Acta Numerica (1999), pp. 001–

Computation of Pseudospectra

Lloyd N. Trefethen
Oxford University Computing Laboratory
Wolfson Bldg., Parks Road
Oxford OX1 3QD, UK
LNT@comlab.ox.ac.uk

There is more to the computation of pseudospectra than the obvious algorithm of computing singular value decompositions on a grid and sending the results to a contour plotter. Other methods may be hundreds of times faster. The state of the art is reviewed, with emphasis on methods for dense matrices, and a MATLAB code is given.

CONTENTS

1. Introduction

A new tool has become popular in the 1990s for the study of matrices and linear operators. The traditional tool is eigenvalues or spectra (for matrices or linear operators, respectively), which may reveal information about the behaviour of systems both linear and nonlinear, including stability, resonance, and accessibility to matrix iterations and preconditioners. Eigenvalues and spectra tend to be less informative, however, when the matrix or operator is non-Hermitian, or more generally, nonnormal (roughly, having nonorthogonal eigenvectors). Pseudospectra are sets in the complex plane that sometimes do better. For each $\epsilon > 0$, the ϵ-pseudospectrum of a given matrix or operator is a nonempty set in the complex plane, and the spectrum and the field of values (= numerical range) can be recovered as special cases from the limits $\epsilon \to 0$ and (after peeling away an ϵ-border region) $\epsilon \to \infty$, respectively.

Pseudospectra seem to have been invented independently (with different names) at least five times: by Landau (1975, 1976, 1977), Varah (1979), Godunov, *et al.* (Godunov, 1992 and 1997; Godunov, Kiriljuk, and Kostin, 1990; Godunov, Antonov, Kiriljuk, and Kostin, 1993; Kostin, 1991), myself (1990, 1992), and Hinrichsen, Pritchard and Kelb (1992, 1993). Aside from one plot by J. Demmel (1987), however, they seem not to have been computed numerically before 1990. This situation changed completely in the 1990s, and pseudospectra have now been computed for dozens of applications. Here is a list of some of them, ordered by year of publication.

- spectral methods for differential equations (Reddy and Trefethen, 1990)
- approximate Fourier analysis (Donato, 1991)
- matrix iterations (Nachtigal, Reichel and Trefethen, 1992)
- Toeplitz matrices and operators (Reichel and Trefethen, 1992)
- control theory (Hinrichsen and Pritchard, 1992)
- random matrices (Trefethen, 1992)
- Orr–Sommerfeld operator (Reddy, Schmid and Henningson, 1993)
- Airy operator (Reddy, Schmid and Henningson, 1993)
- flow in a channel (Trefethen, Trefethen, Reddy and Driscoll, 1993)
- compressible boundary layer flow (Schmid, *et al.*, 1993)
- trailing line vortex flow (Schmid, *et al.*, 1993)
- Wiener–Hopf operators (Reddy, 1993)
- stiffness of ordinary differential equations (Higham and Trefethen, 1993)
- convection-diffusion operators (Reddy and Trefethen, 1994)
- Hille–Phillips and Zabczyk operators (Baggett, 1994)
- polynomial zerofinding (Toh and Trefethen, 1994)
- magnetohydrodynamics (Borba, *et al.*, 1994)
- aerodynamic flutter (Braconnier, Chatelin, and Dunyach, 1995)
- flow down inclined plane (Olsson and Henningson, 1995)
- rounding error analysis (Chaitin-Chatelin and Frayssé, 1996)
- reaction-convection-diffusion equations (Higham and Owren, 1996)
- preconditioners for fluid mechanics (Darmofal and Schmid, 1996)
- absorbing boundary conditions (Driscoll and Trefethen, 1996)
- waveform relaxation (Lumsdaine and Wu, 1997)
- Papkovitch–Fadle problem (Trefethen, 1997)
- Abel integral operators (Plato, 1997)
- Ginzburg-Landau equations (Cossu and Chomaz, 1997)
- non-Hermitian quantum mechanics (Davies, 1998)
- differential operators (Davies, 1998b)
- Markov chain ‘cutoff phenomenon’ (Jónsson and Trefethen, 1998)
- Chebyshev polynomials of matrices (Toh and Trefethen, 1998)
- flow in a pipe (Trefethen, Trefethen and Schmid, 1999)
- ionospheric instabilities (Flaherty, Seyler and Trefethen, 1999)
- lasers and optical resonators (see Section 21)

As is common in the history of scientific computing, this progress has been made possible by developments in both hardware and algorithms.

There is an obvious numerical method for plotting pseudospectra: compute an SVD (singular value decomposition) at each point on a grid in the complex plane, then send the results to a contour plotter. However, one can do better, typically by a factor of about $N/4$ for a problem of dimension N, even without using multiple processors or the techniques of sparse matrices. The aim of this article is to explain the ideas that make this possible.

The style of the article is tutorial. The reader I imagine has an interest in eigenvalue problems for large matrices, probably arising as discretizations of differential or integral operators, and a suspicion that sometimes they do not reveal all they should about his or her problem. Among the questions he or she may ask are, When should I compute pseudospectra? How should I do it? What will they tell me?

Throughout our discussion, ideas for matrices will be formulated in a manner consistent with the fact that in most applications, the matrices we are dealing with are approximations to infinite-dimensional operators. Since pseudospectra are norm-dependent, it is essential to frame the matrix norms in a manner that permits them to converge to the appropriate continuous norms as the approximation is refined. We handle this by defining the inner product and norm associated with a matrix A with respect to a weighting matrix W, which might, for example, be a diagonal matrix of Gauss quadrature coefficients. The similarity transformation $B = WAW^{-1}$ then provides a matrix B for which the equivalent problem of pseudospectra is associated with the usual Euclidean inner product and norm.

The literature on the numerical computation of pseudospectra is growing, but still manageable, and in this article, all the papers I know of are cited. Let me acknowledge here at the beginning those authors I am aware of who have published on this subject: C. Bekas, Thierry Braconnier, Martin Brühl, Jean-François Carpraux, Françoise Chaitin-Chatelin, Jocelyne Erhel, Valérie Frayssé, Eduardo Gallestey, Stratis Gallopoulos, Luc Giraud, Sergei Godunov, Vincent Heuveline, Nicholas Higham, Didi Hinrichsen, Viktor Kostin, Shiu-Hong Lui, P. Lavallée, Osni Marques, Alan McCoy, Bernard Philippe, Tony Pritchard, Axel Ruhe, Milhoud Sadkane, Valeria Simoncini, Kim-Chuan Toh, Vincent Toumazou, and Anne Trefethen. To any others whom I may have overlooked, my sincere apologies.

For an introduction to the noncomputational aspects of pseudospectra, I recommend (Trefethen, 1992), (Trefethen, 1997), (Trefethen, 1999), and (Trefethen, Trefethen, Reddy and Driscoll, 1993).

In 1990, getting a good plot of pseudospectra on a workstation for a 30 × 30 matrix took me several minutes. Today I would expect the same of a 300 × 300 matrix, and pseudospectra of matrices with dimensions in the thousands are around the corner.

2. Norms and adjoints: matrices A and B

Let A be a real or complex matrix or closed linear operator acting in a Hilbert space over the complex numbers $\mathbb{C}$ with inner product $(\cdot\,,\,\cdot)$ and corresponding norm $\|\cdot\|$. (The generalization to Banach spaces is discussed in Section 18.) In practice we are so often concerned with matrix discretizations of infinite-dimensional linear operators that it is important to be more explicit. The following manipulations in the context of pseudospectra were perhaps first written down in §5 and Appendix A of (Reddy, Schmid and Henningson, 1993). In the matrix case, we assume that a nonsingular *weight matrix* W has been prescribed and that $(\cdot\,,\,\cdot)$ and $\|\cdot\|$ are defined by

$$(u,v) = (Wu)^H(Wv) = u^H(W^HW)v, \tag{2.1}$$

$$\|u\|^2 = (u,u) = (Wu)^H(Wu) = u^H(W^HW)u. \tag{2.2}$$

Here and throughout this paper in similar contexts, u and v are column vectors and u^H, the *Hermitian conjugate,* is the complex conjugate transpose of u, and similarly for W. Another way to write (2.1) and (2.2) is

$$(u,v) = (Wu,Wv)_2, \qquad \|u\| = \|Wu\|_2, \tag{2.3}$$

where $(u,v)_2 = u^Hv$ and $\|u\|_2^2 = u^Hu$—'the 2-norm'. In applications, W might be $\sqrt{h}$ times the identity, if A is obtained by discretization on a regular 1D grid of spacing h, or it might be a nonconstant diagonal matrix of quadrature weights for discretizations on irregular grids.

The *adjoint* of A, denoted A^*, is defined by the condition $(Au,v) = (u,A^*v)$ for all u and v in the domains of A and A^*, respectively. In the matrix case, a little calculation shows that A^* is given by

$$A^* = (W^HW)^{-1}A^H(W^HW). \tag{2.4}$$

If $W = I$, all the complications above vanish and we have $(u,v) = (u,v)_2$, $\|u\| = \|u\|_2$, and $A^* = A^H$. Alternatively, for general W, we can make the complications go away by introducing the new matrix

$$B = WAW^{-1}. \tag{2.5}$$

If $v = Au$ for some u and v, then (2.5) implies

$$(Wv) = B(Wu),$$

and by the definition of matrix norms subordinate to vector norms, this implies $\|A\| = \|B\|_2$. More generally, we have $\|f(A)\| = \|f(B)\|_2$ for any function f. From (2.4) we can also compute

$$B^H = WA^*W^{-1}, \tag{2.6}$$

revealing that the same transformation that takes A to B also takes A^* to B^H.

The matrix A is *normal* if $AA^* = A^*A$, or equivalently, if A has a complete set of eigenvectors that are orthogonal with respect to the inner product $(\cdot\,,\,\cdot)$. From (2.4) we can calculate that this is the same as the equality $BB^H = B^HB$, or equivalently, the condition that B has a complete set of eigenvectors that are orthogonal with respect to the inner product $(\cdot\,,\,\cdot)_2$. For example, A is normal with respect to $(\cdot\,,\,\cdot)$ if it is self-adjoint or skew-adjoint, and B is normal with respect to $(\cdot\,,\,\cdot)_2$ if it is Hermitian or skew-Hermitian. Sometimes we will say that a matrix is 'highly nonnormal' or 'far from normal', terms with no precise meaning beyond the idea that its eigenvectors, if they exist, are in some sense 'far from orthogonal'.

3. Spectrum and pseudospectra

There is a function $f(A)$ that we care about especially: the *resolvent.* For any $z \in \mathbb{C}$, the resolvent of A at z is the matrix or linear operator

$$(z - A)^{-1},$$

if this exists and is bounded, where $z - A$ is a shorthand for $zI - A$ and I is the identity. The *spectrum* of A, denoted by $\Lambda(A)$, is the set of $z \in \mathbb{C}$ where the resolvent does not exist or is unbounded.

The norm of the resolvent is

$$\|(z - A)^{-1}\| = \|(z - B)^{-1}\|_2,$$

with B related to A as always by (2.5), and we use the convention that this quantity is defined for all $z \in \mathbb{C}$, including points in the spectrum $\Lambda(A) = \Lambda(B)$, where it takes the value ∞. For each $\epsilon > 0$, the ϵ-pseudospectrum of A is defined by

$$\begin{aligned} \Lambda_\epsilon(A) &= \{\, z \in \mathbb{C} : \ \|(z - A)^{-1}\| \geq \epsilon^{-1} \,\} \\ &= \{\, z \in \mathbb{C} : \ \|(z - B)^{-1}\|_2 \geq \epsilon^{-1} \,\}. \end{aligned} \tag{3.1}$$

In words, the ϵ-pseudospectrum is the subset of the complex plane bounded by the ϵ^{-1} level curve or curves of the resolvent norm. (Some authors use a strict inequality; it makes little difference for applications.) For $z \notin \Lambda(A)$, since $\|(z-A)^{-1}\|$ is the supremum over all unit vectors u and v of the subharmonic function $|(u, (z-A)^{-1}v)|$, it is a subharmonic function itself and hence satisfies the maximum principle, which implies that each bounded component of any ϵ-pseudospectrum contains part of $\Lambda(A)$. The subharmonicity of the norm of the resolvent was pointed out by Boyd and Desoer (1985) and has been exploited for computational purposes by Gallestey (1998,1998b).

Other conditions can be derived that are equivalent to (3.1). Here is the one that is the most important and most different:

$$\Lambda_\epsilon(A) \ = \ \{\, z \in \mathbb{C} : \ z \in \Lambda(A + E) \text{ for some } E \text{ with } \|E\| \leq \epsilon \,\} \tag{3.2}$$

$$= \{ z \in \mathbb{C} : \ z \in \Lambda(B+E) \text{ for some } E \text{ with } \|E\|_2 \le \epsilon \}.$$

In words, the ϵ-pseudospectrum is the set of all complex numbers that are in the spectrum of some matrix or operator obtained by a perturbation of norm at most ϵ. This definition implies that pseudospectra can be interpreted in terms of perturbations of spectra, but this does not mean that the analysis of perturbations is the main thing pseudospectra are useful for. On the contrary, other aspects of behaviour of a matrix or linear operator tend to be more important in applications, including growth or decay of $\|A^n\|$ as a function of n and growth or decay of $\|e^{tA}\|$ as a function of t. I admit that over the years I have become exasperated by hearing so many people make the mistake of assuming that pseudospectra, since they can be defined by perturbations, must be a tool for coping robustly with rounding errors. In most applications, rounding errors are not the point at all.

A starting point for computations is a third equivalent definition of pseudospectra. If $\sigma_{\min}(A)$ denotes the smallest singular value of A, then we have

$$\Lambda_\epsilon(A) \ = \ \{ z \in \mathbb{C} : \ \sigma_{\min}(z-B) \le \epsilon \}. \qquad (3.3)$$

Thus the pseudospectra of A are the sets in the z-plane bounded by level curves of the function $\sigma_{\min}(z-B)$.

For details of the equivalence of (3.1)–(3.3), see for example van Dorsselaer, Kraaijevanger, and Spijker (1993). The mathematical foundations of such material are set forth in the book by Kato (1976).

4. A tutorial example

To make the discussion concrete, this article is built around a single example of a highly nonnormal differential operator, which we shall treat computationally by a succession of methods. The operator is a time-reduced one-dimensional Schrödinger operator of a standard kind, except that the potential function that defines the operator is complex rather than real:

$$Au(x) \ = \ u'' + (cx^2 - dx^4)u, \qquad c = 3+3\mathrm{i}, \ \ d = \tfrac{1}{16}. \qquad (4.1)$$

(The constants have been chosen to make the behaviour interesting.) This operator acts on functions defined on the whole real line $\mathbb{R}$. To be precise, the Hilbert space in which A acts is $L^2 = L^2(-\infty,\infty)$, and the domain on which it is defined is the subset of L^2 of functions that have a second derivative in L^2. Roughly speaking, for small x, the potential defining A looks quadratic and complex, whereas for large x it is quartic and nearly real. Note that A is invariant with respect to negation of x, which implies that if $u(x)$ is an eigenfunction of A with eigenvalue λ, then so is $u(-x)$. In fact, it can be proved that all the eigenvalues of A are simple, and thus each eigenfunction is either even or odd.

The observation that complex Schrödinger operators are highly nonnormal and have interesting pseudospectra is due to Brian Davies of King's College London (Davies, 1998). Our example is adapted from Davies' work.

Here then is our task. We are presented with an operator such as (4.1) and wish to find out: What do its spectrum and pseudospectra look like? What do they tell us about its behaviour?

5. Discretization

If an operator cannot be handled analytically, the usual course is to approximate it by finite matrices. For computations of ϵ-pseudospectra, we are typically interested in small values of ϵ, and thus high-accuracy approximations are desirable. This means that wherever possible, one should discretize by spectral methods rather than finite differences or finite elements, since spectral methods have arbitrarily high order of accuracy for smooth problems (Canuto, *et al.*, 1988; Fornberg and Sloan, 1994; Fornberg, 1996).

