skew-symmetric pseudospectrum of $A$, respectively, using a grid of 100 points in each direction.

## 5. Hermitian pseudospectra

This section is concerned with the class $\mathbf{\Delta}$ of Hermitian perturbations. Letting $\lambda_1$ denote the largest eigenvalue of a Hermitian matrix, it can been shown (Karow 2007) that the Hermitian structured singular value satisfies

$$\mu_{\text{Hermitian}}(B) = \sqrt{\inf_{\gamma \in \mathbb{R}} \lambda_1\left(B^*B + \gamma i(B - B^*)\right)}, \tag{23}$$

provided that the matrix $B_h = i(B - B^*)$ is not definite. Assuming for the moment that this indefiniteness condition is true, we observe that

$$B^*B + \gamma i(B - B^*) = (B - \gamma i I)^*(B - \gamma i I) - \gamma^2 I, \tag{24}$$

which implies that the objective function is equivalent to

$$f_{\text{Hermitian}}(\gamma) = \sigma_1^2(B - \gamma i I) - \gamma^2. \tag{25}$$

Hence, the optimization problem (23) becomes

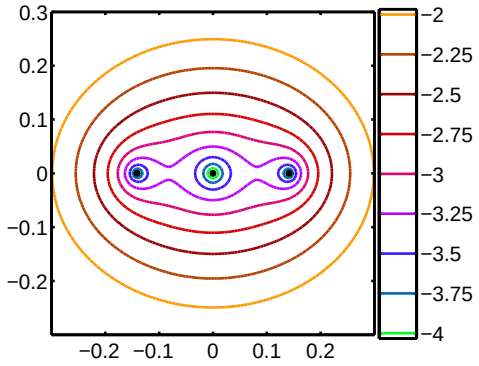
$$\mu_{\text{Hermitian}}(B) = \sqrt{\inf_{\gamma \in \mathbb{R}} f_{\text{Hermitian}}(\gamma)}. \tag{26}$$

Once again, we apply bisection in order to solve this optimization problem, requiring the sign of the derivative of $f_{\text{Hermitian}}(\gamma)$ in each iteration. Differentiating (25) with respect to $\gamma$ gives

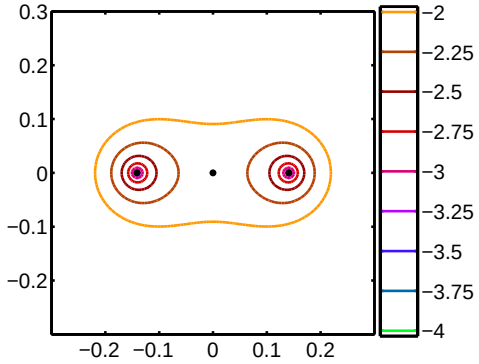
$$f_{\text{Hermitian}}(\gamma) = 2\sigma_1(B - \gamma i I)\dot{\sigma}_1(B - \gamma i I) - 2\gamma, \tag{27}$$

where

$$\dot{\sigma}_1(B - \gamma i I) = Re[-iu_1^*v_1] = Im[u_1^*v_1] \tag{28}$$




(a) Unstructured pseudospectrum




(b) Skew-symmetric pseudospectrum

Figure 3: Unstructured (top) and skew-symmetric pseudospectrum (bottom) of matrix (22)

and $u_1, v_1$ denote the left and right singular vectors belonging to the largest singular value of $C(\gamma) = (B - \gamma iI)$.

Each step of bisection requires the computation of the largest singular triplet of $C(\gamma)$. Applying Lanczos to $C(\gamma)^*C(\gamma)$ requires matrix-vector multiplications with $C(\gamma)$ and $C(\gamma)^*$. As in Section 4, exploiting the Schur decomposition of $A$ (see (14)) reduces the complexity of each matrix-vector multiplication to $O(n^2)$ without the need of forming $B = (A - \lambda I)^{-1}$.

