BIT 36:3 (1996), 422–440.

COMPUTING THE FIELD OF VALUES AND PSEUDOSPECTRA USING THE LANCZOS METHOD WITH CONTINUATION*

THIERRY BRACONNIER and NICHOLAS J. HIGHAM †

Department of Mathematics, University of Manchester, Manchester, M13 9PL, England
email: braconni@ma.man.ac.uk, na.nhigham@na-net.ornl.gov.

Abstract.

The field of values and pseudospectra are useful tools for understanding the behaviour of various matrix processes. To compute these subsets of the complex plane it is necessary to estimate one or two eigenvalues of a large number of parametrized Hermitian matrices; these computations are prohibitively expensive for large, possibly sparse, matrices, if done by use of the QR algorithm. We describe an approach based on the Lanczos method with selective reorthogonalization and Chebyshev acceleration that, when combined with continuation and a shift and invert technique, enables efficient and reliable computation of the field of values and pseudospectra for large matrices. The idea of using the Lanczos method with continuation to compute pseudospectra is not new, but in experiments reported here our algorithm is faster and more accurate than existing algorithms of this type.

AMS subject classification: 65F15.

Key words: Field of values, pseudospectra, Lanczos method, Chebyshev acceleration, continuation.

1 The field of values and pseudospectra.

Our aim in this work is to develop efficient algorithms for computing the field of values and pseudospectra of large, possibly sparse matrices $A \in \mathbb{C}^{n \times n}$. These sets in the complex plane characterize various aspects of the behaviour of a matrix, as we now explain. For a detailed presentation of the theory of the field of values see [15, Ch. 2], and for an application see [28]. Theory and applications of pseudospectra can be found in [12, 26, 33, 34, 35] and the references therein.

The *field of values* of a matrix $A \in \mathbb{C}^{n \times n}$ is the set of all Rayleigh quotients:

$$F(A) = \left\{ \frac{z^*Az}{z^*z} : 0 \neq z \in \mathbb{C}^n \right\}.$$

The set $F(A)$ is convex, and when A is normal it is the convex hull of the eigenvalues. For a Hermitian matrix $F(A)$ is a segment of the real axis and for

*Received November 1995. Revised April 1996.

†This work was supported by Engineering and Physical Sciences Research Council grants GR/H/52139 and GR/H/94528.

a skew-Hermitian matrix it is a segment of the imaginary axis. Associated with the field of values is the *numerical radius*

$$r(A) = \max\{\,|w| : w \in F(A)\,\},$$

which does not differ much from $\|A\|_2 := \max_{x \neq 0} \|Ax\|_2/\|x\|_2$ since, as is readily shown,

$$\tfrac{1}{2}\|A\|_2 \leq r(A) \leq \|A\|_2. \tag{1.1}$$

For a diagonalizable matrix $A = XDX^{-1}$ we have

$$\frac{r(A)}{\rho(A)} \leq \frac{\|A\|_2}{\|D\|_2} \leq \kappa_2(X),$$

where the spectral radius $\rho(A) = \max\{\,|\lambda| : \lambda \in \lambda(A)\,\}$, with $\lambda(A)$ the set of eigenvalues of A. Thus if the numerical radius is much bigger than the spectral radius then any eigenvector basis of A is ill conditioned, which means that A is "far from normal". (We will not precisely define measures of nonnormality, but we mention that $\min_X \kappa(X)$ can exceed the relative distance from A to the nearest normal matrix by an arbitrary factor [7, §9.1.1], [8, §4.2.7].)

The field of values is connected with logarithmic norms, which are used to study the growth of solutions to ordinary differential equations (ODEs) and the growth of errors in discretization schemes for solving ODEs [31]. A logarithmic norm μ is defined in terms of a given matrix norm by

$$\mu(A) = \lim_{\epsilon \downarrow 0} \frac{\|I + \epsilon A\| - 1}{\epsilon}.$$

It is straightforward to show that $\mu(A) = \alpha_F(A)$ for the 2-norm, where the *numerical abscissa*

$$\alpha_F(A) = \sup_{z \in F(A)} \operatorname{Re} z.$$

The numerical abscissa is, trivially, at least as large as the largest real part of any eigenvalue of A,

$$\alpha(A) = \sup_{z \in \lambda(A)} \operatorname{Re} z,$$

which is called the *spectral abscissa*. The numerical and spectral abscissas play opposite roles in determining the behaviour of $\exp(tA)$: while $\|\exp(tA)\| \to 0$ as $t \to \infty$ if and only if $\alpha(A) < 0$, $\alpha_F(A)$ determines the rate of growth of $\|\exp(tA)\|$ for small positive t [12], [34].

Pseudospectra are parametrized sets in the complex plane, so while there is only one field of values, there are many pseudospectra. For $\epsilon > 0$, the *ϵ-pseudospectrum* of $A \in \mathbb{C}^{n \times n}$ is defined by

$$\Lambda_\epsilon(A) = \{\,z : z \in \lambda(A + \Delta A) \text{ for some } \Delta A \text{ with } \|\Delta A\|_2 \leq \epsilon\,\}.$$

An equivalent definition, in terms of the resolvent $(zI - A)^{-1}$, is

$$\begin{aligned}\Lambda_\epsilon(A) &= \{\, z : \|(zI-A)^{-1}\|_2 \ge \epsilon^{-1} \,\}\\ &= \{\, z : \sigma_{\min}(zI - A) \le \epsilon \,\}, \end{aligned} \tag{1.2}$$

where $\sigma_{\min}$ denotes the smallest singular value. By convention, we define $\|(zI - A)^{-1}\|_2 = \infty$ for $z \in \lambda(A)$. For a normal matrix the ϵ-pseudospectrum comprises the union of the closed balls of radius ϵ about each eigenvalue, but for a nonnormal matrix the ϵ-pseudospectrum can be much larger than the union of these ϵ-balls.