For our tutorial example, (4.1) has been discretized by a Chebyshev collocation spectral method on a finite interval $[-L, L]$ with boundary conditions $u(\pm L) = 0$. (One could work on $[0, L]$ and separate the even and odd parts of the problem, which are orthogonal, but we did not do this.) For clarity, especially in the treatment of weight functions, let us spell this out. First the interval $[-L, L]$ is approximated by the set of $N + 2$ Chebyshev points defined by

$$x_j = L\cos\left(\frac{j\pi}{N+1}\right), \qquad j = 0, \ldots, N+1. \tag{5.1}$$

The operator A is then approximated on this grid by an $N \times N$ matrix A_N defined by the following prescription. For any N-vector v, $A_N v$ is the N-vector obtained by two steps:

- Let p be the unique polynomial of degree $\le N+1$ with $p(\pm L) = 0$ and $p(x_j) = v_j$ for $1 \le j \le N$;
- For $j = 1, \ldots, N$, $(A_N v)_j = p''(x_j) + (cx_j^2 - dx_j^4)p(x_j)$.

The eigenfunctions of A decay exponentially, and as a consequence, we find that any particular eigenfunction can be computed accurately via discrete matrices A_N for some finite L. For the portion of the spectrum and pseudospectra considered in this article, $L = 10$ is sufficiently large, and from now on, all of our numerical examples are based on $L = 10$.

In the context of our spectral discretization, any N-vector v is associated with the continuous function $u(x)$ equal to the polynomial interpolant $p(x)$ described above for $|x| \le L$ and to zero for $|x| \ge L$. In particular, for each eigenvector v of A_N, there is an associated continuous function $u(x)$, and if v is sufficiently smooth and decays strongly enough to zero near $x = \pm L$, we expect that $u(x)$ will be close to an eigenfunction of A with approximately the same eigenvalue.

The description above is all we need for differentiation, but to compute pseudospectra, we need to integrate, too. For the spectral discretization our weight matrix W will take the form

$$W = \operatorname{diag}(w_1, w_2, \dots, w_N) \tag{5.2}$$

for suitable weights $w_j > 0$. A sufficient condition for A_N to converge in some sense to A as $N \to \infty$ is

$$N w_j^2 \to \frac{\pi}{2}\sqrt{L^2 - x^2} \tag{5.3}$$

for any sequence of indices j with $x_j \to x$, and any reasonable choice satisfying this condition (and indeed many choices that do not satisfy it) will generally produce good plots of pseudospectra. However, much better performance than mere convergence is achievable if we choose the weights based on ideas of Gauss quadrature, and in applications it is important to get this right if one is to be confident of the results. For our Chebyshev grid, there exists a set of *Gauss* (or *Gauss–Chebyshev–Lobatto*) *weights* $\{w_j\}$ satisfying (5.3) such that

$$\int_{-L}^{L} f(x)\,\mathrm{d}x \;=\; \sum_{j=1}^{N} w_j^2 f(x_j)$$

if f is any function equal to $\sqrt{L^2 - x^2}$ times a polynomial of degree $\le 2N-1$. These weights are simply

$$w_j^2 = \frac{\pi\sqrt{L^2 - x_j^2}}{2(N+1)},$$

and from now on, these are our choice, with W defined accordingly by (5.2) and B by (2.5).

Here is the MATLAB code segment that I used to construct the matrices A and B.

```
D = zeros(N+2,N+2); i = (0:N+1)'; ci = [2;ones(N,1);2];
x = cos(pi*i/(N+1));
for j = 0:N+1
  cj = 1; if j==0 | j==N+1 cj = 2; end
  denom = cj*(x(i+1)-x(j+1)); denom(j+1) = 1;
  D(i+1,j+1) = ci.*(-1).^(i+j)./denom;
  if j>0 & j<N+1 D(j+1,j+1) = -.5*x(j+1)/(1-x(j+1)^2); end;
  end;
D(1,1) = (2*(N+1)^2+1)/6; D(N+2,N+2) = -(2*(N+1)^2+1)/6;
L = 10;
```

```
x = x(2:N+1); x = L*x; D = D/L;
A = D^2; A = A(2:N+1,2:N+1);
A = A + (3+3*sqrt(-1))*diag(x.^2) - (1/16)*diag(x.^4);
w = sqrt(pi*sqrt(L^2-x.^2)/(2*(N+1)));
B = zeros(N,N); for j=1:N, B(:,j) = w.*A(:,j)/w(j); end
```

We now move on to the study of these matrix approximations to the differential operator (4.1). In doing so, however, we must note that for some problems, it may not be realistic to expect an operator to be approximated by a single matrix. An example arises in the large-scale hydrodynamic stability calculations of Trefethen, Trefethen, Reddy and Driscoll (1993). Here, the operator depends on two Fourier parameters α and β, and for each choice of the parameters, there is a different discretization matrix $A_{\alpha\beta}$. Computing the resolvent norm at a point z requires the minimization of $\sigma_{\min}(z-A_{\alpha\beta})$ over all α and β, and the optimal choices vary from one value of z to the next. Situations like this are not unusual in large-scale applications, and when they arise, it may be necessary to consider discretization and computation of pseudospectra in tandem rather than in sequence.

6. Eigenvalues and eigenvectors

The first thing we compute are eigenvalues of A or, equivalently, B. For matrices of dimensions less than 1000 or so, this is easily done by standard 'direct' methods related to the QR algorithm, which deliver results to close to machine precision (not counting what is lost to ill-conditioning) in $\mathcal{O}(N^3)$ floating point operations. For matrices of larger dimensions, Krylov subspace iterations are generally used instead to determine not all the eigenvalues but those in the portion of the complex plane considered important (Lehoucq, Sorensen, and Yang, 1998; Saad, 1992).

For our example problem, we can get away with dimensions small enough for direct methods to be appropriate, and Figure 1 shows eigenvalues calculated by standard methods for the spectral approximations A_N with $N =$ 140, 160, 180, 200. As is typical with discretizations of differential operators, it is the eigenvalues closest to the origin that are obtained for the smaller values of N, since these tend to correspond to smoother eigenmodes, resolvable on coarser grids. Once N is large enough that all the eigenvalues in this frame are essentially correct, we observe a Y-shaped distribution, with all the eigenvalues lying in the left half-plane and an infinite curve of them extending towards $-\infty$. Of course, when approaching a problem like this in practice, one must take pains at every step to vary all possible aspects of the discretization systematically until one is confident that the results are correct. In some cases convergence theorems will be available to give further reassurance (Chatelin, 1983).

Figure 1 is reminiscent of Figure 3 and other figures in the paper by

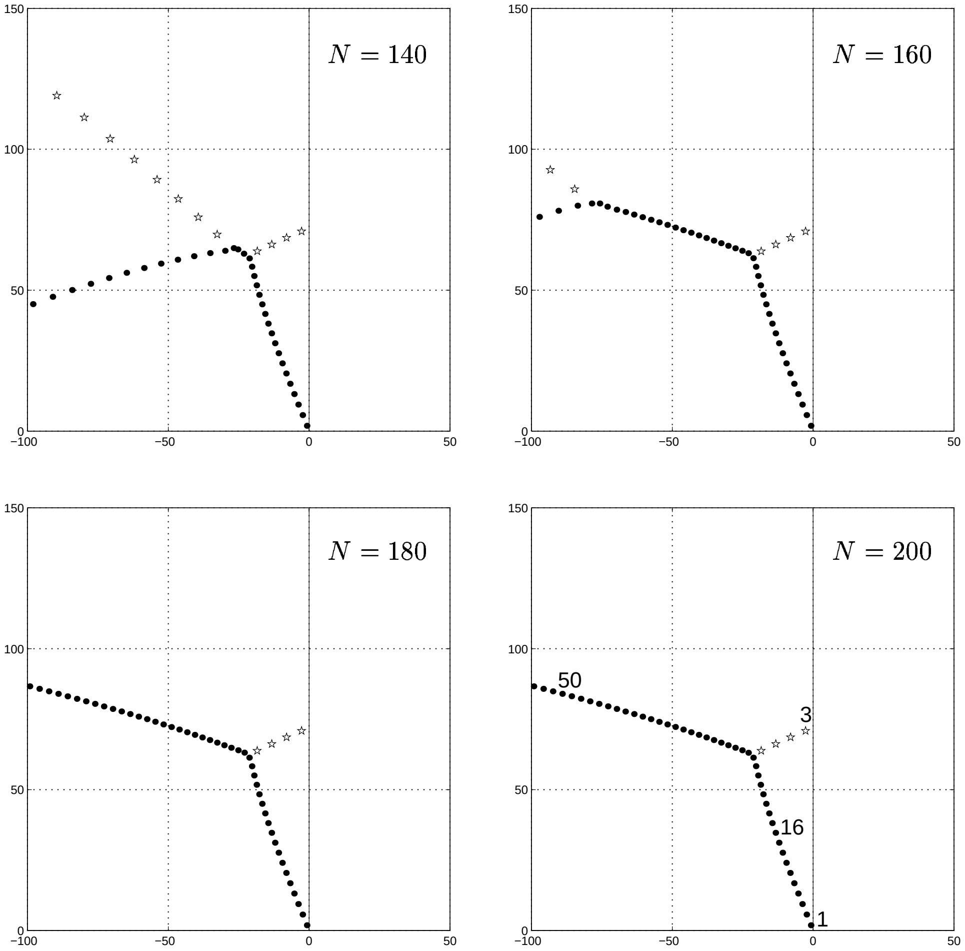

Fig. 1. Eigenvalues of matrix approximations A_N to (4.1) of dimensions $N = 140, 160, 180, 200$. Dots mark simple eigenvalues and stars mark nearly degenerate (though not exactly degenerate) pairs. As N increases, the two paths on the left zip together into a single line. For $N = 200$, the values throughout this part of the complex plane are accurate to 3 digits or more, and we take A_{200} as our matrix A for subsequent computations. (The labels 1, 3, 16, 50 are utilized in Figure 2.) Another 147 eigenvalues of A_{200} lie outside the axis limits to the left.

Reddy, Schmid and Henningson (1993) on pseudospectra of Orr–Sommerfeld operators. That paper represents an outstanding first example of a study in which pseudospectra of a differential operator were computed carefully.

Computation times will occasionally be reported in this article, all based on MATLAB programs executed on a SUN Ultra 30 workstation. To find the eigenvalues of A_{200} takes a little more than one second, whereas the smaller matrix A_{140} can be handled in less than half a second.

It is typical in applications to encounter a picture like Figure 1, which blends some degree of complexity with a great deal of structure. Naturally, one wants to know more, and a first question one may ask is, what do the eigenvectors look like? For this question, A and B are no longer identical: if v is an eigenvector of A, then Wv is the corresponding eigenvector of B. For a plot representing physical space, it is the former that we want, and Figure 2 shows four of the eigenvectors of A_{200}. The four nearly degenerate eigenvalues in the upper-right branch of the Y correspond to even/odd eigenfunction pairs. For example, 'mode 3', illustrated in the figure, is even, whereas mode 4 is odd, but the two eigenvalues differ by less than one part in 10^8. Modes 1 and 16 are representative of eigenvectors that 'live', loosely speaking, in the quadratic, complex part of the potential, where $|x|$ is small enough that the cx^2 term dominates the dx^4 term in (4.1), whereas mode 50 is one for which the dx^4 term is dominant.

7. Scalar measures of nonnormality

Suppose we suspect that A may be highly nonnormal. Before turning to pseudospectra, there are various scalars we might compute in an attempt to shed light on this matter. An early and influential paper on this topic was by Henrici (1962), and further contributions have been due to Chaitin-Chatelin and Frayssé (1996), among others.

For simplicity we use the B formulation of Section 2; all our statements have twins for A.

One scalar we might consider, which goes back essentially to Henrici, is

$$\frac{\|B^H B - BB^H\|_2}{\|B\|_2^2}. \tag{7.1}$$

For the matrix B_{200} this ratio has the value 0.01843, a number that seems fully converged for the limit $N \to \infty$ (for B_{300} we get 0.01843 again). If the denominator of (7.1) is changed to $\|B^2\|_2$, as suggested by Chaitin-Chatelin and Frayssé (1996), we get the same results to five digits, and if we further replace the Euclidean norm by the Frobenius norm $\|B\|_F^2 = \sum\sum |b_{ij}|^2$ to obtain what Chaitin-Chatelin and Frayssé call the *Henrici number*, the numbers change only modestly to 0.02602. These results suggest that in some global sense, B is close to normal. This is a reflection of the

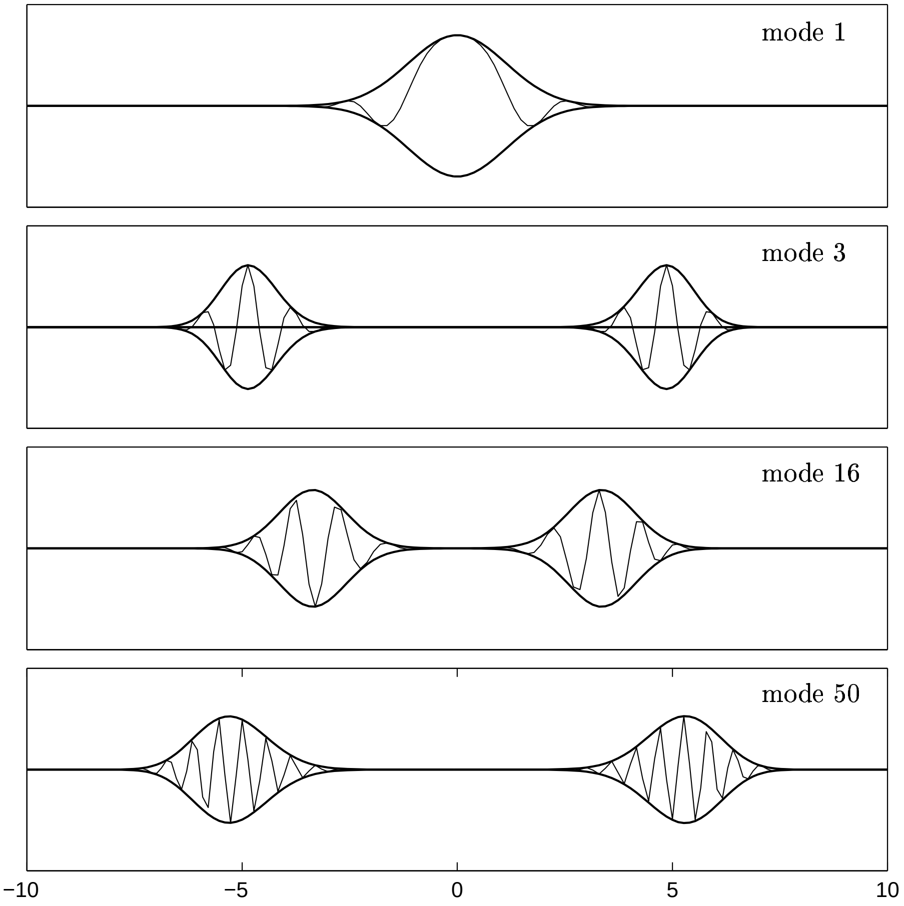

Fig. 2. Eigenvectors corresponding to the four eigenvalues of A_{200} labeled in Figure 1. (The mode numbers are sorted by decreasing real part.) The inner curves are the real parts (subject to change with complex scaling), and the outer envelopes are the absolute values and their negatives. These modes are actually more accurate than they look, for they have been plotted by straight line interpolation between points, whereas in fact the mathematical model is based on polynomial interpolants. Notice that mode 50, with about 2 points per wavelength, is near the limit of resolution for this grid.

fact that since the coefficient d in (4.1) is real, the nonnormality of this operator is localized for small $|x|$.