Let us come back to our initial assumption that $B_h = i(B - B^*) = i((A - \lambda I)^{-1} - (A - \lambda I)^{-*})$ is not definite. If this condition is violated, there is no Hermitian perturbation $\triangle$ such that $A + \triangle$ has the eigenvalue $\lambda$ and consequently it is reasonable to define $\mu_{\text{Hermitian}}((A - \lambda I)^{-1}) = 0$. This can be seen as follows. Assuming that $B_h$ is definite, a basic linear algebra result (Horn and Johnson 1985) implies that also $i((A - \lambda I) - (A - \lambda I)^*)$ is definite. If there was a Hermitian perturbation $\triangle$ such that $\lambda$ becomes an eigenvalue with eigenvector $x \neq 0$ of $A + H$ then $(A+H)x = \lambda x$ implies $0 = x^*(A-\lambda I+H)x$. Taking the imaginary part of the last expression gives $ix^*((A - \lambda I) - (A - \lambda I)^*)x = 0$, which contradicts the definiteness assumption.

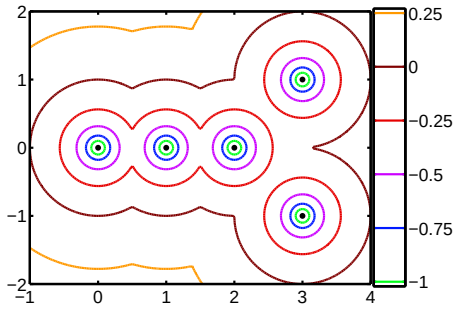
In practice, we test the definiteness of $B_h$ by applying a few steps of the Lanczos method to $B_h$. This will quickly yield good approximations to the extremal (real) eigenvalues of $B_h$. Then, we check whether the signs of the smallest and largest eigenvalue approximations of $B_h$ agree. If this is the case then $B_h$ is considered definite and a zero structured singular value is returned. Otherwise, the computation is continued by solving the optimization problem (26) using bisection. If, for some reason, the definitess test fails and returns a false positive then (26) does not have a minimum, which in turn leads to a failure of the bisection algorithm.

### 5.1. *Numerical example*

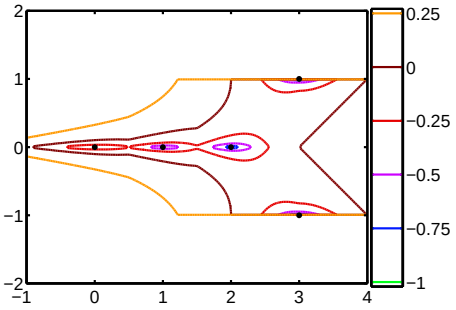
As an example, consider the non-Hermitian matrix

$$A = \text{diag}\left(0, 1, 2, \begin{pmatrix} 3 & -1 \\ 1 & 3 \end{pmatrix}\right), \tag{29}$$

having the eigenvalues $\{0, 1, 2, 3 + i, 3 - i\}$. Figures 4(a) and 4(b) show the unstructured and Hermitian pseudospectrum of $A$, respectively, using a grid of 100 points in each direction. Notice that above the imaginary line 1 and below the imaginary line $-1$ the Hermitian pseudospectrum is void due to fact that for a given complex shift $\lambda \in \mathbb{C}$, with $|Im(\lambda)| > 1$, the matrix $B_h$ becomes definite.




(a) Unstructured pseudospectrum




(b) Hermitian pseudospectrum

Figure 4: Unstructured (top panel) and Hermitian pseudospectrum (middle and bottom panel) of matrix (29)

## 6. Hamiltonian pseudospectra

Following the discussion in (Karow et al. 2006a; Karow 2007), linear structures induced by bilinear and sesquilinear

forms can be handled by structured pseudospectra computation for symmetric, skew-symmetric, and Hermitian perturbations. In the following, this will be demonstrated for the practically relevant case of Hamiltonian perturbations.