Both the field of values and pseudospectra can be used to bound matrix powers. For $A \in \mathbb{C}^{n\times n}$ we have the trivial bound

$$\|A^k\|_2 \le \|A\|_2^k \tag{1.3}$$

and, for a diagonalizable matrix $A = XDX^{-1}$,

$$\|A^k\|_2 \le \kappa(X)\rho(A)^k. \tag{1.4}$$

Generally sharper than (1.3) is

$$\|A^k\|_2 \le 2r(A)^k,$$

which follows from (1.1) since $r(A^k) \le r(A)^k$ [14, p. 333]. An alternative to (1.4) is the bound [33]

$$\|A^k\|_2 \le \epsilon^{-1}\rho_\epsilon(A)^{k+1},$$

where the *ϵ-pseudospectral radius* ρ_ϵ is defined by

$$\rho_\epsilon(A) = \max\{\, |z| : z \in \Lambda_\epsilon(A) \,\}.$$

2 Standard algorithms.

To determine the field of values we need to find its boundary, which is a one-dimensional computation. The boundary can be computed using an algorithm suggested by Johnson [16]. The algorithm is based on the observation that if we write $A = A_H + A_K$, where $A_H = (A + A^*)/2$ is the Hermitian part of A and $A_K = (A - A^*)/2$ is the skew-Hermitian part of A, then, for any $z \in \mathbb{C}$,

$$z^*Az = \underbrace{z^*A_Hz}_{\text{real}} + \underbrace{z^*A_Kz}_{\text{pure imaginary}},$$

which implies that

$$\min\{\, \mathrm{Re}(w) : w \in F(A) \,\} = \min_{z\neq 0} \frac{z^*A_Hz}{z^*z} = \lambda_{\min}(A_H), \tag{2.1a}$$

$$\max\{\, \mathrm{Re}(w) : w \in F(A) \,\} = \max_{z\neq 0} \frac{z^*A_Hz}{z^*z} = \lambda_{\max}(A_H), \tag{2.1b}$$

where $\lambda_{\min}$ and $\lambda_{\max}$ denote the algebraically smallest and largest eigenvalues, respectively, of a Hermitian matrix. Therefore the field of values lies within the vertical strip defined by the lines parallel to the imaginary axis that intersect the real axis at $\lambda_{\min}(A_H)$ and $\lambda_{\max}(A_H)$; these lines intersect the field of values at points given by the Rayleigh quotient with A of the vectors that achieve the minimum and maximum in (2.1). Since $F(e^{i\theta}A) = e^{i\theta}F(A)$, we can apply the same reasoning to the rotated matrix $A(\theta) = e^{i\theta}A$: by computing $\lambda_{\min}(A_H(\theta))$ and $\lambda_{\max}(A_H(\theta))$ we determine two lines making an angle θ with the imaginary axis between which $F(A)$ lies, and the Rayleigh quotients with A of the corresponding eigenvectors give the intersection points of the lines with $F(A)$. Carrying out the computation for a range of $\theta \in [0, \pi]$ we obtain an approximation to the boundary of the field of values. Adjacent points on the boundary can be joined by straight lines to obtain a graphical approximation to $F(A)$. If A is real then the field of values is symmetric about the real axis and we can restrict θ to the range $[0, \pi/2]$. The routine `fv.m` in the Test Matrix Toolbox for MATLAB [13] implements this algorithm, using the QR algorithm to compute the eigensystems of the Hermitian matrices $A_H(\theta)$ (these matrices are complex even if A is real). Note that effort is wasted by computing the whole eigensystem instead of just the extremal eigenvalues and corresponding eigenvectors, and sparsity cannot be exploited by the QR algorithm.

We mention in passing that an iterative algorithm for estimating the numerical radius without computing the field of values is developed by Watson [37] and He and Watson [11]; the algorithm has connections with the power method.

A discrete approximation to the ϵ-pseudospectrum can be obtained by computing $\lambda(A + \Delta A)$ for a number of random perturbations ΔA with $\|\Delta A\|_2 = \epsilon$; the eigenvalues can be plotted as dots in the complex plane, providing information about the "density" of the ϵ-pseudospectrum as well as its shape. However, we can usually learn more about a matrix by determining level curves of the norm of the resolvent, that is, by determining the set of points z at which $\sigma_{\min}(zI - A) = \epsilon$, for several values of ϵ and for z in a given region of the complex plane. For each ϵ we thus obtain the boundary of $\Lambda_\epsilon(A)$. This is a two-dimensional computation and therefore is inherently more expensive than the computation of the field of values. A natural approach is to impose a grid on the given region of the complex plane and to compute $\sigma_{\min}(zI - A)$ at each grid point z. A graphical plot is obtained by feeding the results to a contour or surface plotter. This is the approach used in the routine `pscont.m` in the Test Matrix Toolbox [13]. If A is real then the level curves are symmetric about the x-axis, since $\sigma_{\min}(zI - A) = \sigma_{\min}(\bar{z}I - A)$; this fact can be used to halve the computational work. The routine `pscont.m` computes the complete singular value decomposition (SVD) of $zI - A$ at each point z using the Golub–Reinsch SVD algorithm. Once again, effort is wasted in computing the whole SVD instead of just the smallest singular value, and sparsity is not exploited.

Alternative approaches for computing pseudospectra have recently been developed that are more efficient for large, sparse matrices; these are summarized in §4. There are two basic ideas. The first is to use a method that computes

$\sigma_{\min}(zI - A)$ without computing the complete SVD. The second idea is to use continuation, so that at each grid point the computation of $\sigma_{\min}(zI - A)$ makes use of information from the computation at a nearby gridpoint.

The purpose of this paper is twofold: to develop an improved algorithm for computing pseudospectra using the Lanczos method and continuation and to develop a similar algorithm for computing the field of values.

3 A Lanczos–Chebyshev algorithm.

To compute the field of values we need to compute the largest and smallest eigenvalues of a Hermitian matrix, while to compute pseudospectra we need to compute the smallest singular value of a general matrix or, equivalently, the smallest eigenvalue of a Hermitian positive semidefinite matrix. Because of the similarity of these tasks our algorithms have a common core, which we describe in Algorithm 3.1 below. The differences between the field of values and pseudospectra computations are described at the end of the section.

Let $B \in \mathbb{C}^{n\times n}$ be Hermitian. Starting from an initial vector u, the Lanczos process[1] applied to B iteratively builds an orthonormal basis $V_m \in \mathbb{C}^{n\times m}$ of the Krylov subspace $K_m(u, B) = \mathrm{span}(u, Bu, \dots, B^{m-1}u)$ and a tridiagonal matrix $T_m \in \mathbb{R}^{m\times m}$ such that

$$BV_m = V_mT_m + t_{m+1,m}v_{m+1}e_m^T, \tag{3.1}$$

where e_j denotes the jth column of the identity matrix. If $T_m = S\,\mathrm{diag}(\mu_i)S^*$ is a spectral decomposition, with $S = [s_1, s_2, \dots, s_m]$ unitary, then the Ritz pairs $(\mu_i, V_m s_i)$, $i = 1{:}\,m$, provide approximations to a desired number $r \le m \ll n$ of eigenpairs of B. To maintain the orthogonality of the Krylov basis V_m we use selective reorthogonalization (see, e.g., [10, §9.2.4]). This step ensures the backward stability of the Lanczos process, in an appropriately defined sense [3].