Another scalar we might consider, also going back to Henrici, is

$$\frac{\|T\|_2}{\|B\|_2}, \tag{7.2}$$

where T is the strictly upper-triangular part of a Schur triangularization (unitary triangularization) of B. Different Schur triangularizations may lead to different values of $\|T\|_2$, so (7.2) is it stands is not well defined, though it could be made so, at least for theoretical purposes, by taking the infimum over all Schur triangularizations. For B_{200}, with the triangularization computed arbitrarily by MATLAB, the ratio comes out as 0.02205, again effectively converged for $N \to \infty$ (for B_{300} it is 0.02204). Switching to the Frobenius norm, which makes the ratio independent of Schur triangularization, changes the result to 0.02166 (0.02164).

A third scalar we might consider is the distance of B to the set of normal matrices. For matrices measured in the Frobenius norm, this cannot be too far from the previous estimate, according to an inequality established by Lászlό (1994),

$$N^{-1/2}\|T\|_F \;\le\; \inf\{\, \|B - N\|_F \,:\, N \text{ is normal}\} \;\le\; \|T\|_F. \tag{7.3}$$

So far, the departure from normality of our operator appears modest, perhaps too small to be important. However, there is a further scalar that tells a different story. If $v_1, \dots, v_N$ are a set of linearly independent eigenvectors of B, each normalized by $\|v_j\|_2 = 1$ (the normalization is not necessary, just convenient), then an *eigenvector matrix* for B is an $N \times N$ matrix V whose columns are these vectors taken in any order. The *condition number* of V is the real number

$$\kappa_2(V) = \|V\|_2\, \|V^{-1}\|_2,$$

which is necessarily ≥ 1. The *Bauer–Fike Theorem* asserts that if B is perturbed by E, then the eigenvalues move by at most $\kappa_2(V)\|E\|_2$. If $\kappa_2(V) = 1$, then V must be unitary and B must be normal. If $\kappa_2(V) \gg 1$, on the other hand, perhaps there is a need to look beyond eigenvalues.

For our example B_{200} we find

$$\kappa_2(V) = 2.83 \times 10^{12}.$$

Evidently the matrix of eigenvectors of B is very ill-conditioned indeed. This number, unlike our previous ones, is not quite converged for $N \to \infty$; for B_{160} we get $\kappa_2(V) = 8.95 \times 10^{11}$ and for B_{240} we get $\kappa_2(V) = 3.74 \times 10^{12}$. However, convergence to a finite value as $N \to \infty$ does seem to take place; B_{300} gives $\kappa_2(V) = 3.79 \times 10^{12}$ and B_{340} gives $\kappa_2(V) = 3.75 \times 10^{12}$. In particular, the conclusion that $\kappa_2(V)$ is of order 10^{12} is genuine, and is not a symptom of machine precision on our computer.

Are some of the individual eigenvalues to blame for this pronounced nonnormality? To find out, we can look at the condition numbers of the eigenvalues, defined for a simple eigenvalue λ (Wilkinson, 1965, p. 68) by

$$\kappa(\lambda) = \frac{1}{|w^H v|}, \tag{7.4}$$

where w and v are normalized left and right eigenvectors of B corresponding to the eigenvalue λ, respectively. The significance of $\kappa(\lambda)$ is that a perturbation $B \to B + E$ may alter the eigenvalue λ by as much as $\kappa(\lambda)\|E\|_2$ (in the limit of infinitesimal perturbations), but not more. Each eigenvalue necessarily has $\kappa(\lambda) \leq \kappa_2(V)$. The condition numbers of some of the eigenvalues of our operator are indicated in Figure 3. Evidently the eigenvalues near the origin, in the bottom-right part of the Y, are well-conditioned. As one moves towards the fork of the Y, however, the condition numbers increase to about 10^{11}. (The behaviour in the line of nearly degnerate eigenvalue pairs is similar.) If we continue past the fork further into the left half-plane, $\kappa(\lambda)$ increases gradually to a maximum of about 3.6×10^{11} for $N = 200$, which becomes 4.3×10^{11} for $N = 240$.

It is apparent that the extreme ill-conditioning of the eigenvector matrix of B is reflected in the individual eigenvalues, but not just in one of them—in nearly all. Evidently there is a collective phenomenon at play here, a pattern that transcends individual eigenmodes, and indeed, this is perhaps as far as scalar measures of nonnormality can usefully take us.

8. Random perturbations

Having decided to move beyond scalars, we begin to think about pseudospectra. Now pseudospectra are sets in the complex plane, and the first question to ask is, what does it mean to compute them? Do we want a picture of boundaries of $\Lambda_\epsilon(A)$ for various values of ϵ? Do we want some kind of surface plot? Do we want an approximate functional representation of the function $\|(z - A)^{-1}\|$? The customary answer in the literature to date has been the first of these options, a graphical picture of boundary contours, and that is what we consider in this article, but it is possible that other variations will become popular in the future.

There is a simple idea for producing approximate pictures of pseudospectra: modify A by one or more small complex (even if A is real) random perturbations and look at the spectra of these perturbations. If A is a matrix, the idea of a random perturbation is well defined. Working as usual with the equivalent matrix B, we note that the set of all possible perturbations E of a given norm ϵ is compact and can be uniformly sampled by taking random $N \times N$ matrices D with entries $\sigma + \mathrm{i}\tau$, where σ and τ are independent

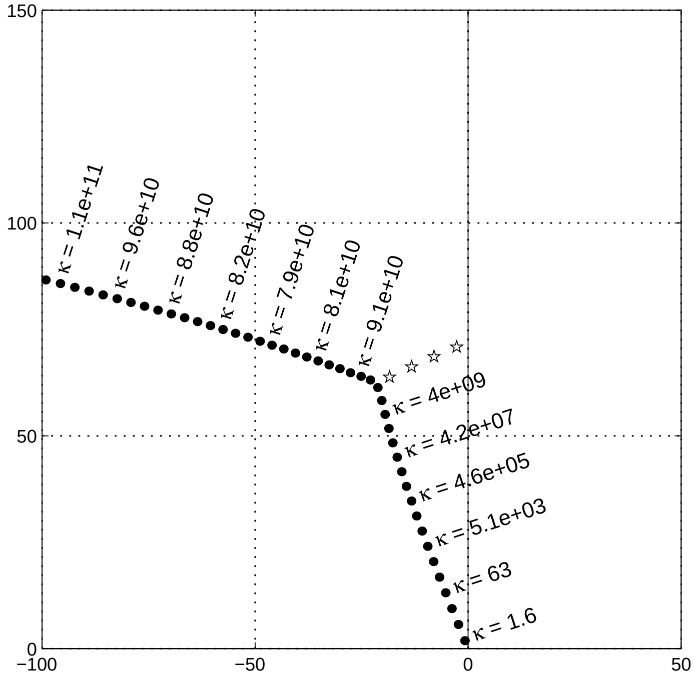

Fig. 3. Condition numbers $\kappa(\lambda)$ of some of the eigenvalues of the matrix $B = B_{200}$. The condition numbers of the four starred nearly degenerate eigenvalue pairs, from right to left, are 1.6×10^6, 6.7×10^7, 1.6×10^9, and 2.9×10^{10}.

standard normals, and then setting $E = \epsilon D/\|D\|_2$. Since $\|D\|_2 \sim \sqrt{2N}$ as $N \to \infty$, where N is the dimension, approximately the same effect can be achieved with less computation by the formula $E = \epsilon D/\sqrt{2N}$.

The approximation of pseudospectra by random perturbations was illustrated in (Trefethen, 1992), where for each of thirteen example matrices with $N = 32$, 100 random perturbations $A + E$ were considered and the 3200 resulting eigenvalues superimposed as small dots. Possibly the first computed examples of this kind appeared in (Trefethen, 1990).

In investigating random perturbations, there is no need to consider matrices E of full rank. As pointed out perhaps first by Riedel (Riedel, 1994), the boundary of the ϵ-pseudospectrum can equally well or better be traced by matrices of rank one, and so an alternative to the formulas above would be $D = xy^H$, where x and y are vectors of independent entries $\sigma + \mathrm{i}\tau$, followed by $E = \epsilon D/\|D\|_2 = \epsilon D/(\|x\|_2\,\|y\|_2)$. Proceeding in this way, we may trace the boundaries of the pseudospectra somewhat more efficiently, and there is no need for the computation of the norm of a matrix.

As a practical matter, random perturbations are a valuable tool that should be used routinely in dealing with highly nonnormal matrices. Random perturbations are more important than the scalar measures of nonnormality discussed in the last section, for they reveal more without being much more expensive to calculate. To illustrate how compelling this technique may be, Figure 4 shows the eigenvalues of one random rank one perturbation in each case of the form $B \to B + E$, where $B = B_{200}$ for our problem and $\|E\|_2 = \epsilon = 10^{-1}, 10^{-3}, 10^{-5}, 10^{-7}$. (The results look about the same for perturbations of full rank.) For the first time we begin to see the 'shape' of this matrix A. As we would expect on the basis of Figure 3, the degree of nonnormality is pronounced all along the tail of the Y extending into the left half-plane, and reasonably uniform along that path.

There are three problems with the technique of computing eigenvalues of random perturbations. One is that it gives only an approximate picture of the pseudospectra. Another is that pictures of this kind all too easily mislead people into presuming that the main point of analysis of pseudospectra is the investigation of perturbations. Finally, if A is an operator of infinite dimension, the notion of a random perturbation does not make sense, because E must range over a space that is not compact and thus cannot be sampled uniformly. In practice, therefore, I recommend that one begin by computing eigenvalues of random perturbations of finite matrices, without worrying too much what the precise definition of 'random' is in the case of a discretized operator, but then move on to other methods if the sensitivity to perturbations of the physically important eigenvalues is large.

A different aspect of random perturbations is that in some applications, a perturbation of a structured kind may reveal certain algebraic properties of a matrix or operator. An important special case is that if A is real, plotting the eigenvalues of real perturbations $A + E$ may reveal the Jordan structure of A; the 'spider plots' of Chaitin-Chatelin and Frayssé that show this effect are beautiful and fascinating, and one of them appears on the cover of their book (Chaitin-Chatelin and Frayssé, 1996). A systematic study of structured perturbations has been the subject of a number of papers by Hinrichsen and Pritchard and Kelb (1992, 1993, 1994).

9. Contour plots via the SVD

Let us now calculate pseudospectra properly. The place to begin is with the singular value decomposition, taking advantage of definition (3.3). The obvious algorithm is to evaluate $\sigma_{\min}(z - B)$ for values of z on a grid in the complex plane and then generate a contour plot from this data. (If B is Hermitian, the picture will be symmetric with respect to the real axis, and one halves the computation time by taking advantage of this symmetry.) Here for example is a Matlab code fragment for this kind of computation, assuming ν points in each direction on the grid:

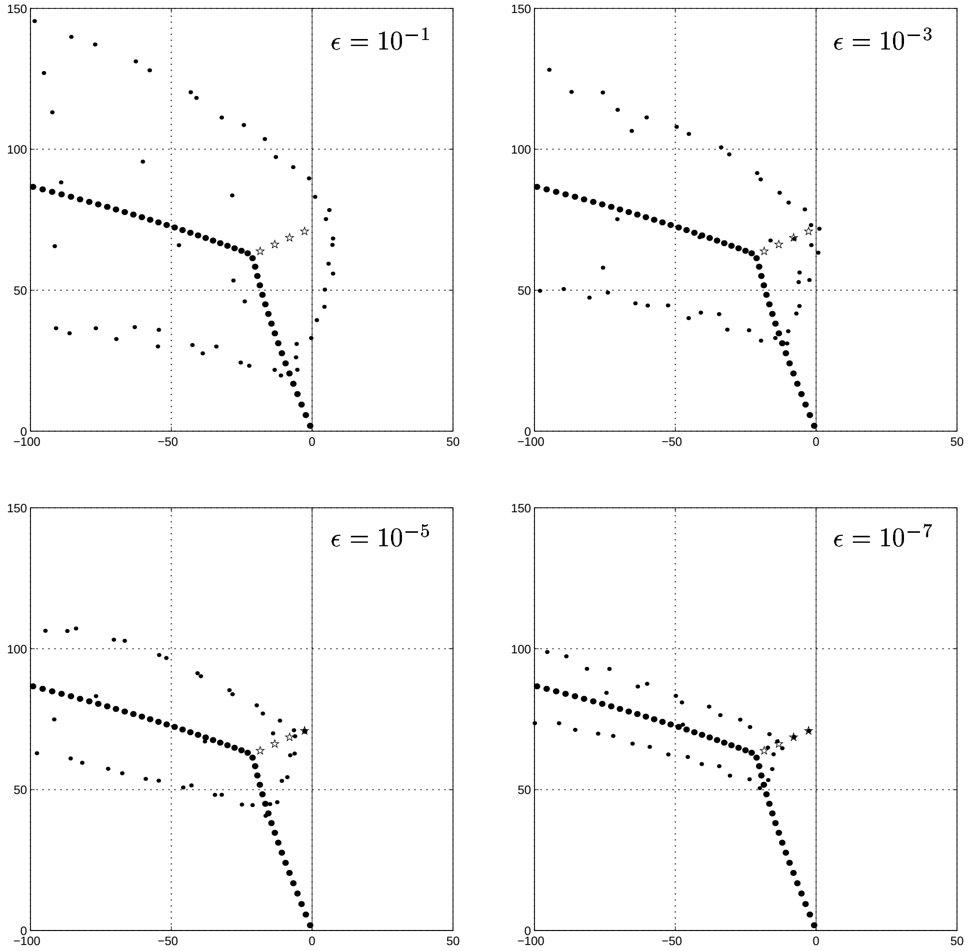

Fig. 4. Poor man's pseudospectra: eigenvalues of random rank one perturbations $B_{200} + E$, $\|E\|_2 = \epsilon = 10^{-1}, 10^{-3}, 10^{-5}, 10^{-7}$. The great sensitivity to perturbations confirms that B_{200} is a highly nonnormal matrix.

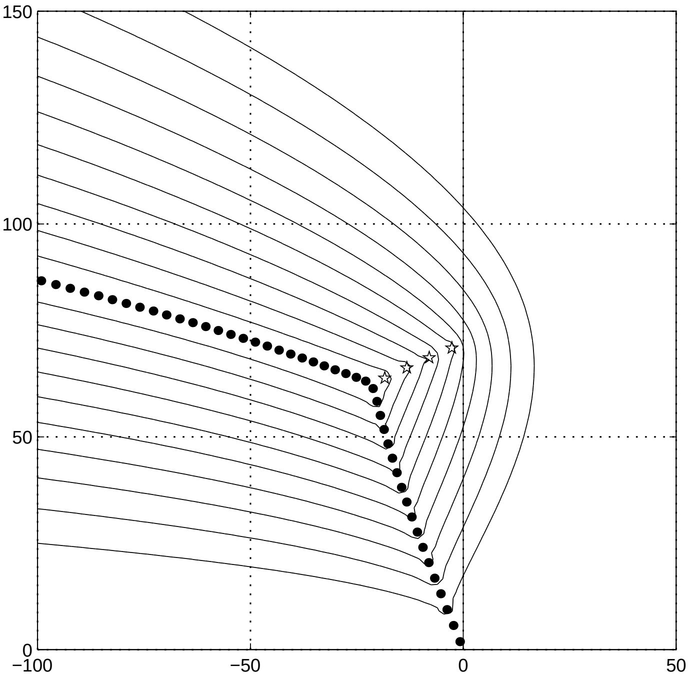

Fig. 5. Boundaries of ϵ-pseudospectra of the matrix $A = A_{200}$ for $\epsilon = 10^{-1}, 10^{-2}, \ldots, 10^{-10}$, from outside in. This is a fine picture, but producing it by the obvious SVD-based algorithm involving a 100×100 grid requires 4 hours of computing time on a SUN Ultra 30 workstation.

```
I = eye(size(B));
for j = 1:nu
  for i = 1:nu
    z = zz(i,j);
    sigmin(i,j) = min(svd(z*I-B));
  end
end
contour(x,y,log10(sigmin),-10:-1);
```

Figure 5 shows numerically computed pseudospectra for our matrix $B = B_{200}$. This is typical of dozens of images of pseudospectra that have appeared in the literature since (Trefethen, 1992). For this image, $\sigma_{\min}(z_{ij} - B)$ was evaluated for 10,000 points z_{ij} on a 100×100 regular grid in a square portion of the complex plane, and the resulting values were given as data to Matlab's contour plotter, just as in the code fragment above. (Whether or not one introduces the logarithm makes negligible difference.) We see at a glance that the eigenvalues in the two finite branches of the Y have sensitivities that increase as one approaches the fork, and that the eigenvalues along the infinite branch to the left of the fork have roughly constant sensitivities on the order of 10^{10}, as we knew already from Figure 3.