A matrix $H$ is called Hamiltonian if it takes the form

$$H = \begin{bmatrix} A & B \\ C & -A^* \end{bmatrix} \in \mathbb{C}^{2n\times 2n},\ B = B^*,\ C = C^*.$$

The eigenvalues of $H$ are symmetric with respect to the imaginary axis, and so is its Hamiltonian pseudospectrum. Defining $J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}$ it is easy to see that $H$ is Hamiltonian if and only if $JH$ is Hermitian. For a general matrix $B \in \mathbb{C}^{2n\times 2n}$ we therefore obtain

$$\mu_{\text{Hamiltonian}}(B) = \mu_{\text{Hermitian}}(JB).$$

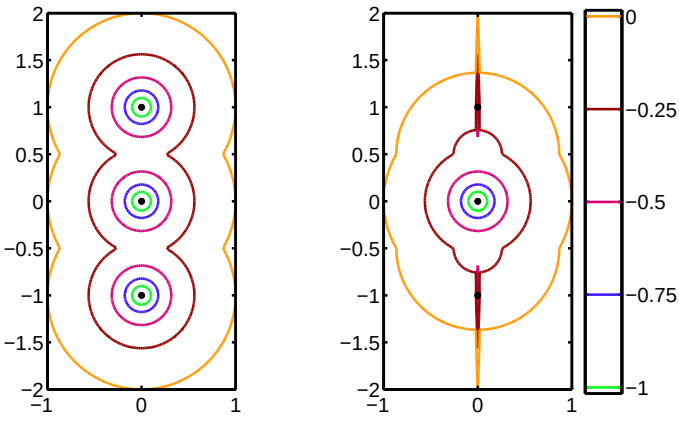
and hence the Hamiltonian pseudospectrum can be computed by solving the optimization problem (26) for each grid point with $B$ replaced by $JB$.

### 6.1. *Numerical examples*

The following two examples illustrate the use of Hamiltonian pseudospectra to provide insight into the movement of purely imaginary eigenvalues under Hamiltonian perturbations. In both examples we use a grid of 100 points in each direction. First, consider the matrix

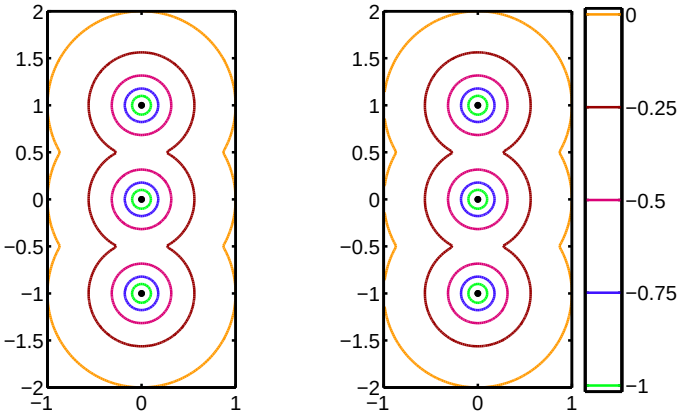
$$A = \begin{bmatrix} 0 & \text{diag}(0,1,1) \\ \text{diag}(0,-1,-1) & 0 \end{bmatrix} \tag{30}$$

with eigenvalues $\{0, i, -i\}$, each having algebraic multiplicity two. Figure 5(a) and 5(b) show the unstructured and Hamiltonian pseudospectrum of $A$, respectively. The latter reveals that the eigenvalues $\pm i$ stay on the imaginary axis under Hamiltonian perturbations up to the point when they meet the perturbed zero eigenvalue, which allows them to leave the imaginary axis.