We use the Lanczos method in conjunction with Chebyshev acceleration to improve the speed of convergence to the desired eigenvalues [30]. The Chebyshev ellipse reduces to an interval because B is Hermitian.

The proposed algorithm can be summarized as follows.

ALGORITHM 3.1.

Parameters: integers r (number of desired eigenpairs) and m (number of Lanczos steps), with $r \le m \ll n$; Lanczos starting vector u; degree k of Chebyshev acceleration polynomial and starting vector z_0.

1. *Perform m steps of the Lanczos process on B with selective reorthogonalization to compute V_m and T_m.*
2. *Compute the Ritz pairs $(\mu_i, y_i)_{i=1:m}$ of B by applying the QR algorithm to T_m.*
3. *If the stopping criterion is satisfied then exit.*
4. *Build the Chebyshev interval.*
5. *Perform Chebyshev acceleration to obtain a better starting vector u for the Lanczos process; go to step* 1.

[1] We use the terminology of Paige [22] that distinguishes between the Lanczos process (construction of the Krylov basis) and the Lanczos method (the Lanczos process together with a method for computing the Ritz pairs from the resulting tridiagonal matrix).

Variations on Algorithm 3.1 are possible. For example, for economy and simplicity of coding one could compute the desired eigenpairs using the production codes from ARPACK [18], which employ the Arnoldi method but can exploit symmetry. However, our purpose is to develop a particular Lanczos implementation for comparison with previously proposed methods that are described in the next section.

For the field of values computation, B is the matrix $A_H(\theta)$ of §2. For the pseudospectra computation we choose to find the largest eigenvalue of $B = ((A-zI)^*(A-zI))^{-1}$ rather than the smallest eigenvalue of $B = (A-zI)^*(A-zI)$. By using this (zero) shift and invert technique we generally obtain a more favourable eigenvalue distribution, since small, clustered eigenvalues of $(A-zI)^*(A-zI)$ map to well separated large eigenvalues. Of course, this approach requires us to solve linear systems with coefficient matrices $A-zI$ and its conjugate transpose. As an alternative to the Lanczos method we could use the power method for the pseudospectra computation; reasons for preferring the Lanczos method are given by Parlett, Simon and Stringer [23] and Kuczyński and Woźniakowski [17].

It is not necessary to compute all the Ritz values in step 2 of Algorithm 3.1; we need only the largest and, possibly, the smallest. Parlett, Simon and Stringer [23] give an efficient algorithm for determining the extremal Ritz values at each step of the process. However, since $m \ll n$, the cost of step 2 in our algorithm is negligible.

Before discussing the choice of the parameters and the stopping criterion we explain the use of continuation in Algorithm 3.1.

Consider the pseudospectra computation, with a grid imposed on the region of interest in the complex plane. At each grid point we have to compute the smallest eigenvalue λ_1 of a Hermitian positive semidefinite matrix. If, at a given grid point, we obtain λ_1 and a corresponding eigenvector x_1, then as the Lanczos starting vector for Algorithm 3.1 at the next grid point we take $u = x_1$. Some justification for this continuation strategy is given by Lui [19]. For the field of values computation, if, for a given angle θ, we obtain extremal eigenvalues λ_1 and λ_n and corresponding eigenvectors x_1 and x_n, then our starting vector for the next value of θ is $u = (x_1 + x_n)/\|x_1 + x_n\|_2$. By choosing the starting vectors in this way we hope to obtain fast convergence and to minimize the need for restarts in Algorithm 3.1.

We take the Ritz values μ_i and the eigenvalues $\lambda_i = \lambda_i(B)$ to be ordered so that $\mu_1 \le \mu_2 \le \cdots \le \mu_m$ and $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$.

Choice of m. The Kaniel–Saad theory suggests that we take m close to $2\sqrt{n}$ for the classical Lanczos method (see, e.g., [24, p. 259]); this we do for the field of values computation. For the shift and invert technique used in the pseudospectra computation we take m close to $\sqrt{n}/2$, which our experience has shown is usually sufficient.

Choice of stopping criterion. We define the normwise backward error associated with the approximate eigenpair (λ, x) of the matrix B to be

$$\eta = \min\{\, \epsilon > 0 : \|\Delta B\|_2 \le \epsilon \|B\|_2,\ (B + \Delta B)x = \lambda x \,\}.$$

From a standard result on linear system backward errors [27], or via a simple

Table 3.1: Chebyshev interval and z_0.

Case	d, a	z_0
1	$d = (\mu_2 + \mu_m)/2$, $a = \max(d - \mu_2, \mu_m - d)$	$z_0 = V_m s_1$
2	$d = (\mu_1 + \mu_{m-1})/2$, $a = \max(d - \mu_1, \mu_{m-1} - d)$	$z_0 = V_m s_m$
3	$d = (\mu_2 + \mu_{m-1})/2$, $a = \max(d - \mu_2, \mu_{m-1} - d)$	$z_0 = V_m(s_1 + s_m)$

computation, we find that

$$\eta = \frac{\|r\|_2}{\|B\|_2 \|x\|_2},$$

where $r = Bx - \lambda x$. For any Ritz vector y_i and its corresponding Ritz value μ_i we can compute η without forming r, because, from (3.1), we find that $\|r\|_2 = |t_{m+1,m}||s_{mi}|$, where $T_m = S\,\mathrm{diag}(\mu_i)S^*$ is a spectral decomposition. As suggested by Bennani and Braconnier [2], our stopping criterion for Algorithm 3.1 is that the "Lanczos backward error" η_L is less than or equal to the unit roundoff, where

$$\eta_L = \frac{|t_{m+1,m}||s_{mi}|}{\|B\|_2}$$

and the ith Ritz pair approximates the eigenpair of interest. In forming η_L, we approximate $\|B\|_2$ by $\max_i |\mu_i|$, which is a lower bound for $\|B\|_2$ [10, pp. 480–481]. If this lower bound is weak then we overestimate η_L, which is acceptable since Algorithm 3.1 has not converged, while close to convergence the lower bound is, necessarily, a good approximation to $\|B\|_2$. For the field of values computation the convergence test in Algorithm 3.1 is declared to be passed only if η_L does not exceed the tolerance for both the desired computed eigenpairs.