Figure 5 has a striking feature, which would prove important in many applications: though the spectrum is in the left half-plane, the pseudospectra protrude significantly into the right half-plane. We shall say more about this in Section 19.

The trouble with Figure 5 is that producing it by the method we have described involves a disturbingly long computation. Computing the SVD of an $N\times N$ matrix at each point on a $\nu\times\nu$ grid requires $\mathcal{O}(\nu^2N^3)$ floating point operations, and for $N = 200$ and $\nu = 100$, as in this figure, the computation time on my workstation works out to about 4 hours. For $N = 1000$, it would rise to three weeks—or possibly much longer because of memory limitations.

Of course, we can speed up the calculation by using a coarser grid, and in practice one would usually do this in the exploratory phase of any project. Figure 6 illustrates pseudospectra plotted on four different grids corresponding to $\nu = 5, 10, 20, 100$. Yet this set of plots mainly serves to emphasize the need for better computational methods. Roughly speaking, one might say that only the first of the four plots of Figure 6 is satisfactory in terms of computing time, and only the last is satisfactory in terms of appearance.

The next three sections will describe three ways to accelerate this computation, which can be used in combination. For our example, the speedups achieved are factors of approximately 1.5, 8, and 8, and when the methods are combined, we get a speedup by a factor of better than 60.

10. Avoiding uninteresting sections of the z-plane

The first way to speed up the calculation of pseudospectra is the simplest: avoid computing singular values in uninteresting regions of the complex plane, where the resolvent norm is small and there are no boundaries of the pseudospectra of interest. For our example, we note that in about a third of the portion of $\mathbb{C}$ shown in our plots, not much is happening, and it is a waste of time to evaluate $\|(z-A)^{-1}\|$. Bypassing this step should accelerate the computation by a worthwhile constant factor. We find that a crude device of this sort improves the computation time for Figure 5 from about 4 hours to 2.5 hours, a speedup by a factor of about 1.5.

Some interesting ideas for automating this kind of acceleration have been proposed by Gallestey (1997, 1998) under the name of the *SH algorithm* ('subharmonic'). Gallestey divides the region of $\mathbb{C}$ of interest into squares of various sizes and then uses the maximum principle for $\|(z-A)^{-1}\|$ to prune away squares automatically on which nothing interesting can be expected to be happening. Any 'industrial strength' software package for computing pseudospectra should incorporate ideas like these.

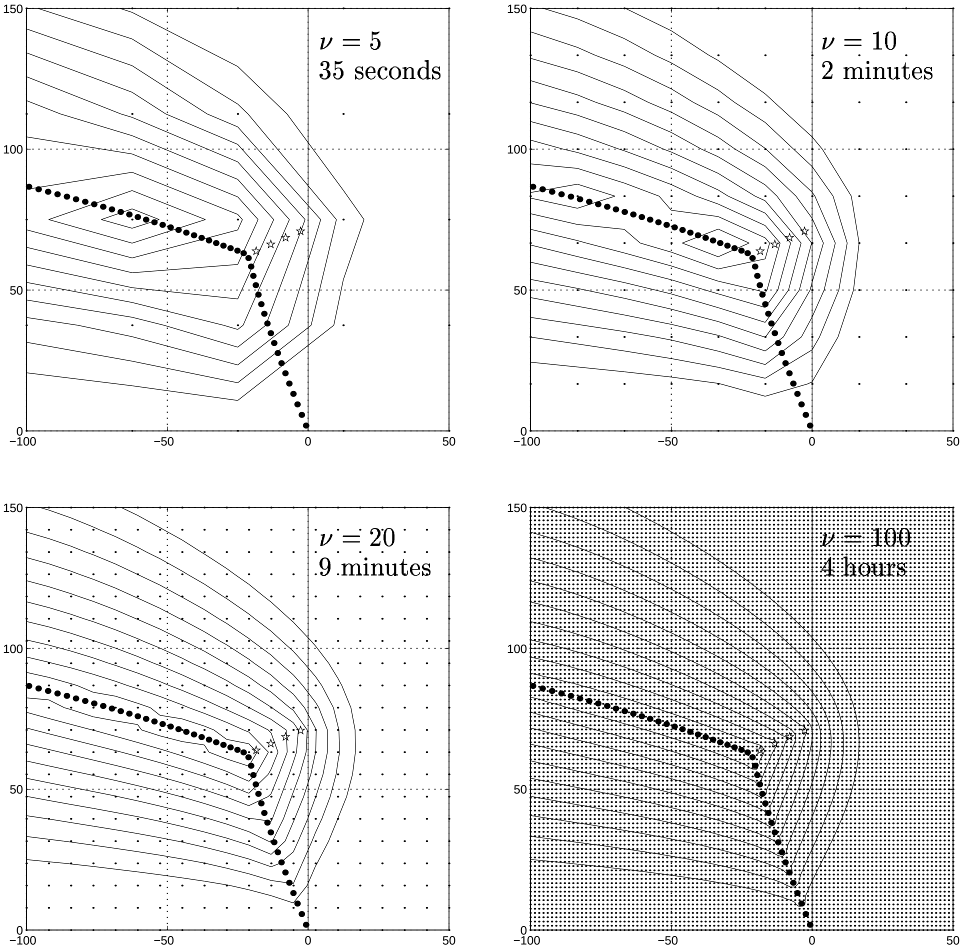

Fig. 6. Pseudospectra as in Figure 4 computed on successively finer grids; the points z at which $\|(z - A)^{-1}\|$ has been evaluated are marked by dots. For $\nu = 5$, there are just 25 SVDs to evaluate and the computation is fast, but the plot is crude. The 'publication quality' grid with $\nu = 100$ is prohibitively expensive.

A related, pointwise variant of the same idea is relevant to the various iterative methods for computing $\sigma_{\min}(z-B)$ discussed in Sections 12 and 16. Sometimes, we do not know in advance that a value z is uninteresting but discover this in the course of iteration—say, if it becomes clear that $\sigma_{\min}(z-B)$ is >1 and we only want to plot level curves below 10^{-1}. In this case it may be advantageous to terminate an iteration before convergence.

11. Projection to a lower-dimensional subspace

The second way to speed up calculation of pseudospectra, independent of the first, is to reduce the dimension of the $N \times N$ matrix A by orthogonal projection onto an invariant subspace of dimension $n < N$. The idea is that in many applications, most of the 'action' of interest can be captured by the lower-dimensional projection. This technique was perhaps first employed by Reddy, Schmid and Henningson (1993) and is described in Appendix B of that paper and in §6 of (Toh and Trefethen, 1996). It is elementary, but crucial in practice, and too often overlooked. We can often get an improvement in this way by a factor of 10 or more.

Following the authors just cited, we first describe a procedure of this kind based on explicit matrix diagonalization. Suppose V is an $N \times n$ matrix whose columns are selected linearly independent eigenvectors of B, satisfying $BV = VD$ for some $n \times n$ diagonal matrix D of corresponding eigenvalues. If $V = QR$ is a QR decomposition of V, with Q of dimension $N \times n$ and R of dimension $n \times n$ (Trefethen and Bau, 1997), then we have $Q^H V = R$ and $Q = VR^{-1}$ and therefore

$$Q^H B Q = Q^H B V R^{-1} = Q^H V D R^{-1} = R D R^{-1}.$$

Thus $T = RDR^{-1}$, which is an upper-triangular $n \times n$ matrix, is the matrix representation of the projection of B onto the subspace spanned by the selected eigenvectors. We can illustrate this projection process by the following MATLAB code segment, which projects B onto the invariant subspace corresponding to eigenvalues λ with $\operatorname{Re}\lambda > \gamma$ for some constant γ. Of course, different selections of special eigenvalues will be appropriate in other applications (see e.g. Section 21).

```
[V,D] = eig(B); eigB = diag(D);
select = find(real(eigB) > gamma);
V = V(:,select); D = D(select,select);
[Q,R] = qr(V,0);
T = R*D/R;
```

Figure 7 illustrates the effect of applying this projection to our matrix $B = B_{200}$ with $\gamma = -50$, -100, -150, and -250. As more eigenvalues are included, n rises from 37 to 53 to 66 to 92, but this is still far less than

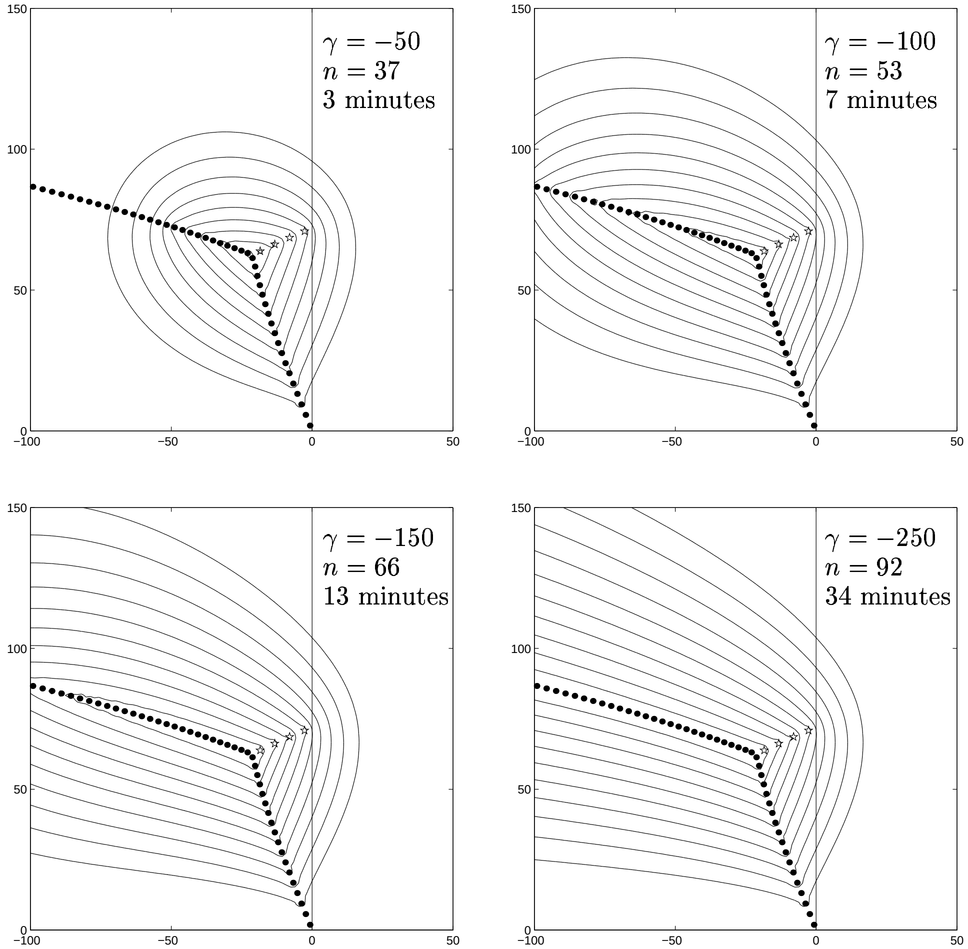

Fig. 7. Acceleration by preliminary projection onto an invariant subspace of dimension $n < N$. For this example we consider just eigenvalues of real parts $> \gamma$ for various γ.

200, and as the final operation count depends on n^3, it is a very significant improvement—about a factor of eight in this example.

A peculiar feature of the projection process just described is that it makes use of a matrix diagonalization. This sounds like a bad idea, since in applications B will often be highly nonnormal or even nondiagonalisable, implying that its eigenvalue problem may be very badly conditioned. In practice, it seems that the use of diagonalization does not cause much trouble, for reasons of backward error analysis. Though each individual numerically computed eigenvalue and eigenvector of a highly nonnormal matrix may be very much in error, their collective behaviour is generally better.

Nevertheless, it seems that in principle one ought to avoid the diagonalization, and this can be done by using a Schur decomposition (unitary triangularization) instead. Suppose a unitary similarity transformation is found of the form

$$B = Q \begin{bmatrix} T & X \\ 0 & Y \end{bmatrix} Q^H \tag{11.1}$$

where $Q \in \mathbb{C}^{N\times N}$ is unitary, $T \in \mathbb{C}^{n\times n}$ is upper-triangular, and $X \in \mathbb{C}^{n\times(N-n)}$ and $Y \in \mathbb{C}^{(N-n)\times(N-n)}$ are arbitrary. If $Q_1 \in \mathbb{C}^{N\times n}$ is the matrix consisting of the first n columns of Q, then (11.1) implies $BQ_1 = Q_1T$, which implies that if $Tx = \lambda x$, then $B(Q_1x) = \lambda(Q_1x)$. Thus the diagonal entries of T are n of the eigenvalues of B, and T is the projection of B onto the corresponding invariant subspace.

The factorization (11.1) is known as a *partial Schur decomposition* (Dongarra, Duff, Sorensen and van der Vorst, 1998; Lehoucq, Sorensen, and Yang, 1998). Since X and Y are arbitrary, all that is really involved here is the determination of an $N \times n$ matrix Q_1 with orthonormal columns such that $T = Q_1^H BQ_1$ is upper-triangular. Such a matrix might be found by various methods, but we shall consider just the simplest: computing a complete Schur decomposition and then reordering the diagonal entries to bring those of interest to the upper-left. Reorderings of this kind are a standard option in LAPACK (Anderson, *et al.,* 1995). They are not standard in the current version of MATLAB, but the desired effect can be achieved the following code segment adapted from programs of Diederik Fokkema (Fokkema, 1996; Fokkema, Sleijpen, and van der Vorst, 1999):

```
[U,T] = schur(B);
if isreal(B), [U,T] = rsf2csf(U,T); end, T = triu(T);
eigB = diag(T);
select = find(real(eigB) > gamma);
n = length(select);
for i = 1:n
  for k = select(i)-1:-1:i
```

```
    G([2 1],[2 1]) = planerot([T(k,k+1) T(k,k)-T(k+1,k+1)]')';
    J = k:k+1; T(:,J) = T(:,J)*G; T(J,:) = G'*T(J,:);
  end
end
T = triu(T(1:n,1:n));
```

Like the one given earlier, this code segment produces an upper-triangular matrix T corresponding to the projection of B onto the selected subspace.

Orthogonal projections have a monotonicity property: they never increase the resolvent norm at any point z. It follows that if $\Lambda_\epsilon(T)$ is the ϵ-pseudospectrum of the projected matrix, then $\Lambda_\epsilon(T) \subseteq \Lambda_\epsilon(B)$, with the ϵ-pseudospectra of T increasing monotonically to those of B as successively larger invariant subspaces are selected.

Our orthogonal projections can be viewed as a special case of a more general class of two-sided projections that may be applied for problems of computing pseudospectra. These have been studied under the name of *transfer functions* by Hinrichsen and Pritchard and Kelb (1992, 1993) and Simoncini and Gallopoulos (1998).

Finally, it should be mentioned that a different projection idea has also been advocated by Godunov and Sadkane (1996): the numerical use of resolvent integrals (Kato, 1976) for the computation of projections associated with subsets of $\mathbb{C}$.