(a) Unstructured pseudospectrum (b) Hamiltonian pseudospectrum

Figure 5: Pseudospectra of the matrix (30)



(a) Unstructured pseudospectrum (b) Hamiltonian pseudospectrum

Figure 6: Pseudospectra of the matrix (31)

Next, consider the following matrix $A$

$$A = \begin{bmatrix} 0 & \text{diag}(0,-1,1) \\ \text{diag}(0,1,-1) & 0 \end{bmatrix}, \tag{31}$$

where we have simply switched the signs of the ones. This matrix has the same eigenvalues $\{0, i, -i\}$ as (30). Figures 6(a) and 6(b) show the unstructured and Hamiltonian pseudospectrum of $A$ respectively. In remarkable contrast to Figure 5, there is virtually no difference between both pseudospectra. In particular, the eigenvalues $\pm i$ may leave the imaginary axis for arbitrarily small Hamiltonian perturbations. A theoretical explanation for this effect of the sign change can be found, for example, in (Grivet-Talocia 2004; Mehrmann and Xu 2008).

## 7. A MATLAB interface for plotting structured pseudospectra

The algorithms described in this paper have been implemented into a software package for computing structured pseudospectra.[1] For convenience, Table 1 summarizes the SVD calculations required in the computation of structured pseudospectra, depending on the perturbation class $\Delta$.

Our software inherits the complete interface of EigTool (Wright 2002), with a few modifications to allow the specification of the perturbation structure, see Figure 7. In its present state, this software should be understood as research code with potential for further improvements. In particular, all computations are currently implemented as MATLAB code. Significant speedup could be obtained by outsourcing low-level computations (e.g., the Lanczos method) to a low-level programming language, as done in EigTool. Moreover, in view of the non-smooth boundaries of structured pseudospectra, it would be advantageous to use an adaptive grid, see (Breda et al. 2009) for work in this direction.

[1] See `http://www.sam.math.ethz.ch/NLAgroup/software.html`.

| Real | Skew-symmetric |
|---|---|
| $\sigma_2\left(\begin{bmatrix} \mathcal{R}B & -\gamma^{-1}\mathcal{I}B \\ \gamma\mathcal{I}B & \mathcal{R}B \end{bmatrix}\right)$ | $\sigma_2\left(\begin{bmatrix} B & \gamma I \\ \gamma I & B \end{bmatrix}\right)$ |
| Hermitian | Hamiltonian |
| $\sigma_{\max}(B-\gamma iI)$ | $\sigma_{\max}(JB-\gamma iI)$ |

Table 1: SVD calculations involved in structured pseudospectra.

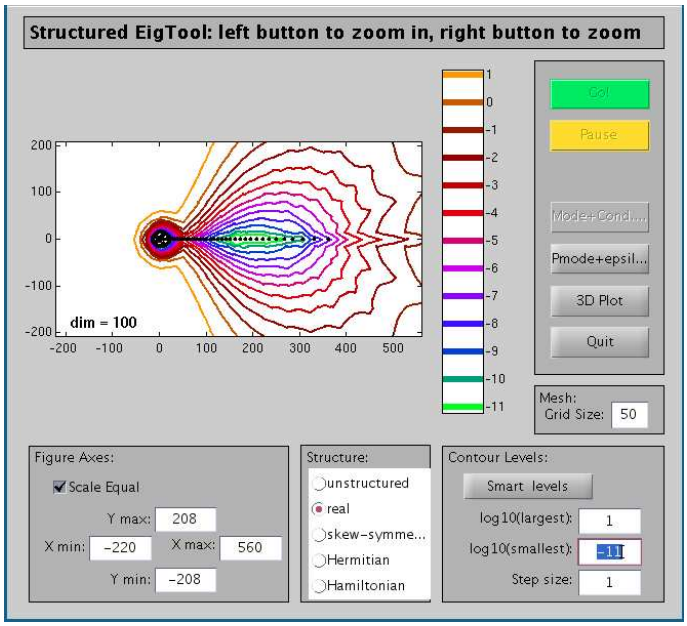



Figure 7: MATLAB interface for computing structured pseudospectra . The plot shows the real structured pseudospectra of the $100 \times 100$ Frank matrix (Trefethen and Embree 2005).