Chebyshev interval and z_0. There are three choices of Chebyshev interval and starting vector, corresponding to the eigenvalues we want to compute. Case 1 corresponds to the computation of λ_1, case 2 to the computation of λ_n and case 3 to the computation of λ_1 and λ_n. Let d be the centre and a the "real semiaxis" of the Chebyshev interval. Table 3.1 gives the values of these parameters for each case. For the computation of $\Lambda_\epsilon(A)$, we have adopted the strategy of case 2 because we use the shift and invert transformation. For the computation of $F(A)$, during the first iterations the strategy of case 3 is adopted. When λ_1 (respectively λ_n) has satisfied the stopping criterion, then we switch to the strategy of case 2 (respectively case 1).

Degree k of Chebyshev polynomial. The choice of the degree k of the polynomial used in Chebyshev acceleration is discussed by Saad [30], who suggests a value depending on the convergence tolerance and certain "convergence ratios". We take $k = 25$, which appears to work well for our application.

4 Other algorithms for computing pseudospectra.

We briefly compare our algorithm for computing pseudospectra with those of other authors. As a base reference we take the "SVD algorithm" described in

§2, which computes the full SVD of $A - zI$ at each gridpoint z.

Carpraux, Erhel and Sadkane [6] use the Davidson method with preconditioning and continuation. They give limited numerical results without timings, and their algorithm has difficulty computing small values of $\sigma_{\min}(A - zI)$.

Marques and Toumazou [20] use the Lanczos method with reorthogonalization to compute $\lambda_{\min}((A - zI)^*(A - zI))$ at each point on the grid. Their stopping criterion, like ours, is based on the backward error. They report results both with and without continuation, and observe convergence problems when using continuation. In [21] the same authors apply the Lanczos method with reorthogonalization to the matrix

$$\begin{bmatrix} 0 & A - zI \\ (A - zI)^* & 0 \end{bmatrix}^{-1}$$

in order to compute $\sigma_{\min}(A - zI)$; they do not use continuation. They obtain plots of comparable quality to those from the SVD method.

Lui [19] presents several related methods. For large, possibly sparse matrices, which are our interest here, he proposes two algorithms. The first applies inverse iteration to $(A - zI)(A - zI)^*$. The second applies the Lanczos method to the same matrix (the use of reorthogonalization is not mentioned in [19]). In both cases continuation is used and the iteration at each grid point is terminated when (an estimate of) the relative error in the computed eigenvalue of interest is at most 0.1. Lui obtains plots visually indistinguishable from those produced by the SVD algorithm at much lower cost, but observes occasional large relative errors. We believe that Lui's relative error-based convergence test with its large tolerance can lead to misconvergence of the Lanczos method, as discussed by Parlett [24] and van der Sluis and van der Vorst [36] (see the numerical examples in §5.2).

A significant difference between our algorithm and those of Marques and Toumazou and of Lui is that their algorithms compute the Ritz pairs and test for convergence on every step of the Lanczos process, whereas our algorithm tests for convergence only after a full m steps. A difficulty with testing the backward error on every Lanczos step is that the test may be passed in the early stages by Ritz pairs that approximate eigenpairs other than the desired extremal ones; an example of this phenomenon is given at the end of §5.

Toh and Trefethen [32] apply the Arnoldi process to A for a certain number of steps and approximate the pseudospectra of A by those of the resulting Hessenberg matrix. They also approximate the field of values of A by that of the Hessenberg matrix. Any of the methods described here can be used for the computations with the Hessenberg matrix.

An alternative to the grid approach is developed by Brühl [5], for situations in which only a few ϵ-pseudospectra are required. He computes the boundary of $\Lambda_\epsilon(A)$ for a given ϵ by curve tracing techniques, using a prediction-correction scheme in which the correction is determined by a Newton step.

Finally, we mention that Ruhe [29] investigates computation of the pseudospectrum of a matrix pencil using the rational Krylov algorithm.

5 Numerical experiments.

We investigate the speed and reliability of our proposed Lanczos–Chebyshev algorithms, comparing them with the standard algorithms described in §2 based on the QR algorithm (for the field of values) and the SVD (for pseudospectra). All the computations were performed in MATLAB 4.2 on a Sun Sparcstation 10; the unit roundoff is $2^{-53} \approx 1.11 \times 10^{-16}$. For the standard algorithms we use the routines `fv.m` and `pscont.m` from the Test Matrix Toolbox [13].

5.1 The Field of Values

The first example is the `circul` matrix of size $n = 100$ from the Test Matrix Toolbox [13], which is the circulant matrix whose first row comprises the integers from 1 to n. Since this matrix is normal, its field of values is the convex hull of its eigenvalues. In fact, the boundary of the field of values is a triangle, with one side parallel to the imaginary axis, so errors should be relatively easy to see in the plot.

The second example is the `tolosa` matrix of size 400 from the Harwell–Boeing sparse matrix collection [9] and Bai's test matrix collection [1]. This matrix arises in an aircraft flutter analysis problem modelled at Aerospatiale Avion, Toulouse [4]; it is the eigenvalues having the largest imaginary parts that are of interest to the engineers. Since the `tolosa(400)` matrix is sparse (approximately 1.4% of its elements are nonzero), we expect a large speedup compared with the QR-based algorithm.

For all the field of values computations we took 16 equally spaced angles. Figures 5.1 and 5.2 plot the fields of values for the Lanczos–Chebyshev algorithm, with the eigenvalues marked as crosses; the plots produced by the QR-based algorithm (not shown) are visually indistinguishable.

Table 5.1 gives the execution times and the number of flops (obtained using MATLAB's `etime` and `flops` commands) for the methods on these two examples. For the full matrix `circul` the ratio of execution times T_{QR}/T_{LC} is of order 10, while for the sparse `tolosa` matrix the ratio is of order 100; the ratios of flop counts are of the same order of magnitude. In these two examples, then, our Lanczos–Chebyshev algorithm is significantly faster than the standard QR-based algorithm, while being just as accurate to "visual precision".

To investigate the benefits of continuation we ran our algorithm on the matrices `circul(100)` and `chebvand(200)`, using 16 and 64 equally spaced angles respectively. The `chebvand` matrix is a Chebyshev–Vandermonde matrix from the Test Matrix Toolbox; the field of values of `chebvand(200)` is plotted in Figure 5.3. We ran the algorithm both with and without continuation and looked at the number of "Chebyshev restarts" required. For the `circul` matrix, 61 restarts were required without continuation and only 3 with continuation; for the `chebvand` matrix the corresponding figures are 65 and 63. Thus the improvement using continuation is problem-dependent. For the `circul` matrix the smallest and largest eigenvectors of the matrix $A_H(\theta)$ change only slowly from one angle to the next, but the change is much more rapid for the `chebvand` matrix.