12. Triangularization + inverse iteration or Lanczos

A third, major new idea for speeding up the computation of pseudospectra was introduced by S.-H. Lui in an article published in 1997 (Lui, 1997). Lui's method is described in his own paper and elsewhere as a method of 'continuation', but his key contribution is really the technique of *triangularization* followed by *inverse iteration* or *inverse Lanczos iteration.*

The idea is as follows. If B is a dense matrix, the computation of the smallest singular value of each $N \times N$ matrix $(z - B)^{-1}$ takes $\mathcal{O}(N^3)$ operations, for a total of $\mathcal{O}(\nu^2 N^3)$ operations on a $\nu \times \nu$ grid. However, suppose that before computing any singular values, we perform a Schur decomposition, with or without compression, to replace B by a unitarily equivalent upper-triangular matrix T. Then for any z, $z - B$ is unitarily equivalent to the upper-triangular matrix $z - T$, and hence will have the same singular values. Since $z - T$ is triangular, however, its smallest singular value can be computed in $\mathcal{O}(N^2)$ rather than $\mathcal{O}(N^3)$ operations. Thus at the price of a single computation involving $\mathcal{O}(N^3)$ operations, we have reduced the cost of each subsequent SVD to $\mathcal{O}(N^2)$. The overall improvement is from $\mathcal{O}(\nu^2 N^3)$ to $\mathcal{O}(N^3 + \nu^2 N^2)$ floating point operations, which for most applications is effectively an improvement to $\mathcal{O}(\nu^2 N^2)$.

If B has been orthogonally projected onto a lower-dimensional invariant subspace as described in the last section, then it has been rendered triangular already. In this case there is no need for a further Schur triangularization.

It remains to describe how $\sigma_{\min}(z - T)$ can be computed in $\mathcal{O}(N^2)$ operations. The idea for this is that $\sigma_{\min}(z - T)$ is the square root of the smallest eigenvalue of $(z-T)^H(z-T)$, and this can be computed by various iterations; since T is triangular, each step requires only $\mathcal{O}(N^2)$ operations. The simplest method is inverse iteration applied to $(z - T)^H(z - T)$, that is, power iteration applied to $(z - T)^{-1}(z - T)^{-H}$. (An early use of inverse iteration for computing pseudospectra, without triangularization, was by Baggett (1994).) For example, the following rather crudely put together MATLAB code segment is functionally equivalent to the shorter code on p. 17, but many times faster for matrices of larger dimensions.

```
I = eye(size(B));
[U,T] = schur(B);
if isreal(B), [U,T] = rsf2csf(U,T); end, T = triu(T);
for j = 1:nu
  for i = 1:nu
    z = zz(i,j);
    T1 = z*I-T; T2 = T1';
    v = randn(n,1) + sqrt(-1)*randn(n,1); v = v/norm(v);
    sigold = 0;
    for k = 1:99
      v = T1\(T2\v);
      sig = norm(v);
      if abs(sigold/sig-1) < .001 break; end
      sigold = sig;
      v = v/sig;
    end
    sigmin(i,j) = 1/sqrt(sig);
  end
end
contour(x,y,log10(sigmin+1e-20),-10:-1);
```

The main part of this code is a double loop just as on p. 17, except that inside the loop, $\sigma_{\min}(z_{ij} - B)$ is now computed by inverse iteration. The convergence criterion used here is crude: we stop when two successive estimates of $\sigma_{\min}(z_{ij} - B)^2$ agree to a tenth of a percent, taking up to a maximum of 99 steps.

This simple method does well in many cases, but as always with power iteration, the convergence may be slow if the dominant eigenvalue (of $(z - T)^{-1}(z - T)^{-H}$) is not well separated from the others. To retain speedy convergence in such cases one can replace the inverse iteration by an inverse

Lanczos iteration. Here is a modified Matlab fragment to achieve the desired effect:

```
I = eye(size(B));
[U,T] = schur(B);
if isreal(B), [U,T] = rsf2csf(U,T); end, T = triu(T);
for j = 1:nu
  for i = 1:nu
    z = zz(i,j);
    T1 = z*I-T; T2 = T1';
    sigold = 0; qold = zeros(n,1); beta = 0; H = [];
    q = randn(n,1) + sqrt(-1)*randn(n,1); q = q/norm(q);
    for k = 1:99
      v = T1\(T2\q) - beta*qold;
      alpha = real(q'*v); v = v - alpha*q;
      beta = norm(v); qold = q; q = v/beta;
      H(k+1,k) = beta; H(k,k+1) = beta; H(k,k) = alpha;
      sig = max(eig(H(1:k,1:k)));
      if (abs(sigold/sig-1)<.001) | (sig<3 & k>2) break, end
      sigold = sig;
      end
    sigmin(i,j) = 1/sqrt(sig);
  end
end
contour(x,y,log10(sigmin+1e-20),-10:-1);
```

Suppose we apply this code to the matrix $B = B_{200}$ of our example, using no other acceleration methods. We find that for most points z on the grid, 3 iterations are taken inside the inner loop, as illustrated in Figure 8. Thus the cost of each evaluation of $\sigma_{\min}(z - B)$ is essentially that of 6 triangular matrix solves. The computing time improves from 4 hours to about 29 minutes, a speedup by a factor of about 8. If uninteresting parts of the z-plane are avoided, the improvement is from 2.5 hours to 19 minutes, again a speedup by a factor of about 8. If in addition we first project B onto the subspace of dimension 92 associated with eigenvalues λ with $\operatorname{Re}\lambda > \sigma$ with $\sigma = -250$, as in the last section, the improvement is from 18 minutes to 3.7 minutes. This last speedup is by a factor of about 5, not 8, since the effectiveness of the preliminary triangularization is diminished for a matrix of dimension 92 rather than 200.

The Lanczos iteration we have just described is just one possibility for this kind of computation. Alternative methods have been studied in detail by Braconnier (1996, 1997), Braconnier and Higham (1996), Lui (1997), and Marques and Toumazou (1995, 1995b). Braconnier and Higham improve the Lanczos iteration by selective reorthogonalization and Chebyshev accel-

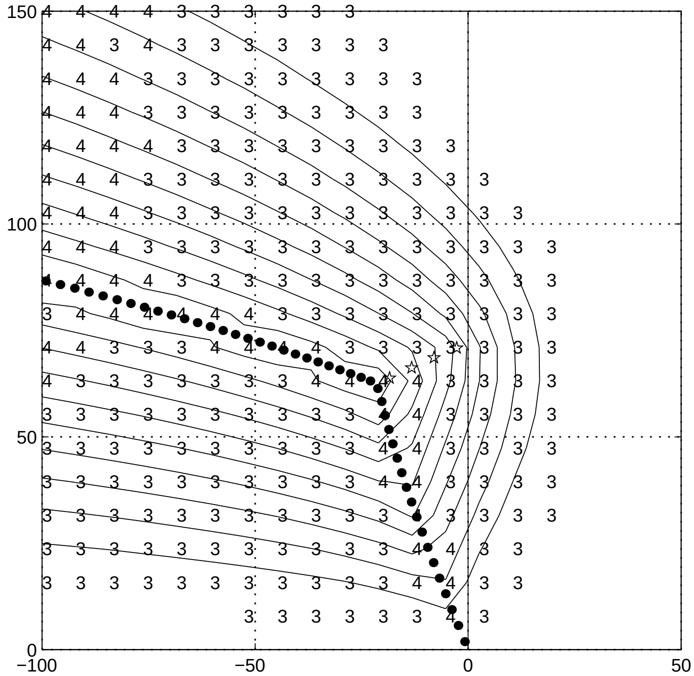

Fig. 8. Pseudospectra of the matrix $B = B_{200}$ computed on a 20×20 grid by projection to dimension $n = 92$ followed by inverse Lanczos iteration. The numbers of Lanczos steps at each point of the grid are marked, illustrating that even with cold starts for each z, 3 steps typically suffice for convergence. Blank sections of the plot correspond to areas of the complex plane that have been pruned away as described in Section 10. This computation took 30 seconds.

eration, and they emphasize the importance for robustness of a carefully designed and conservative convergence criterion. All of these authors use continuation from one point z to the next so that an iteration starts with a better than random initial guess. Since a cold start tends to produce convergence in three iterations, however, it seems that the use of continuation idea is not indispensable.

13. Summary of speedups discussed so far

For our tutorial example of dimension $N = 200$, the various algorithms we have described have performed roughly as follows:

straightforward use of SVD	4 hours
prune uninteresting portions of z-plane	2.5 hours
prune and project to $\operatorname{Re}\lambda > -250$	18 minutes
prune and use preliminary triangularization	19 minutes

combination of all speedups	3.7 minutes

By a succession of three improvements implementable in 40 lines of Matlab (see Section 20), we have reduced the computation time for Figure 5 from 4 hours to 4 minutes. This factor of 60 is comparable to the factor by which workstations have speeded up in the years since (Trefethen, 1992), signalling the roughly equal roles of hardware and algorithmic improvements in the field of computation of pseudospectra.

Of course, to assess various algorithms systematically one would like asymptotic formulas rather than examples. Unfortunately, outside of the context of a particular class of matrices, it is hard to see how to derive asymptotic formulas with much substance for the computation of pseudospectra of matrices. How much can one gain by projection to a lower-dimensional subspace? It depends on how far the dimension can be lowered, and this depends on the problem at hand, not on any general parameters.

Nevertheless we offer this rough guide to the improvement factors that seem to be achievable for many examples:

prune uninteresting portions of z-plane	cuts no. of grid points in half
project to interesting subspace	cuts N in half
preliminary triangularization	speeds up by factor $N/30$
combination of all three	speeds up by factor $N/4$

This last figure $N/4$ is roughly the product of the three speedup factors $2^3 = 8$, 2, and $(N/2)/30$. We may call it a *Rule of Thumb*. For a typical problem of size $N = 1000$, for example, one should expect to be able to compute a publication-quality plot of pseudospectra with about 1/250 as much work as by using the algorithm of p. 17.

14. Parallel computation of pseudospectra

Much further speedups are available via a fourth method: the use of multiple processors. In many situations the computation of pseudospectra falls in the class of problems known as *embarrassingly parallel.* This means that the computation decouples so readily that taking advantage of multiple processors requires little effort. The first parallel computations of pseudospectra appear to have been those by A. E. Trefethen reported in (Trefethen, Trefethen, Reddy, and Driscoll, 1993), and subsequent contributions in this area have been due to Braconnier (1996), A. E. Trefethen, *et al.* (1996), Frayssé, Giraud, and Toumazou (1996), Heuveline, Philippe, and Sadkane (1997), Trefethen, Trefethen and Schmid (1999), and Bekas and Gallopoulos (1999).

In the simplest case, suppose one is computing a plot of pseudospectra by evaluating $\sigma_{\min}(z_{ij} - B)$ on a grid of points z_{ij} by means of a standard SVD.

The work involved is approximately independent of z_{ij}, and accordingly, all one needs to do to take advantage of p processors is divide the points $\{z_{ij}\}$ at the beginning into p sets of roughly equal size. Each processor needs to know its set of points and the matrix B, but no communication or synchronization is needed until the computation is finished.

A slightly more complicated situation may arise if the work varies significantly with z_{ij}, which may happen in the context of inverse iteration or Lanczos iteration, or in any problem where the numerical discretization itself depends on z_{ij}, as may easily occur if, for example, values z with $|z| \gg 1$ require finer grid resolutions than those near the origin. In this case, if the points z_{ij} are to be treated independently, then load-balancing can be achieved by maintaining a list of points z_{ij} not yet treated and assigning a new point to a processor whenever it finishes with an old one. If the treatment of the points is dependent, which might be the case because of some kind of initial guess continued from point to point, then the management of these lists will benefit from some geometric structuring.

We will not go into further details, but just summarize the subject of parallel computation of pseudospectra with the statement that if p processors are available, one can usually achieve a speedup by a factor close to p.

15. Global Krylov subspace iterations

Up to now, we have discussed methods belonging to the realm of dense linear algebra, where all N^2 entries of a matrix are manipulated explicitly and fundamental matrix calculations require $\mathcal{O}(N^3)$ operations. However, many people have had the idea that the well-developed techniques of Krylov subspace iterations should also have a role to play in computing pseudospectra, and for matrices of dimensions in the thousands, this conclusion seems inescapable. For information on Krylov subspace iterations see (Barrett, *et al.*, 1994), (Dongarra, Duff, Sorensen and van der Vorst, 1998), (Greenbaum, 1997), (Lehoucq, Sorensen, and Yang, 1998), (Saad, 1992), and (Trefethen and Bau, 1997).

The very many ideas of this kind that might be considered fall roughly into two classes. One can attempt to approximate the pseudospectra of a matrix or operator all at once with a single sequence of Krylov subspaces, or one can use Krylov methods pointwise to accelerate the computation of resolvent norms for individual values of z. In this section we consider the first of these ideas. (In the end the greatest power may come from combining the two, working locally with small regions of the z plane but not individual points z.)

The motivation for methods of this kind is that Krylov subspace iterations extract essential information from a matrix within the context of low-dimensional subspaces; they are projection processes closely related to those discussed in Section 11. Instead of computing just eigenvalues (Ritz values) in these subspaces, why not compute pseudospectra? Preliminary ideas in this direction can be found in (Freund, 1992) and in (Nachtigal, Reichel, and Trefethen, 1992), and the method has been explored further in (Toh and Trefethen, 1996) and in (Simoncini and Gallopoulos, 1998).

The simplest procedure, as described by Toh and Trefethen, goes as follows. Starting from a random initial vector, we carry out an Arnoldi iteration in the usual manner, obtaining thereby an increasing sequence of initial columns of a Hessenberg matrix unitarily similar to B. We then compute pseudospectra of successive sections of this Hessenberg matrix, and take these as approximations to the pseudospectra of B. Rectangular sections with dimensions of the form $(n+1) \times n$ are more appropriate than square ones; the pseudospectra of a rectangular matrix B can be defined via (3.3) or, equivalently, via the pseudoinverse of $z - B$ (Toh and Trefethen, 1996). (It will be interesting to see whether pseudospectra of rectangular matrices achieve importance in other contexts in the years ahead.)

Figure 9 gives an indication of how this method performs for our example problem. Starting from the full matrix $B = B_{200}$, pseudospectra are plotted corresponding to Krylov subspace approximations of dimensions 120, 140, 160, and 180. The results are disappointing. Not until n is close to 200 do the pictures look reasonable, and of course, in numerical computation one wants more than merely something that looks reasonable. What has gone wrong is that the spectrum of B_{200} extends very far into the left half-plane, with a leftmost (nonphysical) eigenvalue at about $-700{,}000 + 300\mathrm{i}$, which will only get worse if N is increased. Under such circumstances a straightforward Arnoldi iteration has little chance of capturing the interesting behaviour near the origin efficiently. One can improve the situation in various ways, for example by working with B^{-1} instead of B, but this is not an easily used general technology.

For large-scale problems, the first thing one might do in the exploratory phase of a computational project involving highly nonnormal matrices should perhaps be a computation of the kind described in this section. Indeed, it might be argued that pictures of estimated pseudospectra should be a routine by-product of all large-scale Krylov subspace calculations; the dimensions of the Hessenberg matrices will usually be low, so the cost will be small. If one decides that accurate pictures of pseudospectra are needed, however, in most cases one will want to move on to other methods.