Note that the non-smoothness of structured pseudospectra may pose a particular challenge for grid-based algorithms. In particular, structured pseudospectra may partially collapse to intervals on the real or imaginary line. We deal with this situation by enforcing the real axis (in the case of Hermitian and real perturbations) and the imaginary axis (in the case of Hamiltonian perturbations) to be part of the computational grid. Let us emphasize, however, that there might be other (nearly) one-dimensional structures that are not well captured by a uniform grid. For example, the spikes in Figure 2 are not well resolved; at a finer resolution one can observe small isolated patches along the spikes, which are artifacts due to the contour interpolation used by MATLAB. This could be avoided by applying an edge detection algorithm (Canny 1983), refining the grid along the edges, and using a different contour interpolation.

To give an impression of the performance of the current implementation: On a single core of a 2.2GHz Intel 2 Core processor, computing the real structured pseudospectra displayed in Figure 7 requires about 7 minutes with the algorithms described in the paper. In comparison, a naive implementation based on full SVD computations instead of the Lanczos method requires about 79 minutes. However, it should also be mentioned that

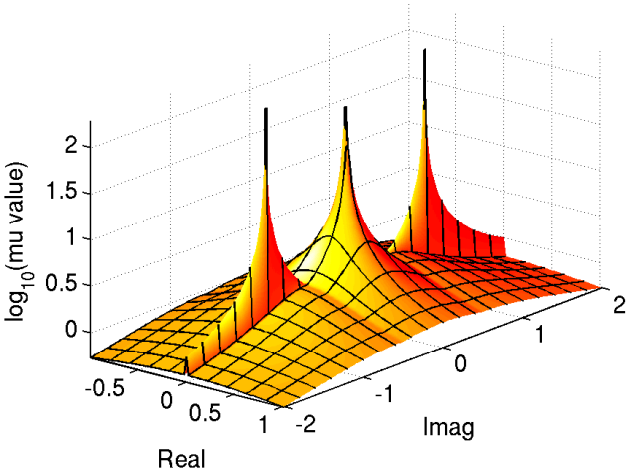



Figure 8: $\mu$ values for Hamiltonian perturbations of the Hamiltonian matrix (30).

computing the unstructured pseudospectra of the same matrix requires only a few seconds.

Finally, we mention that our software also provides the possibility – inherited by EigTool – to display 3D plots of the $\mu$ values. In certain situations this might provide more intuitive visualization than contour plots. For example, Figure 8 displays such a 3D plot for the Hamiltonian pseudospectrum shown in Figure 5(b).

## 8. Acknowledgment

We are grateful for the generosity of the developers and maintainers of the EigTool package allowing to reuse the graphical interface and infrastructure of EigTool for our implementation. We thank Mark Embree and a referee for suggesting improvements to an earlier version of this paper. The first author acknowledges the hospitability of the FIM, ETH Zurich, where he was staying while part of the work on this paper was performed.

## References

T. G. Wright, EigTool, see `http://www.comlab.ox.ac.uk/pseudospectra/eigtool/`, 2002.

L. N. Trefethen, M. Embree, Spectra and Pseudospectra. The Behavior of Nonnormal Matrices and Operators, Princeton University Press, Princeton, NJ, 2005.

Z. Bai, J. W. Demmel, J. J. Dongarra, A. Ruhe, H. van der Vorst (Eds.), Templates for the Solution of Algebraic Eigenvalue Problems, Software, Environments, and Tools, SIAM, Philadelphia, PA, 2000.

G. H. Golub, C. F. Van Loan, Matrix Computations, Johns Hopkins University Press, Baltimore, MD, third edn., 1996.

S. M. Rump, Eigenvalues, pseudospectrum and structured perturbations, Linear Algebra Appl. 413 (2-3) (2006) 567–593.

D. S. Mackey, N. Mackey, F. Tisseur, Structured mapping problems for matrices associated with scalar products. I. Lie and Jordan algebras, SIAM J. Matrix Anal. Appl. 29 (4).