Table 5.1: Times T (seconds) and flop counts F for the computation of $F(A)$ by the QR method and the Lanczos–Chebyshev algorithm.

Matrix	T_{QR}	T_{LC}	T_{QR}/T_{LC}	F_{QR}	F_{LC}	F_{QR}/F_{LC}
circul(1: 100)	93.7	9.75	9.6	6.05×10^8	2.32×10^7	26.1
tolosa(400)	6183.8	62.0	99.7	1.26×10^{10}	9.99×10^7	126.1

To show the influence of the chosen tolerance on the accuracy and the computing time, we ran our algorithm on the chebvand(200) matrix for a range of values of the tolerance. Table 5.2 summarizes the results. We measure the accuracy in two ways: by the relative differences

$$e_{\mathrm{diff}} = \frac{\|e_{LC} - e_{QR}\|_2}{\|e_{QR}\|_2}, \qquad z_{\mathrm{diff}} = \frac{\|z_{LC} - z_{QR}\|_2}{\|z_{QR}\|_2},$$

where e is a real vector comprising all the extremal eigenvalues of the $A_H(\theta)$ matrices, and z is a vector of complex numbers made up of the approximations to boundary points of the field of values. We recall the result [24, p. 69] that if B is symmetric then, for any scalar α,

$$\min_i |\lambda_i(B) - \alpha| \le \frac{\|r\|_2}{\|x\|_2}, \qquad r = Bx - \alpha x. \tag{5.1}$$

Our backward error stopping criterion bounds $\|r\|_2/(\|B\|_2\|x\|_2)$, with (α, x) a Ritz pair, so e_{diff} is necessarily no larger than approximately $\|B\|_2$ times the tolerance. It may happen that $z_{\mathrm{diff}} \gg e_{\mathrm{diff}}$, but, if so, the quality of the plot is not necessarily affected, since any computed Rayleigh quotient must belong to $F(A)$ to within a bounded absolute error. For all the values of the tolerance, our algorithm is faster than the QR method ($T_{QR} = 763.65$). The values of e_{diff} in this example are approximately the square of the tolerance, which is better than would be expected from (5.1). An explanation is given by a result from [24, p. 222]: for a symmetric matrix B, if α is the Rayleigh quotient of x and μ is the eigenvalue closest to α, then

$$|\mu - \alpha| \le \gamma^{-1}\|r\|_2^2, \qquad r = Bx - \alpha x, \quad \gamma = \min_{\lambda_i(B) \ne \mu} |\lambda_i(B) - \alpha|.$$

Again, we can take (α, x) to be a Ritz pair, because the Lanczos method is an orthogonal projection method, and the gap γ is not small for the chebvand matrix. To achieve the same accuracy as the QR method a tolerance of 10^{-8} suffices for the Lanczos–Chebyshev algorithm in this example. For larger tolerances the run time is smaller, because less work is required, but the computed field of values is less accurate than the one computed using the QR method.

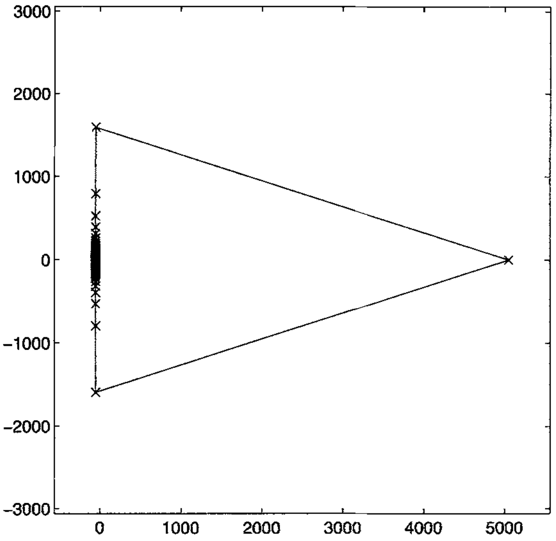

Figure 5.1: Computation of $F($circul(100)$)$ by the Lanczos–Chebyshev algorithm.

5.2 Computation of Pseudospectra

We give two examples of computation of pseudospectra. The first matrix is the tolosa matrix of size 135, which we choose because it caused difficulties for the algorithm of Marques and Toumazou [20] when used with continuation. The region of the complex plane of interest is

$$\{\, z = a + ib : -300 \le a \le 200,\ -250 \le b \le 250 \,\},$$

with a meshsize equal to 1/32 in each direction. The second example is taken from Reddy and Trefethen [26]. The matrix condif is the result of the dis-

Table 5.2: Experiments on chebvand(200); times T are in seconds.

Tolerance	e_{diff}	z_{diff}	T_{LC}	T_{QR}/T_{LC}	No. of restarts
10^{-16}	4.9×10^{-15}	2.7×10^{-15}	256.4	3.0	64
10^{-14}	4.9×10^{-15}	7.5×10^{-14}	236.8	3.2	56
10^{-12}	4.7×10^{-15}	3.1×10^{-12}	207.7	3.7	48
10^{-10}	5.1×10^{-15}	9.7×10^{-11}	183.6	4.2	39
10^{-8}	3.9×10^{-15}	1.4×10^{-8}	159.2	4.8	32
10^{-6}	5.6×10^{-13}	4.1×10^{-7}	125.4	6.1	22
10^{-4}	5.1×10^{-9}	3.2×10^{-5}	103.4	7.4	15
10^{-2}	2.7×10^{-3}	5.7×10^{-2}	53.5	14.3	4

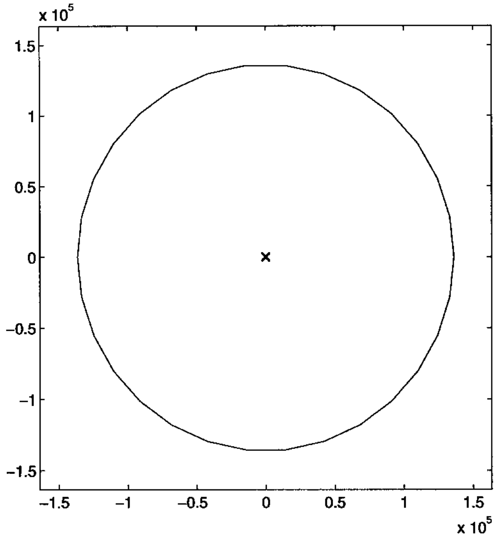

Figure 5.2: Computation of F(`tolosa(400)`) by the Lanczos–Chebyshev algorithm.