16. Local Krylov subspace iterations

Krylov subspace methods are much more powerful than the last section may seem to suggest. The crucial modification is that they be applied for individual points $z \in \mathbb{C}$, or in localized regions. Then one has the

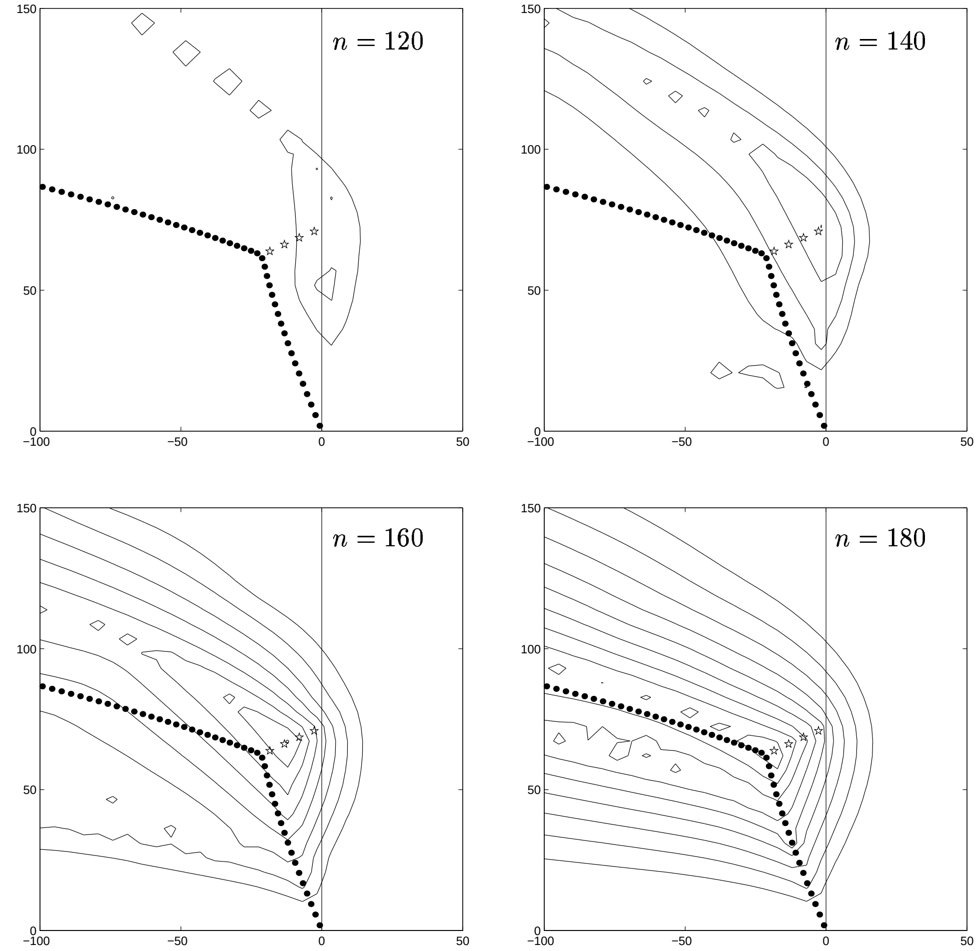

Fig. 9. Pseudospectra of Arnoldi projections of various dimensions for $B = B_{200}$. Convergence takes place eventually, but it is too slow for Krylov subspace iterations in this pure form to be of much use for this example.

potential for convergence to arbitrary accuracy at high speed, and for large-scale problems, these are the most powerful methods known.

I will not attempt to describe these methods in any detail, as my experience in this area is small and new developments are occurring very fast. However, here is a quick outline. One of the first papers to discuss methods of this kind was by Carpraux, Erhel, and Sadkane (1994), who used a Davidson iteration with continuation (Davidson, 1975). This method was subsequently parallelized by Heuveline, Philippe, and Sadkane (1997) and applied by them to a matrix of dimension $N = 8192$. Other contributions in this area are due to Lui (1997), Braconnier (1996, 1997), Braconnier and Higham (1996), and Ruhe (1994, 1998), who has developed a *rational Krylov* algorithm. In this and other computations it is crucial to use exact or approximate inverses of the matrix being analysed, wherever possible, as this may greatly speed up the convergence. The starting point is the idea of shift-and-invert Arnoldi iteration, but from there, many different paths can be taken.

One of the leading projects currently underway for large-scale computation of pseudospectra is being carried out by numerical analysts and plasma physicists at the University of Utrecht and the CWI in the Netherlands (van Dorsselaer, Goedbloed, Nool, van der Ploeg, van der Vorst and others). In a research project on the stability of plasmas, these researchers have succeeded in computing the eigenvalues of generalized unsymmetric eigenvalue problems, associated with Tokamak plasmas, of order up to 262,144. On a CRAY T3E, a relevant part of the so-called Alfvén spectrum (12 eigenvalues) could be computed in 7 seconds of wall-clock time. This was done with the Jacobi–Davidson method (Dongarra, Duff, Sorensen and van der Vorst, 1998; Sleijpen and van der Vorst, 1996), using a direct decomposition of a shifted matrix for one single shift in the neighbourhood of the desired part of the spectrum. The group in Utrecht has now started work on the evaluation of pseudospectra for this problem, also with the Jacobi–Davidson method and with similar preconditionings as for the generalized eigenproblem. The idea is that one single preconditioner can be used for a portion of the pseudospectra, and the research focuses on the efficient re-use of search subspaces in sweeping over the spectrum with the Jacobi–Davidson method.

Continuation of data from point to point appears to be more important for these large-scale Krylov subspace iterations than for the dense matrix computations discussed in Section 12. The reason is that the large-scale methods depend crucially on the use of subspaces that get enlarged and deflated as the iteration proceeds. To evaluate a resolvent norm $\|(z-B)^{-1}\|_2$, it may save a great deal of work if one starts with the subspaces already determined for a neighbouring point z'.

17. Curve-tracing for pseudospectral boundaries

Quite a different approach to producing plots of pseudospectra has been proposed by Kostin (1991) and worked out in detail by Brühl (1996). Rather than use a contour plotter with data based on a grid, why not trace the boundary curves of the pseudospectra directly? Such a technique has two potential advantages. One is that we can determine the boundary curves to great accuracy, if that is desired. The other is that fewer evaluations of $\sigma_{\min}(z-B)$ may be needed since no grid is involved.

Brühl put this idea into practice with a Newton iteration at each step and showed that it can be effective. An appealing feature of his method is that any speedups one gets in this fashion can be combined with those provided by methods for accelerating the computation of $\sigma_{\min}(z-B)$, such as projection and Lanczos iteration.

For general use, the method of curve tracing runs into questions of how to handle corners and, more seriously, how to cope robustly with pseudospectra that have two or many components. It is very attractive, however, for problems where one wants to concentrate accurately on particular sections of the boundaries of pseudospectra. For example, by a method of this kind one can design a program that enables the user to click with the mouse at a point in the complex plane and have the computer draw the boundary of a pseudospectrum that passes through this point. This can be informative and beautiful graphically, and it can provide an elegant route to computational estimates of matrix functions of interest based on contour integrals. One could click on a point z, for example, and see not only the pseudospectral boundary that passes through z but also some numerically computed upper and lower bounds based on that curve.

Brühl's work on curve-tracing methods has been carried further by Bekas and Gallopoulos (1998, 1999) in a method called COBRA. These authors combine curve-tracing and grid methods, using 'a small, moving grid that follows the boundary $\partial\Lambda_\epsilon$, almost like the head of a cobra that follows the movements of its prey'. This hybrid approach, they argue, offers the advantages of curve-tracing combined with greater robustness and opportunities for parallelism.

18. Pseudospectra in Banach spaces

For a fixed matrix, all norms are equivalent, and thus the resolvent norms associated with two different norms differ at most by constants. Since the effects of interest in plotting pseudospectra are often in some sense exponentially strong, it follows that in many cases, the pseudospectra of a matrix do not change much when one switches, say, from $\|\cdot\|_2$ to $\|\cdot\|_1$ or $\|\cdot\|_\infty$. In such cases the choice of norm may not be too important. (More extreme changes of norm may have more extreme effects; after all, the pseudospectra can be rendered trivial by the switch to a norm defined by the coefficients in an expansion in eigenvectors.)

It would be a mistake to presume, however, that the difference between $\|\cdot\|_2$ and $\|\cdot\|_1$, say, or more generally between Hilbert spaces and Banach spaces, is always minor. Once one is dealing with operators of infinite dimension or their matrix discretizations, the gaps between p-norms may be arbitrarily large, and in some cases the 'physics' of the problem lies in the gap.

Pseudospectra in non-Euclidean norms are discussed from a theoretical point of view in (van Dorsselaer, Kraaijevanger, and Spijker, 1993), and an example of a paper in which they are computed numerically is the study of Abel integral operators by Plato (1997). My own conversion to the importance of this subject came with a study of the 'cutoff phenomenon' that occurs in certain Markov chains (Diaconis, 1996). Diaconis and others have shown that for various random processes such as random walk on a hypercube (Diaconis, Graham, and Morrison, 1990) and riffle shuffling of a deck of cards (Bayer and Diaconis, 1992), convergence to a uniform probability distribution, when measured in a certain way, occurs not gradually but in a sudden fashion after a certain number of steps. Since the processes in question involve powers of matrices, this nonsteady behaviour suggests that pseudospectra must be important. The matrix dimensions in these problems are sometimes combinatorially large, however, and in such instances it may be crucial to use $\|\cdot\|_1$ (the natural norm for probability) rather than $\|\cdot\|_2$. Indeed, the matrices of interest are sometimes normal with respect to the Euclidean norm. For example, the problem of random walk on an n-dimensional hypercube leads to a matrix of dimension $N = 2^n$. The matrix is real and symmetric, hence normal, so a normalized matrix of eigenvectors has $\kappa_2(V) = 1$ in the 2-norm and uninteresting pseudospectra. In the 1-norm, however, we get $\kappa_1(V) \approx 10^6$ for n as low as 40, and the corresponding pseudospectra reach outside the unit disk. Similarly, the problem of riffle shuffling of a deck of 52 cards leads to a matrix of dimension $52! \approx 8 \times 10^{67}$ with $\kappa_1(V) \approx 10^{40}$ (Jónsson and Trefethen, 1998). The largest eigenvalue is $\lambda = 1/2$, but the pseudospectra protrude outside the unit disk.

The obvious algorithm for computing pseudospectra with respect to $\|\cdot\|_1$ or $\|\cdot\|_\infty$ requires the inversion of $z - A$ at a cost of $\mathcal{O}(N^3)$ flops at each point z. The acceleration methods we have described do not apply directly in this case, but it is possible that they could be adapted to this purpose by the use of dual norms. Very recently, the first contribution that I know of to the fast computation of pseudospectra in $\|\cdot\|_1$ or $\|\cdot\|_\infty$ has appeared, by Higham and Tisseur (1999). These authors combine two ideas with impressive results. The first is the Schur reduction of A to triangular form, as in Section 12; in contrast to the situation with $\|\cdot\|_2$, here we must retain the matrix Q of the reduction $A = QTQ^*$ for use in further computations, as the unitary similarity transformation does not leave $\|\cdot\|_1$ or $\|\cdot\|_\infty$ invariant. The second is an iteration to determine the norm of the inverse of a triangular matrix that is developed as a fast algorithm for condition number estimation based on block matrices. Aside from the treatment of nonstandard norms, one of the useful features of the work by Higham and Tisseur for readers of the present paper is that it relates the computation of pseudospectra to the estimation of condition numbers.

19. Pseudospectra and behaviour

Now, then, briefly, what is the purpose of all these computations of pseudospectra?

Eigenvalues are generally computed for one or both of two reasons: to aid in the solution of a problem via diagonalization, or to give insight into how a system behaves (Trefethen, 1997). Important examples of behaviour are stability or resonance for various physical or numerical processes and speed of convergence for numerical iterations. Behavioural phenomena are typically quantified by norms of functions of the matrix or operator in question, such as $\|A^n\|$, $\|\exp(tA)\|$, or $\|p(A)\|$, where p is a polynomial or a rational function. If A is an unbounded operator, as with our example (4.1), the notion of $\exp(tA)$ can be made rigorous by various methods considered in the theory of semigroups (Kato, 1976; Pazy, 1983).

If A is far from normal, pseudospectra are likely to do better than eigenvalues alone in the second of these two roles. It is known that pseudospectra cannot in general give exact information about norms of functions of matrices or operators (Greenbaum and Trefethen, 1993). However, they may provide bounds that are much sharper than those obtained from eigenvalues. Some such bounds are described in (Trefethen, 1997), and examples can be found in many of the articles cited in the introduction.

Here we will not discuss these matters in generality but just illustrate what the pseudospectra of our example operator A of (4.1) may reveal about its time evolution. To be specific, suppose we are interested in the linear evolution process $\mathrm{d}u/\mathrm{d}t = Au$, with solution $u(t) = \exp(tA)u(0)$. Looking first at the spectrum of A, we note that the rightmost eigenvalue in the complex plane is $\lambda = -0.7803 + 1.8951\mathrm{i}$ (labelled 1 in Figure 1), and the second-rightmost is $\lambda = -2.3246 + 5.6695\mathrm{i}$. These numbers appear to suggest that the evolution of this system will exhibit gentle decay at a rate approximately $e^{-0.8t}$, with the dominant modes being smooth ones.

A glance at the pseudospectra in Figure 5 suggests a different time evolution. In fact, the most conspicuous part of the behaviour of this process will be associated with the nearly degenerate eigenvalues along the starred branch, of which the rightmost is the mode 3/4 pair, with $\lambda \approx -2.6809 + 70.8747\mathrm{i}$. Because the pseudospectra in this part of the plane

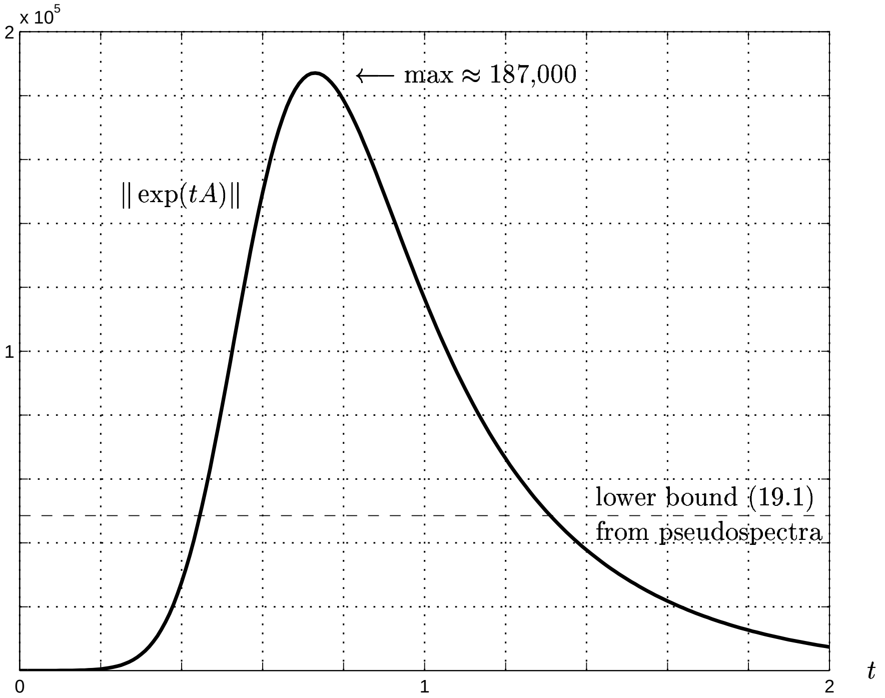

Fig. 10. Transient behaviour of $\|\exp(tA)\|$: the actual curve, and the lower bound $\mathcal{K}$ obtained from pseudospectra. The pseudospectra correctly capture the transient growth of order 10^5.

protrude so strongly into the right half-plane, the evolution process will be susceptible to large transient effects, and the structures involved will have frequencies closer to 70 than 0. The low-frequency modes 1 and 2 will be significant only for long time integrations, and then only if the dynamical system is purely linear, governed by A alone, with no variations of coefficients or forcing terms or nonlinearities.