S. M. Rump, Structured perturbations. I. Normwise distances, SIAM J. Matrix Anal. Appl. 25 (1) (2003) 1–30.

S. Graillat, A note on structured pseudospectra, J. Comput. Appl. Math. 191 (1) (2006) 68–76.
M. Karow, $\mu$-values and spectral value sets for linear perturbation classes defined by a scalar product, submitted, 2007.
M. Karow, D. Kressner, F. Tisseur, Structured eigenvalue condition numbers, SIAM J. Matrix Anal. Appl. 28 (4) (2006a) 1052–1068.
L. Qiu, B. Bernhardsson, A. Rantzer, E. J. Davison, P. M. Young, J. C. Doyle, A formula for computation of the real stability radius, Automatica J. IFAC 31 (6) (1995) 879–890.
J. Doyle, Analysis of feedback systems with structured uncertainties, Proc. IEE-D 129 (6) (1982) 242–250.
M. Karow, D. Hinrichsen, A. J. Pritchard, Interconnected systems with uncertain couplings: explicit formulae for $\mu$-values, spectral value sets, and stability radii, SIAM J. Control Optim. 45 (3) (2006b) 856–884.
G. Pappas, D. Hinrichsen, Robust stability of linear systems described by higher order dynamic equations, IEEE Trans. Automat. Control 38 (9) (1993) 1430–1435, ISSN 0018-9286.
C. B. Soh, C. S. Berger, K. P. Dabke, On the stability properties of polynomials with perturbed coefficients, IEEE Trans. Automat. Control 30 (10) (1985) 1033–1036, ISSN 0018-9286.
F. Tisseur, N. J. Higham, Structured pseudospectra for polynomial eigenvalue problems, with applications, SIAM J. Matrix Anal. Appl. 23 (1) (2001) 187–208.
J. W. Demmel, The componentwise distance to the nearest singular matrix, SIAM J. Matrix Anal. Appl. 13 (1) (1992) 10–19, ISSN 0895-4798.
S. Poljak, J. Rohn, Checking robust nonsingularity is NP-hard, Math. Control Signals Systems 6 (1) (1993) 1–9, ISSN 0932-4194.
G. J. Balas, J. C. Doyle, K. Glover, A. Packard, R. Smith, $\mu$-Analysis and Synthesis ToolBox, Natick, MA: Mathworks, 1993.
A. Packard, J. Doyle, The complex structured singular value, Automatica J. IFAC 29 (1) (1993) 71–109.
J.-G. Sun, A note on simple nonzero singular values, J. Comput. Math. 6 (3) (1988) 258–266.
A. S. Lewis, H. S. Sendov, Nonsmooth analysis of singular values. I. Theory, Set-Valued Anal. 13 (3) (2005) 213–241.
J. W. Demmel, A counterexample for two conjectures about stability, IEEE Trans. Automat. Control 32 (1987) 340–342.
D. Kressner, M. J. Peláez, J. Moro, Structured Hölder condition numbers for multiple eigenvalues, SIAM J. Matrix Anal. Appl. 31 (1) (2009) 175–201.
M. Karow, Structured pseudospectra for small perturbations, MATHEON preprint 530, MATHEON Berlin, electronically available from `http://www.matheon.de/research/list_preprints.asp`, 2008.
F. Chaitin-Chatelin, V. Frayssé, Lectures on finite precision computations, Software, Environments, and Tools, SIAM, Philadelphia, PA, 1996.
R. A. Horn, C. R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.
S. Grivet-Talocia, Passivity enforcement via perturbation of Hamiltonian matrices, IEEE Trans. Circuits Syst. I Regul. Pap. 51 (9) (2004) 1755–1769.
V. Mehrmann, H. Xu, Perturbation of purely imaginary eigenvalues of Hamiltonian matrices under structured perturbations, Electron. J. Linear Algebra 17 (2008) 234–257.
D. Breda, S. Maset, R. Vermiglio, An adaptive algorithm for efficient computation of level curves of surfaces, to appear in Numerical Algorithms, 2009.
J. F. Canny, Finding edges and lines in images, TR-720, M.I.T. Artif. Intell. Lab., Cambridge, MA, 1983.