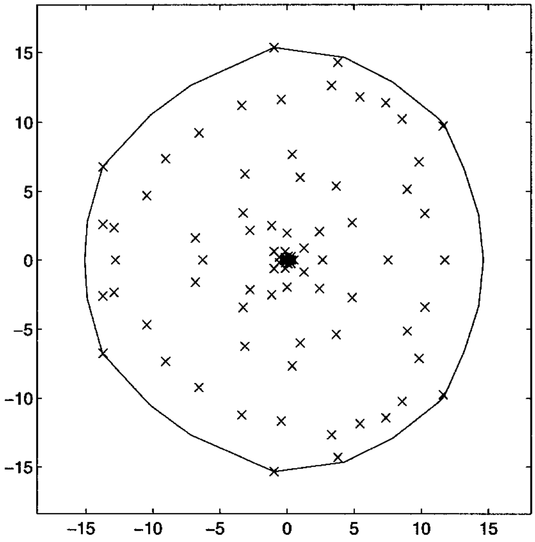

Figure 5.3: Computation of F(`chebvand(200)`) by the Lanczos–Chebyshev algorithm.

cretization of the one-dimensional convection–diffusion operator T defined in the Hilbert space $L^2(0,d)$ by

$$T(u) = \ddot{u} + \dot{u}, \qquad u(0) = u(d) = 0,$$

using central differences at the Chebyshev points

$$x_i = \tfrac{1}{2}d(1 - \cos((i\pi)/n), \quad i = 1\colon n.$$

The associated matrix is tridiagonal. The spectrum of the operator is given by $\lambda_i = -1/4 - (\pi^2 i^2)/d^2$, $i = 1, 2, \dots$, but, as explained in [26], the pseudospectra are large regions in the left half-plane shaped approximately like parabolas. For the experiments, we have chosen $n = 400$ and $d = 4000$, and we focus on the region

$$\{\, z = a + ib : -0.5 \le a \le 0.1,\ -0.3 \le b \le 0.3 \,\},$$

with a 32×32 mesh.

The computations have been done using the Lanczos–Chebyshev algorithm (using MATLAB's backslash operator to solve linear systems involving $A - zI$ and its conjugate transpose), the SVD method, and our MATLAB implementation of Lui's algorithm based on the classical Lanczos method with reorthogonalization but without shift and invert, using continuation and a stopping criterion based on a relative error bound of 0.1, as described in [19]. Our implementations exploit symmetry of the pseudospectra about the x-axis.

For each computation we plot the level curves of $\sigma_{\min}(A - zI)$, which trace out boundaries of pseudospectra $\Lambda_\epsilon(A)$, on a logarithmic scale. The contour lines in the plots of $\Lambda_\epsilon(A)$ represent

- $\epsilon = 10^{-1.5}, 10^{-1.25}, 10^{-1}, \dots, 10^{0.5}$ for the `tolosa` matrix, and
- $\epsilon = 10^{-16}, 10^{-14}, \dots, 10^{-2}$ for the `condif` matrix.

For the `tolosa` matrix, from Figure 5.4 we can see that the pseudospectra computed using the SVD algorithm and the Lanczos–Chebyshev algorithm are in good agreement. On the other hand the pseudospectra computed using Lui's algorithm (Figure 5.5) show significant qualitative differences from those computed by the SVD algorithm. We think that this difference is due to misconvergence problems illustrated in [25], [36], stemming from the stopping criterion used.

For the `condif(400)` matrix the pseudospectra computed using the SVD method and the Lanczos–Chebyshev algorithm are similar (Figure 5.6), but the pseudospectra computed using Lui's algorithm (Figure 5.7) are again unsatisfactory, for the same reason as in the first example.

We can see from Table 5.3 that for the `tolosa` matrix the Lanczos–Chebyshev algorithm is more than twice as fast as the SVD method and uses fewer flops. It is also somewhat faster than Lui's algorithm. For the tridiagonal `condif` matrix the Lanczos–Chebyshev algorithm gives a speedup of 52.4 over the SVD algorithm and is an order of magnitude faster than Lui's algorithm, with similar improvements in flop counts.

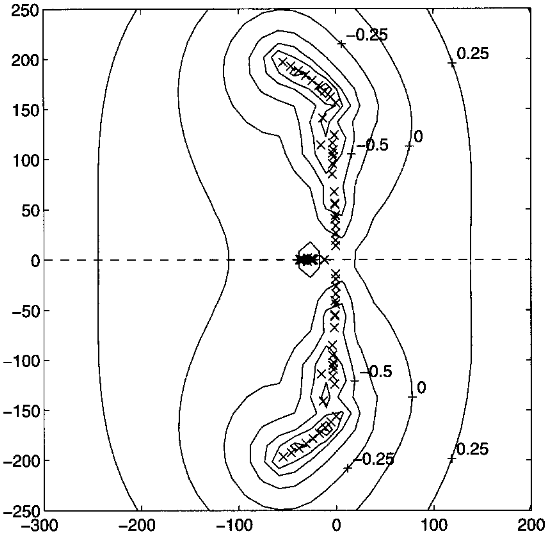

Figure 5.4: Pseudospectra of `tolosa(135)`: SVD algorithm (upper half) and Lanczos–Chebyshev algorithm (lower half).

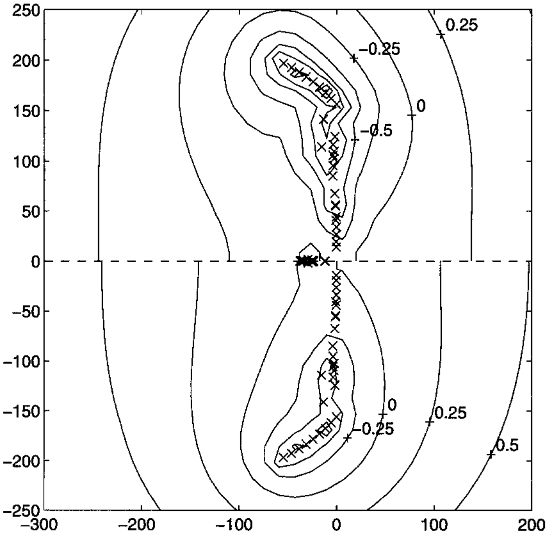

Figure 5.5: Pseudospectra of `tolosa(135)`: SVD algorithm (upper half) and Lui's algorithm (lower half).