Figure 10 shows a plot of $\|\exp(tA)\|$ against t. We see that there is very strong transient amplification of some initial vectors, rising to a maximum of about 187,000 around $t = 0.73$. The order of magnitude of this growth can be predicted from the pseudospectra. For any A and ϵ, if $\Lambda_\epsilon(A)$ extends a distance η into the right half-plane, then it can be shown by a Laplace transform that $\|\exp(tA)\|$ must be as large as η/ϵ for some $t > 0$. Given A, let the *Kreiss constant* $\mathcal{K}$ for A be defined as the supremum of this ratio over all ϵ (Kreiss, 1962). Equivalently, $\mathcal{K}$ is the smallest constant such that

$$\|(z - A)^{-1}\| \leq \frac{\mathcal{K}}{\operatorname{Re} z}$$

for all z with $\operatorname{Re} z > 0$. Then the inequality just mentioned takes the form

$$\sup_{t\ge 0} \|\exp(tA)\| \ge \mathcal{K}. \tag{19.1}$$

For our example the Kreiss constant is approximately $\mathcal{K} = 48570$, attained for $z = 1.25+68.88\mathrm{i}$, with $\|(z-A)^{-1}\| = 38850$. The dashed line in Figure 10 marks this lower bound.

The size of the transient hump in $\|\exp(tA)\|$ is only one of the aspects of the behaviour of A that pseudospectra may shed light on. Another would be the response of a system governed by this operator to oscillatory inputs at various real frequencies ω, corresponding to points on the imaginary axis of Figure 5. Judging by the spectrum, one would guess that only frequencies $\omega \approx 0$ or $\omega \approx 70$ should excite much response, but the pseudospectra imply that amplifications on the order of 10^3 or more can be expected for the full range of frequencies $\omega \in [40, 80]$. This phenomenon of *pseudo-resonance* and other aspects of the physics of nonnormality are discussed in Trefethen, Trefethen, Reddy and Driscoll (1993), Butler and Farrell (1992), and Farrell and Ioannou (1996).

More generally, pseudospectra may provide bounds on $\|f(A)\|$ for any function f. The most general results along this line are to be found in Toh and Trefethen (1999b), where relationships are derived between the size of $\|f(A)\|$, the size of $f(z)$ on a complex domain Ω, and the Kreiss constant of A with respect to Ω.

20. A Matlab program

An historic event in numerical computation was the codification of algorithms for computing matrix eigenvalues into the Fortran software package EISPACK in the 1970s (Smith *et al.*, 1976). Major algorithmic advances had been made in that subject in the preceding decade and a half, which had advanced the state of the art far beyond what an average scientist could expect to program for him- or herself, and equally important, these were problems for which potential users *knew* that they needed help. The impact of EISPACK was enormous: a set of problems that had earlier been challenging were reduced very quickly, as it were, to a black box.

The problem of computing pseudospectra is not yet in a comparable situation. The problem is too new and too rapidly changing, and the algorithms are not yet sufficiently well worked out for it to be appropriate to aim for black boxes in this area. Nevertheless, in a modest way, small-scale software for computing pseudospectra may prove useful. The Matlab program of Figure 11 is one that I hope readers may find helpful, after adapting the details to their needs, as a starting point for dense matrix computations. (The code is available at `www.comlab.ox.ac.uk/oucl/people/nick.trefethen.html`.) Its main purpose is to show how much better one can do for many

```
% psa.m - Simple code for 2-norm pseudospectra of given matrix A.
%         Typically about N/4 times faster than the obvious SVD method.
%         Comes with no guarantees!   - L. N. Trefethen, March 1999.

% Set up grid for contour plot:
  npts = 20; s = .8*norm(A,1);               % <- ALTER GRID RESOLUTION
  xmin = -s; xmax = s; ymin = -s; ymax = s;  % <- ALTER AXES
  x = xmin:(xmax-xmin)/(npts-1):xmax;
  y = ymin:(ymax-ymin)/(npts-1):ymax;
  [xx,yy] = meshgrid(x,y); zz = xx + sqrt(-1)*yy;

% Compute Schur form and plot eigenvalues:
  [U,T] = schur(A);
  if isreal(A), [U,T] = rsf2csf(U,T); end, T = triu(T); eigA = diag(T);
  hold off, plot(real(eigA),imag(eigA),'.','markersize',15), hold on
  axis([xmin xmax ymin ymax]), axis square, grid on, drawnow

% Reorder Schur decomposition and compress to interesting subspace:
  select = find(real(eigA)>-250);            % <- ALTER SUBSPACE SELECTION
  n = length(select);
  for i = 1:n
    for k = select(i)-1:-1:i
      G([2 1],[2 1]) = planerot([T(k,k+1) T(k,k)-T(k+1,k+1)]')';
      J = k:k+1; T(:,J) = T(:,J)*G; T(J,:) = G'*T(J,:); end, end
  T = triu(T(1:n,1:n)); I = eye(n);

% Compute resolvent norms by inverse Lanczos iteration and plot contours:
  sigmin = Inf*ones(length(y),length(x));
  for i = 1:length(y)
    if isreal(A) & (ymax==-ymin) & (i>length(y)/2)
      sigmin(i,:) = sigmin(length(y)+1-i,:);
    else
      for j = 1:length(x)
        z = zz(i,j); T1 = z*I-T; T2 = T1';
        if real(z)<100                       % <- ALTER GRID POINT SELECTION
          sigold = 0; qold = zeros(n,1); beta = 0; H = [];
          q = randn(n,1) + sqrt(-1)*randn(n,1); q = q/norm(q);
          for k = 1:99
            v = T1\(T2\q) - beta*qold;
            alpha = real(q'*v); v = v - alpha*q;
            beta = norm(v); qold = q; q = v/beta;
            H(k+1,k) = beta; H(k,k+1) = beta; H(k,k) = alpha;
            sig = max(eig(H(1:k,1:k)));
            if (abs(sigold/sig-1)<.001) | (sig<3 & k>2) break, end
            sigold = sig; end
         %text(x(j),y(i),num2str(k))         % <- SHOW ITERATION COUNTS
          sigmin(i,j) = 1/sqrt(sig); end, end, end
    disp(['finished line ' int2str(i) ' out of ' int2str(length(y))]), end
  contour(x,y,log10(sigmin+1e-20),-8:-1);    % <- ALTER LEVEL LINES
```

Fig. 11. Fast (not robust) Matlab code for pseudospectra of dense matrices.

problems than to use the obvious SVD algorithm. But this is nothing like robust software, and makes no claim to be. This code is filled with arbitrary features that can easily be broken.

Just one item of software has achieved something like general use for the calculation of pseudospectra: the MATLAB program `pscont` from the Test Matrix Toolbox of Higham (1995). That code makes use of a straight SVD algorithm, however, and thus is not as fast as we would like. Other authors have taken steps to develop software for high-performance computations, but none appear to be in wide use at present.

21. Another example

I would like to finish by presenting a plot of pseudospectra of a second example operator, one with a special meaning for this field. So far as I know, the first person to define the notion of pseudospectra was Henry Landau at Bell Laboratories in the 1970s, who was motivated in part by applications in lasers and optical resonators (Landau 1975, 1976, 1977). One of the operators that Landau considered in detail was the complex symmetric (but non-Hermitian) compact integral operator

$$Au(x) = \sqrt{iF/\pi} \int_{-1}^{1} e^{-iF(x-y)^2} u(y) dy, \tag{21.1}$$

acting on functions in $L_2[-1,1]$, where F is a large parameter, the *Fresnel number* (Landau, 1976, 1977). This operator is easily described in words: it convolves a function on $[-1,1]$ with a high-frequency complex Gaussian. The eigenvalues lie on a spiral in the unit disk that converges to the origin (Cochran and Hinds, 1974), but Landau proved that for any $\epsilon > 0$, the ϵ-pseudospectrum $\Lambda_\epsilon(A)$ contains the entire unit disk, for all sufficiently large F. He further showed that each z with $|z| \le 1$ has an ϵ-pseudo-invariant subspace of dimension at least $\mathcal{O}\left(\sqrt{F}\right)$ as $F \to \infty$.

Twenty-two years later, Andrew Spratley and I have carried out numerical computations of the pseudospectra of this operator considered by Landau. We use a spectral collocation discretization much as in Section 5; the details will be reported elsewhere. The eigenfunctions and pseudoeigenfunctions are highly oscillatory, and fine grids are needed to resolve them. For the case $F = 64$, we find that an $N \times N$ matrix with $N = 600$ suffices for a good picture of pseudospectra, shown in Figure 12.

With $N = 600$ rather than $N = 200$ as before, the evaluation of a single resolvent norm $\|(z - A)^{-1}\|$ for Figure 12 takes about 27 times longer than for Figure 5, about 15 seconds on the SUN Ultra 30. Furthermore, because of the fine structure to be resolved in the plot, we have used a 200×200 rather than 100×100 grid. This means that the total time to compute Figure 12 by the obvious SVD algorithm should be about 100 times greater

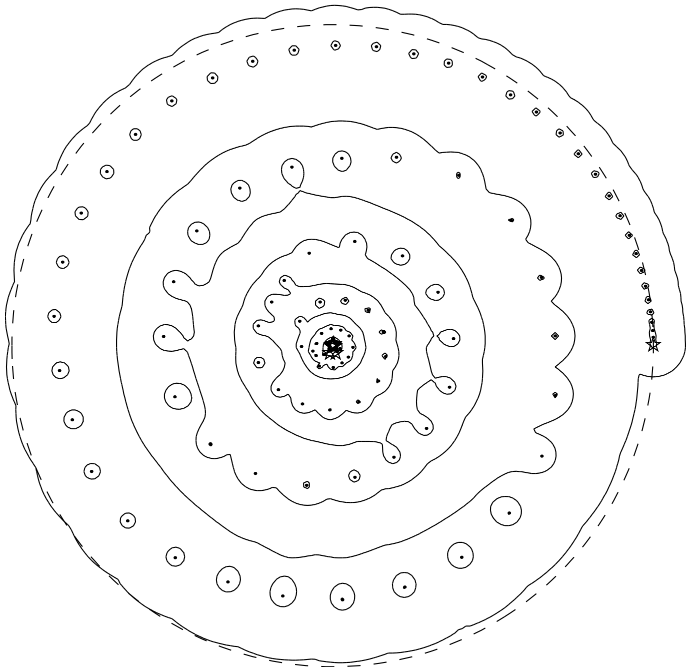

Fig. 12. Spectrum and ϵ-pseudospectra of the operator (21.1) ($F = 64$) studied in Landau's original work on pseudospectra, for $\epsilon = 10^{-1}, 10^{-2}, \ldots, 10^{-8}$. The dashed curve is the unit circle. Unlike the differential operator (4.1), this integral operator is compact, and the eigenvalues spiral in to the origin. As $F \to \infty$, for any $\epsilon > 0$, $\Lambda_\epsilon(A_F)$ converges to the disk $|z| \leq 1 + \epsilon$. This figure, based on a spectral discretization of dimension $N = 600$, is probably accurate to plotting accuracy except in a central region of radius about 0.1.

than the previous figure of 4 hours, that is, about 15 days. According to the Rule of Thumb of Section 13, however, it should be possible to speed this up by a factor of about $N/4 \approx 150$ by the dense matrix methods described in Sections 10–12 and in the Matlab program of Figure 11, bringing the computation time down to two or three hours. We used a projection onto the space spanned by eigenvectors associated with eigenvalues λ with $|\lambda| > 0.0001$. This reduced the matrix dimension to $N = 161$, and in the event, the computation took about 1 hour.

22. Discussion

The computation of pseudospectra is only a decade old. Remarkable progress has been made in the 1990s; of the 107 items in the list of references below, 93 date from this decade!

In this survey I have concentrated mostly on techniques for dense matrices, typically those of dimension less than 1000. Many of the developments to come in the years ahead will pertain to the other case of sparse or structured matrices of larger dimensions, where variations on the themes of Arnoldi, Jacobi–Davidson and rational Krylov iterations are powerful.

Our discussion has been confined to the standard matrix problem $Ax = \lambda x$, i.e., pseudospectra of matrices or operators, but in many applications it may be more appropriate to consider the generalized problem $Ax = \lambda Bx$, i.e., pseudospectra of matrix or operator pencils (van Dorsselaer, 1997; Frayssé, Gueury, Nicoud, and Toumazou, 1996; Frayssé and Toumazou, 1996; Riedel, 1994).

Computing pseudospectra is not yet a routine matter among scientists and engineers who deal with nonnormal matrices, but I think it will become so. The nature of the software that is available will play a decisive role in determining how the field develops. As time goes by, more software products for large-scale eigenvalue computations will appear, descendents of today's codes such as ARPACK (Lehoucq, Sorensen, and Yang, 1998), and these will show an increasing emphasis on graphical interaction with the user. In such an environment it is inevitable that scientists will be encouraged to calculate more than just eigenvalues, and gradually, the computation of the eigenvalues of nonnormal matrices and the computation of their pseudospectra will fuse into one subject.

This field will also participate in a broader trend in the scientific computing of the future, the gradual breaking down of the walls between the two steps of discretization (operator → matrix) and solution (matrix → spectrum or pseudospectra). For many problems, the matrix → operator limit is ill-behaved in the spectrum but well-behaved in the pseudospectra. Perhaps pseudospectra will play a role too in breaking down walls between the theorists of functional analysis and the engineers of scientific computing.

ACKNOWLEDGEMENTS

I am grateful to Andrew Spratley for his computations for Section 21 and to Jos van Dorsselaer and Henk van der Vorst for their contributions to the discussion of large-scale Krylov subspace iterations. (That the discussion is still far from complete is my responsibility alone.) In addition, van Dorsselaer provided indispensable assistance on the subject of partial Schur decompositions. For comments on a draft manuscript I thank Spratley and van Dorsselaer and also Mark Embree, Anne Greenbaum, Nick Higham, Richard Lehoucq, Satish Reddy, and Endre Süli. Finally, special thanks go to Anna Aslanyan and Brian Davies for taking a detailed interest in this work, which included the computational verification of some of my numbers, and for pointing out an error in a preliminary draft.

REFERENCES

E. Anderson, *et al.* (1995), *LAPACK Users' Guide, 2nd ed.,* SIAM, Philadelphia.

J. S. Baggett (1994), 'Pseudospectra of an operator of Hille and Phillips', Res. Rep. 94-15, Interdisc. Proj. Ctr. Supercomp., Swiss Federal Institute of Technology, Zurich.

R. Barrett, *et al.* (1994), *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods,* SIAM, Philadelphia.

D. Bau, III and L. N. Trefethen (1997), *Numerical Linear Algebra,* SIAM, Philadelphia.

D. Bayer and P. Diaconis (1992), 'Trailing the dovetail shuffle to its lair', *Ann. Appl. Prob.* **2**, 294–313.

C. Bekas and E. Gallopoulos (1998), 'COBRA: A hybrid method for computing the matrix pseudospectrum', abstract, Copper Mountain Conf. on Iterative Methods.

C. Bekas and E. Gallopoulos (1999), 'COBRA: Parallel parth following for computing the matrix pseudospectrum', manuscript in preparation.

D. Borba, *et al.* (1994), 'The pseudospectrum of the resistive magnetohydrodynamics operator: resolving the resistive Alfvén paradox', *Phys. Plasmas* **1**, 3151–3160.

A. Böttcher (1994), 'Pseudospectra and singular values of large convolution operators', *J. Int. Eqs. and Applics.* **6**, 267–301.

S. Boyd and C. A. Desoer (1985), 'Subharmonic functions and performance bounds on linear time-invariant feedback systems', *IMA J. Math. Control Inf.* **2**, 153–170.

T. Braconnier (1996), Fvpspack: A Fortran and PVM package to compute the field of values and pseudospectra of large matrices, Numer. Anal. Rep. 293, Manchester Ctr. Comp. Maths., Manchester, England.

T. Braconnier (1997), 'Complete iterative method for computing pseudospectra', preprint.

T. Braconnier, F. Chatelin, and J.-C. Dunyach (1995), 'Highly nonnormal eigenvalue problems in the aeronautical industry', *Japan J. Ind. Appl. Math.* **12**, 123–136.

T. Braconnier and N. J. Higham (1996), 'Computing the field of values and pseudospectra using the Lanczos method with continuation', *BIT* **36**, 422–440.

T. Braconnier, A. McCoy, and V. Toumazou (1997), 'Using the field of values for pseudospectra generation', CERFACS, TR/PA/97/28.