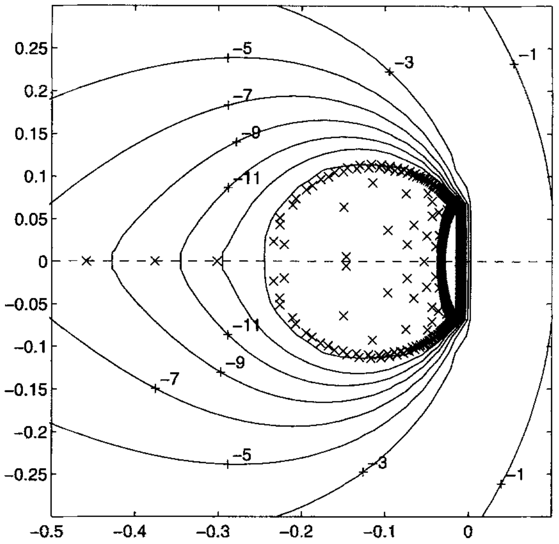

Figure 5.6: Pseudospectra of **condif(400)**: SVD algorithm (upper half) and Lanczos–Chebyshev algorithm (lower half).

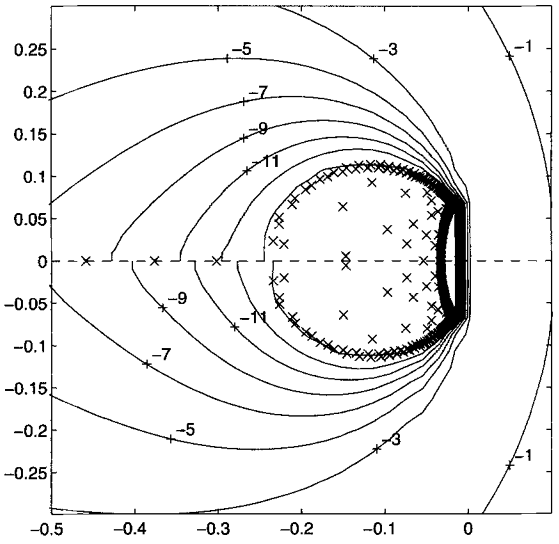

Figure 5.7: Pseudospectra of **condif(400)**: SVD algorithm (upper half) and Lui's algorithm (lower half).

Table 5.3: Times T (seconds) and flop counts F for the computation of pseudospectra.

Matrix	Algorithm	T	T_{SVD}/T	F	F_{SVD}/F
`tolosa(135)`	SVD	3498	1	1.37×10^{10}	1
	Lui's algorithm	1950	1.8	8.81×10^{9}	1.6
	Lanczos–Chebyshev	1541	2.3	2.02×10^{9}	6.8
`condif(400)`	SVD	90457	1	3.53×10^{11}	1
	Lui's algorithm	26260	3.5	7.20×10^{10}	4.9
	Lanczos–Chebyshev	1727	52.4	1.45×10^{9}	243.4

Table 5.4: Experiments on the Lanczos strategy and on the restarting method; times T (seconds) and flop counts F.

Algorithm	T	T_{LC}/T	F	F_{LC}/F	No. of restarts
Lanczos–Chebyshev	1541	1	2.02×10^{9}	1	314
LC1	450	3.4	7.92×10^{8}	2.6	2
LR	3221	0.48	3.53×10^{10}	0.06	3985

To show the effect of our chosen convergence and restart strategies we have compared Algorithm 3.1 with the following two variants on a pseudospectra computation for the `tolosa(135)` matrix, with the same grid as in the first example.

1. A Lanczos–Chebyshev algorithm (LC1) where the Ritz pairs are computed and convergence is tested on every step (as is done in [19, 20, 21]) rather than every m steps.

2. A Lanczos algorithm (LR) restarted with the computed Ritz vector.

Table 5.4 summarizes the results.

Restarting the Lanczos algorithm using Chebyshev acceleration is twice as fast as using the computed Ritz vector. Using the strategy LC1 rather than our algorithm leads to a speedup greater than 3. However, the computed pseudospectra, shown in Figure 5.8, are incorrect. The reason is that the starting vector of the Lanczos process is frequently (exactly) orthogonal to the desired eigenvector, and the LC1 strategy detects convergence prematurely to an eigenpair other than the desired maximal eigenpair (similar observations are made in [20]).

6 Conclusions.

When computing the field of values and pseudospectra of large matrices, significant savings in computational effort can be achieved over the standard methods based on application of the QR algorithm and the Golub–Reinsch SVD algorithm to matrices defined on a grid in the complex plane. Using Lanczos or

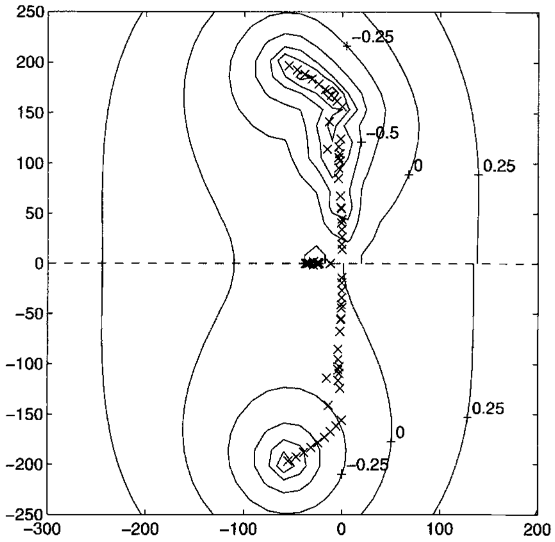

Figure 5.8: Pseudospectra of `tolosa(135)`: SVD algorithm (upper half) and `LC1` algorithm (lower half).

power-type eigenvalue methods continuation can be employed by taking as the starting vector the solution vector from an adjacent grid point, and these same methods enable sparsity to be exploited.

We have built on previous work to derive an improved Lanczos-based algorithm for computing pseudospectra. The key features of our algorithm, not all of which are present in previous algorithms, are a backward error stopping criterion, with a tolerance of the order of the unit roundoff; Chebyshev acceleration; selective reorthogonalization; and a shift and invert technique. By adapting the techniques used for computing pseudospectra we have also derived a new, efficient algorithm for computing the field of values.

Acknowledgements.

We thank Des Higham and Nick Trefethen for valuable comments on a draft manuscript, and the referees for suggestions that led to an improved paper.

REFERENCES

1. Zhaojun Bai, *A collection of test matrices for large scale nonsymmetric eigenvalue problems (version 1.0)*, Manuscript, July 1994.
2. Maria Bennani and Thierry Braconnier, *Stopping criteria for eigensolvers*, Technical Report TR/PA/94/22, CERFACS, Toulouse, France, 1994.
3. Thierry Braconnier, *Influence of orthogonality on the backward error and the stopping criterion for Krylov methods*, Numerical Analysis Report No. 281, Manchester

Centre for Computational Mathematics, Manchester, England, December 1995.

4. Thierry Braconnier, Françoise Chatelin, and J. C. Dunyach. *Highly nonnormal eigenproblems in the aeronautical industry,* Japan J. Indust. Appl. Math., 12 (1995), pp. 123–136.
5. Martin Brühl, *A curve tracing algorithm for computing the pseudospectrum,* BIT, 36 (1996), pp. 441–454.
6. J. Carpraux, J. Erhel, and M. Sadkane, *Spectral portrait for non Hermitian large sparse matrices,* Technical Report 777, INRIA/IRISA, Unité de Recherche, Rennes, France, 1993.
7. Françoise Chaitin-Chatelin and Valérie Frayssé, *Lectures on Finite Precision Computations,* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1996.
8. Françoise Chatelin, *Eigenvalues of Matrices,* Wiley, Chichester, UK, 1993.
9. Iain S. Duff, Roger G. Grimes, and John G. Lewis, *Users' guide for the Harwell–Boeing sparse matrix collection (release 1),* Report RAL-92-086, Atlas Centre, Rutherford Appleton Laboratory, Didcot, Oxon, UK, December 1992.
10. Gene H. Golub and Charles F. Van Loan, *Matrix Computations,* 2nd edition, Johns Hopkins University Press, Baltimore, MD, USA, 1989.
11. Chunyang He and G. A. Watson, *An algorithm for computing the numerical radius,* Manuscript, 1995.
12. Desmond J. Higham and Lloyd N. Trefethen, *Stiffness of ODEs,* BIT, 33 (1993), pp. 285–303.
13. Nicholas J. Higham, *The Test Matrix Toolbox for* MATLAB *(version 3.0),* Numerical Analysis Report No. 276, Manchester Centre for Computational Mathematics, Manchester, England, September 1995.
14. Roger A. Horn and Charles R. Johnson, *Matrix Analysis,* Cambridge University Press, 1985.
15. Roger A. Horn and Charles R. Johnson, *Topics in Matrix Analysis,* Cambridge University Press, 1991.
16. Charles R. Johnson, *Numerical determination of the field of values of a general complex matrix,* SIAM J. Numer. Anal., 15:3 (1978), pp. 595–602.
17. J. Kuczyński and H. Woźniakowski, *Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start,* SIAM J. Matrix Anal. Appl., 13:4 (1992), pp. 1094–1122.
18. R. B. Lehoucq, D. C. Sorensen, and P. Vu, *ARPACK: An implementation of the implicitly re-started Arnoldi iteration that computes some of the eigenvalues and eigenvectors of a large sparse matrix,* 1995. Available from `netlib@ornl.gov` under the directory `scalapack`.
19. S. H. Lui, *Computation of pseudospectra by continuation,* SIAM J. Sci. Comput., 1996. To appear.
20. Osni A. Marques and Vincent Toumazou, *Spectral portrait computation by a Lanczos method (normal equation version),* Technical Report TR/PA/95/02, CERFACS, Toulouse, France, December 1994.
21. Osni A. Marques and Vincent Toumazou, *Spectral portrait computation by a Lanczos method (augmented matrix version),* Technical Report TR/PA/95/05, CERFACS, Toulouse, France, February 1995.

22. C. C. Paige, *Krylov subspace processes, Krylov subspace methods, and iteration polynomials,* in Proceedings of the Cornelius Lanczos International Centenary Conference, J. David Brown, Moody T. Chu, Donald C. Ellison, and Robert J. Plemmons, eds., Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1994, pp. 83–92.

23. B. N. Parlett, H. Simon, and L. M. Stringer, *On estimating the largest eigenvalue with the Lanczos algorithm*, Math. Comp., 38 (1982), pp. 153–165.

24. Beresford N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1980.

25. Beresford N. Parlett, *Misconvergence in the Lanczos algorithm*, in Reliable Numerical Computation, M. G. Cox and S. J. Hammarling, eds., Oxford University Press, 1990, pp. 7–24.

26. Satish C. Reddy and Lloyd N. Trefethen, *Pseudospectra of the convection-diffusion operator*, SIAM J. Appl. Math., 54:6 (1994), pp. 1634–1649.

27. J. L. Rigal and J. Gaches, *On the compatibility of a given solution with the data of a linear system*, J. Assoc. Comput. Mach., 14:3 (1967), pp. 543–548.

28. Axel Ruhe, *Closest normal matrix finally found!*, BIT, 27 (1987), pp. 585–598.

29. Axel Ruhe, *The rational Krylov algorithm for large nonsymmetric eigenvalues—Mapping the resolvent norms (pseudospectrum)*, Manuscript, July 1994.

30. Youcef Saad, *Chebyshev acceleration techniques for solving nonsymmetric eigenvalue problems*, Math. Comp., 42:166 (1984), pp. 567–588.

31. Torsten Ström, *On logarithmic norms*, SIAM J. Numer. Anal., 12:5 (1975), pp. 741–753.

32. Kim-Chuan Toh and Lloyd N. Trefethen, *Calculation of pseudospectra by the Arnoldi iteration*, SIAM J. Sci. Comput., 17:1 (1996), pp. 1–15.

33. Lloyd N. Trefethen, *Pseudospectra of matrices*, in Numerical Analysis 1991, Proceedings of the 14th Dundee Conference, D. F. Griffiths and G. A. Watson, eds., volume 260 of *Pitman Research Notes in Mathematics*, Longman Scientific and Technical, Essex, UK, 1992, pp. 234–266.

34. Lloyd N. Trefethen, *Spectra and Pseudospectra: The Behavior of Non-Normal Matrices and Operators*, Book in preparation.

35. Lloyd N. Trefethen, Anne E. Trefethen, Satish C. Reddy, and Tobin A. Driscoll, *Hydrodynamic stability without eigenvalues*, Science, 261 (1993), pp. 578–584.

36. A. van der Sluis and H. A. van der Vorst, *The convergence behaviour of Ritz values in the presence of close eigenvalues*, Linear Algebra Appl., 88/89 (1987), pp. 651–694.

37. G. A. Watson, *Computing the numerical radius*, Linear Algebra Appl., 234 (1996), pp. 163–172.