M. Brühl (1996), 'A curve tracing algorithm for computing the pseudospectrum', *BIT* **36**, 441–454.

K. M. Butler and B. F. Farrell (1992), 'Three-dimensional optimal perturbations in viscous shear flow', *Phys. Fluids A* **4**, 1637–1650.

C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang (1988), *Spectral Methods in Fluid Dynamics,* Springer, New York.

J. F. Carpraux, J. Erhel, and M. Sadkane (1994), 'Spectral portrait for non-Hermitian large sparse matrices', *Computing* **53**, 301–310.

F. Chaitin-Chatelin and V. Frayssé (1996), *Lectures on Finite Precision Computations,* SIAM, Philadelphia.

F. Chatelin (1983), *Spectral Approximation of Linear Operators,* Academic Press, London.

J. A. Cochran and E. W. Hinds (1974), 'Eigensystems associated with the complex-symmetric kernels of laser theory', *SIAM J. Appl. Math.* **26**, 776–786.

C. Cossu and J. M. Chomaz (1997), 'Global measures of local convective instabilities', *Phys. Rev. Lett.* **78**, 4387–4390.

D. L. Darmofal and P. J. Schmid (1996), 'The importance of eigenvectors for local preconditioners of the Euler equations', *J. Comp. Phys.* **127**, 346–362.

E. R. Davidson (1975), 'The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices', *J. Comp. Phys.* **17**, 87–94.

E. B. Davies (1998), 'Pseudospectra, the harmonic oscillator and complex resonances', to appear.

E. B. Davies (1998b), 'Pseudospectra of differential operators,' to appear.

J. W. Demmel (1987), 'A counterexample for two conjectures about stability', *IEEE Trans. Aut. Control* **AC-32**, 340–342.

P. Diaconis (1996), 'The cutoff phenomenon in finite Markov chains', *Proc. Natl. Acad. Sci. USA* **93**, 1659–1664.

P. Diaconis, R. L. Graham, and J. A. Morrison (1990), 'Asymptotic analysis of a random walk on a hypercube with many dimensions', *Random Struct. and Alg.* **1**, 51–72.

J. M. Donato (1991), *Iterative Methods for Scalar and Coupled Systems of Elliptic Equations,* PhD thesis, Dept. of Math., U. of California, Los Angeles.

J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst (1998), *Numerical Linear Algebra for High-Performance Computers,* SIAM, Philadelphia.

J. L. M. van Dorsselaer (1997), 'Pseudospectra for matrix pencils and stability of equilibria', *BIT* **37**, 833–845.

J. L. M. van Dorsselaer, J. F. B. M. Kraaijevanger, and M. N. Spijker (1993), 'Linear stability analysis in the numerical solution of initial value problems', *Acta Numerica* **1993**, 199–237.

T. A. Driscoll and L. N. Trefethen (1996), 'Pseudospectra for the wave equation with an absorbing boundary', *J. Comp. Appl. Math.* **69**, 125–142.

B. F. Farrell and P. J. Ioannou (1996), 'Generalized stability theory. Part I: autonomous operators and Part II: Nonautonomous Operators', *J. Atmos. Sci.* **53**, 2025–2040 and 2041–2053.

J. Flaherty, C. E. Seyler, and L. N. Trefethen (1999), 'Large-amplitude transient growth in the linear evolution of equatorial spread F in the presence of shear', *J. Geophys. Research*, to appear.

D. R. Fokkema (1996), *Subspace methods for linear, nonlinear, and eigen problems*, PhD thesis, U. of Utrecht, 1996.

D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst (1999), 'Jacobi–Davidson style QR and QZ algorithms for the reduction of matrix pencils', *SIAM J. Sci. Comput.* **20**, 94–125.

B. Fornberg (1996), *A Practical Guide to Pseudospectral Methods,* Cambridge U. Press, Cambridge.

B. Fornberg and D. M. Sloan (1994), 'A review of pseudospectral methods for solving partial differential equations', *Acta Numerica* **1994**, 203–267.

V. Frayssé, L. Giraud and V. Toumazou (1996), 'Parallel computation of spectral portraits on the Meiko CS2', *High-Performance Computing and Networking*, H. Liddell, A. Colbrook, B. Hertzberger and P. Sloot, eds., Springer-Verlag, 312–318, 1067.

V. Frayssé, M. Gueury, F. Nicoud, and V. Toumazou (1996), 'Spectral portraits for matrix pencils', Tech. Rep. TR/PA/96/19, CERFACS.

V. Frayssé and V. Toumazou (1998), 'A note on the normwise perturbation theory for the regular generalized eigenproblem', *Numer. Lin. Alg. Appl.* **5**, 1–10.

R. W. Freund (1992), 'Quasi-kernel polynomials and their use in non-Hermitian matrix iterations', *J. Comput. Appl. Math.* **43**, 135–158.

E. Gallestey (1998), 'Computing spectral value sets using the subharmonicity of the norm of rational matrices', *BIT* **38**, 22–33.

E. Gallestey (1998b), *Theory and Numerics of Spectral Value Sets,* PhD thesis, U. Bremen.

S. K. Godunov (1992), Spectral portraits of matrices and criteria for spectrum dichotomy, *Proc. Third IMACS-GAMM Symp. Comp. Arith. Sci. Comp. (SCAN-91)*, L. Atanassova and J. Herzberger, eds., North-Holland, Amsterdam.

S. K. Godunov (1997), *Modern Aspects of Linear Algebra (Russian),* Nauchnaya Kniga, Novosibirsk.

S. K. Godunov, A. Antonov, O. P. Kiriljuk, and V. I. Kostin (1993), *Guaranteed Accuracy in Numerical Linear Algebra*, Kluwer, 1993.

S. K. Godunov, O. P. Kiriljuk, and V. I. Kostin (1990), Spectral portraits of matrices, Tech. Rep. 3, Inst. of Math., Acad. Sci. USSR, Novosibirsk.

S. K. Godunov and M. Sadkane (1996), 'Elliptic dichotomy of a matrix spectrum', *Lin. Alg. Applics.* **248**, 205–232.

A. Greenbaum (1997), *Iterative Methods for Solving Linear Systems,* SIAM, Philadelphia.

A. Greenbaum and L. N. Trefethen (1993), 'Do the pseudospectra of a matrix determine its behavior?', Tech. Rep. 93-1371, Dept. Comp. Sci., Cornell U.

P. Henrici (1962), 'Bounds for iterates, inverses, spectral variation and field of values of nonnormal matrices', *Numer. Math.* **4**, 24–40.

V. Heuveline, B. Philippe, and M. Sadkane (1997), 'Parallel computation of spectral portrait of large matrices by Davidson type methods', *Numer. Algs.* **16**, 55–75.

D. J. Higham and B. Owren (1996), 'Non-normality effects in a discretised nonlinear reaction-convection-convection-diffusion equation', *J. Comp. Phys.* **124**, 309–323.

D. J. Higham and L. N. Trefethen (1993), 'Stiffness of ODEs', *BIT* **33**, 285–303.

N. J. Higham (1995), 'The Test Matrix Toolbox for Matlab (Version 3.0)', Numer. Anal. Rep. 276, Manchester Ctr. Comp. Maths., Manchester, England; available online at `http://www.maths.man.ac.uk/~higham/`.

N. J. Higham and F. Tisseur (1999), 'A block algorithm for matrix 1-norm estimation, with an application to 1-norm pseudospectra', manuscript.

D. Hinrichsen and B. Kelb (1993), 'Spectral value sets: a graphical tool for robustness analysis', *Systems and Control Letters* **21**, 127–136.

D. Hinrichsen and A. J. Pritchard (1992), 'On spectral variations under bounded real matrix perturbations', *Numer. Math.* **60**, 509–524.

D. Hinrichsen and A. J. Pritchard (1994), 'Stability of uncertain systems', in *Systems and Networks: Mathematical Theory and Applications*, v. I, Akademie–Verlag, Berlin, pp. 159–182.

G. F. Jónsson and L. N. Trefethen (1998), 'A numerical analyst looks at the "cutoff phenomenon" in card shuffling and other Markov chains', in D. F. Griffiths, D. J. Higham, and G. A. Watson, eds., *Numerical Analysis 1997*, Longman, Harlow, Essex, UK.

T. Kato (1976), *Perturbation Theory for Linear Operators,* 2nd ed., Springer-Verlag, New York.

V. Kostin (1991), 'On definition of matrices' spectra', *High Performance Computing II*, M. Durand and F. El Dabaghi, eds., Elsevier.

H. O. Kreiss (1962), 'Über die Stabilitätsdefinition für Differenzengleichungen die partielle Differenzialgleichungen approximieren', *BIT* **2**, 153–181.

H. J. Landau (1975), 'On Szegő's eigenvalue distribution theory and non-Hermitian kernels', *J. d'Analyse Math.* **28**, 335–357.

H. J. Landau (1976), 'Loss in unstable resonators', *J. Opt. Soc. Am.* **66**, 525–529.

H. J. Landau (1977), 'The notion of approximate eigenvalues applied to an integral equation of laser theory', *Quart. Appl. Math.* **35**, 165–172.

L. László (1994), 'An attainable lower bound for the best normal approximation', *SIAM J. Matrix Anal. Applics.* **15**, 1035–1043.

P. Lavallée and M. Sadkane (1997), 'Une méthode stable de bloc-diagonalisation de matrices: Application au calcul de portrait spectral', Tech. Rep. 3141, INRIA, Rennes, 1997.

R. B. Lehoucq, D. C. Sorensen, and C. Yang (1998), *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods,* SIAM, Philadelphia.

S. H. Lui (1997), 'Computation of pseudospectra by continuation', *SIAM J. Sci. Comp.* **18**, 567–573.

A. Lumsdaine and D. Wu (1997), 'Spectra and pseudospectra of waveform relaxation operators', *SIAM J. Sci. Comp* **18**, 286–304.

O. Marques and V. Toumazou (1995), 'Spectral portrait computation by Lanczos method (augmented matrix method)', TR/PA/95/05, CERFACS.

O. Marques and V. Toumazou (1995b), 'Spectral portrait computation by Lanczos method (normal equation method)', TR/PA/95/02, CERFACS.

R. A. McCoy and V. Toumazou (1997), PRECISE User's Guide–Version 1.0, CERFACS, 1997, TR/PA/97/38.

N. M. Nachtigal, L. Reichel, and L. N. Trefethen (1992), 'A hybrid GMRES algorithm for nonsymmetric linear systems', *SIAM J. Matrix Anal. Applics.* **13**, 796–825.

P. J. Olsson and D. S. Henningson (1995), 'Optimal disturbance growth in watertable flow', *Stud. Appl. Math.* **94**, 183–210.

A. Pazy (1983), *Semigroups of Linear Operators and Applications to Partial Differential Equations,* Springer, New York.

R. Plato (1997), 'Resolvent estimates for Abel integral operators and the regularization of associated first kind integral equations', *J. Int. Eqs. Appl.* **9**, 253–278.

S. C. Reddy (1993), 'Pseudospectra of Wiener-Hopf integral operators and constant-coefficient differential operators', *J. Int. Eqs. Appl.* **5**, 369–403.

S. C. Reddy and D. S. Henningson (1993), 'Energy growth in viscous channel flows', *J. Fluid Mech.* **252**, 209–238.

S. C. Reddy, P. J. Schmid, and D. S. Henningson (1993), 'Pseudospectra of the Orr–Sommerfeld operator', *SIAM J. Appl. Math* **53**, 15–47.

S. C. Reddy and L. N. Trefethen (1990), 'Lax-stability of fully discrete spectral methods via stability regions and pseudo-eigenvalues', *Comp. Meth. Appl. Mech. Engr.* **80**, 147–164.

S. C. Reddy and L. N. Trefethen (1994), 'Pseudospectra of the convection-diffusion operator', *SIAM J. Appl. Math.* **54**, 1634–1649.

L. Reichel and L. N. Trefethen (1992), 'Eigenvalues and pseudo-eigenvalues of Toeplitz matrices', *Lin. Alg. Applics.* **162–164**, 153–185.

K. Riedel (1994), 'Generalized epsilon-pseudospectra', *SIAM J. Numer. Anal.* **31**, 1219–1225.

A. Ruhe (1994), 'The rational Krylov algorithm for large nonsymmetric eigenvalues—mapping the resolvent norms (pseudospectrum)', unpublished manuscript.

A. Ruhe (1998), 'Rational Krylov: A practical algorithm for large sparse nonsymmetric matrix pencils', *SIAM J. Sci. Comput.* **19**, 1535–1551.

Y. Saad (1992), *Numerical Methods for Large Eigenvalue Problems,* Manchester U. Press, Manchester, UK.

P. J. Schmid, D. S. Henningson, M. Khorrami, and M. Malik (1993), 'A sensitivity study of hydrodynamic stability operators', *Theor. Comp. Fluid Dyn.* **4**, 227–240.

V. Simoncini and E. Gallopoulos (1998), 'Transfer functions and resolvent norm approximation of large matrices', *Elec. Trans. Numer. Anal.* **7**, 190–201.

G. L. G. Sleijpen and H. A. van der Vorst (1996), 'A Jacobi–Davidson iteration method for linear eigenvalue problems', *SIAM J. Matrix Anal. Appl.* **17**, 401–425.

B. T. Smith, *et al.* (1976), *Matrix Eigensystem Routines—EISPACK Guide,* Springer, Berlin.

K.-C. Toh and L. N. Trefethen (1994), 'Pseudozeros of polynomials and pseudospectra of companion matrices', *Numer. Math.* **68**, 403–425.

K.-C. Toh and L. N. Trefethen (1996), 'Computation of pseudospectra by the Arnoldi iteration', *SIAM J. Sci. Comput.* **17**, 1–15.

K.-C. Toh and L. N. Trefethen (1999), 'The Chebyshev polynomials of a matrix', *SIAM J. Matrix Anal. Appl.*, to appear.

K.-C. Toh and L. N. Trefethen (SIAM J. Matrix Anal. Appl.), 'The Kreiss matrix theorem on a general complex domain', , to appear. 1999b/

V. Toumazou (1966), *Portraits Spectraux de Matrices: Un Outil d'Analyse de la Stabilité*, PhD thesis, U. Raymond Poincaré, Nancy I, CERFACS Tech. Rep. TH/PA/96/46.

A. E. Trefethen, *et al.* (1996), 'MultiMATLAB: MATLAB on multiple processors', Tech. Rep. CTC96TR293, Cornell Theory Center, `http://cs-tr.cs.cornell.edu:80/Dienst/UI/1.0/Display/ncstrl.cornell/TR96-1586`.

A. E. Trefethen, L. N. Trefethen, and P. J. Schmid (1999), 'Spectra and pseudospectra for pipe Poiseuille flow', *Comp. Meth. Appl. Mech. Engr.*, to appear.

L. N. Trefethen (1990), 'Approximation theory and numerical linear algebra', in J. C. Mason and M. G. Cox, eds., *Algorithms for Approximation II*, Chapman and Hall, London.

L. N. Trefethen (1992), *Pseudospectra of matrices*, in D. F. Griffiths and G. A. Watson, eds., Numerical Analysis 1991, Longman Scientific and Technical, Harlow, Essex, UK, pp. 234–266.

L. N. Trefethen (1997), 'Pseudospectra of linear operators', *SIAM Review* **39**, 383–406.

L. N. Trefethen (1999), 'Spectra and pseudospectra: The behavior of non-normal matrices and operators', *The Graduate Student's Guide to Numerical Analysis*, Springer, Berlin, to appear.

L. N. Trefethen and D. Bau, III (1997), *Numerical Linear Algebra,* SIAM, Philadelphia.

L. N. Trefethen, A. E. Trefethen, S. C. Reddy and T. A. Driscoll (1993), 'Hydrodynamic stability without eigenvalues', *Science* **261**, 578–584.

J. M. Varah (1979), 'On the separation of two matrices', *SIAM J. Numer. Anal.* **16**, 216–222.

J. H. Wilkinson (1965), *The Algebraic Eigenvalue Problem,* Clarendon Press, Oxford.