9. Eigenvalues and Pseudospectra

Spectral methods are powerful tools for the computation of eigenvalues of differential and integral operators and their generalizations for strongly nonsymmetric problems, pseudospectra. Indeed, it was Orszag's 1971 computation of the critical Reynolds number $R = 5772.22$ for eigenvalue instability of plane Poiseuille fluid flow, a problem we shall discuss in Chapter 14, that did as much as anything to establish spectral methods as an important tool in scientific computing [Ors71]. Perhaps the reason why spectral methods are so important for eigenvalue computations is that these are applications where high accuracy tends to be crucial.

So far in this book we have seen two examples of eigenvalue calculations. Program 8 (p. 38) solved the harmonic oscillator problem

$$-u_{xx} + x^2 u \;=\; \lambda u, \qquad x \in \mathbb{R}$$

by a Fourier spectral method, taking advantage of the exponential decay of the eigenfunctions to replace the real line $\mathbb{R}$ by the periodic interval $[-L, L]$. Program 15 (p. 66) solved the even simpler problem

$$u_{xx} = \lambda u, \qquad -1 < x < 1, \quad u(\pm 1) = 0$$

by a Chebyshev spectral method that imposed the homogeneous Dirichlet conditions explicitly. In this chapter, we will develop such methods further with the aid of four additional examples. In each case we apply spectral ideas via matrices rather than the FFT, since it is so convenient to take advantage of the standard powerful algorithms for matrix eigenvalue and generalized eigenvalue problems embodied in the MATLAB commands `eig` and `eigs`. Our four examples and the special features they illustrate can be summarized as follows:

Program 21	Mathieu equation	periodic domain
Program 22	Airy equation	generalized eigenvalue problem
Program 23	membrane oscillations	2D domain
Program 24	complex harmonic oscillator	pseudospectra

We begin with the Mathieu equation, an ODE that arises in problems of forced oscillations. (This example has been taken from Weideman and Reddy [WeRe00].) The equation can be written

$$-u_{xx} + 2q\cos(2x)u \;=\; \lambda u, \tag{9.1}$$

where q is a real parameter, and we look for periodic solutions on $[-\pi, \pi]$. For $q = 0$, we have the linear pendulum equation of Program 15, with eigenvalues $\pi^2 n^2/4$ for $n = 1, 2, 3, \ldots$. The scientific interest arises in the behavior of these eigenvalues as q is increased.

To compute eigenvalues of the Mathieu equation by a spectral method, Program 21 discretizes (9.1) in a routine fashion. Translating the equation from the domain $[-\pi, \pi]$ to $[0, 2\pi]$ leaves the eigenvalues unaltered, so our discretization takes the form

$$L_N = -D_N^{(2)} + 2q\,\mathrm{diag}(\cos(2x_1), \ldots, \cos(2x_N)),$$

where $D_N^{(2)}$ is the second-order Fourier differentiation matrix. The computation is straightforward, and with $N = 42$, we get about 13 digits of accuracy. Output 21 presents the plot generated by this program, showing the curves traced by the first 11 eigenvalues as q increases from 0 to 15. Producing this image took about half a second on my workstation. As shown in the figure, it is almost identical to Figure 20.1 on p. 724 of the classic *Handbook of Mathematical Functions* published by Abramowitz and Stegun in the 1960s [AbSt65]. (A few imperfections in the *Handbook* plot can be discerned, for example in the slope of the a_2 curve at $q = 0$.) We do not know how many seconds it took Gertrude Blanch, the author of the chapter on Mathieu functions in the *Handbook*, to produce that figure.

For our second example we turn to another classical problem of applied mathematics, the Airy equation [AbSt65,BeOr78]. Traditionally, the Airy equation is posed on the real line,

$$u_{xx} = xu, \qquad x \in \mathbb{R}. \tag{9.2}$$

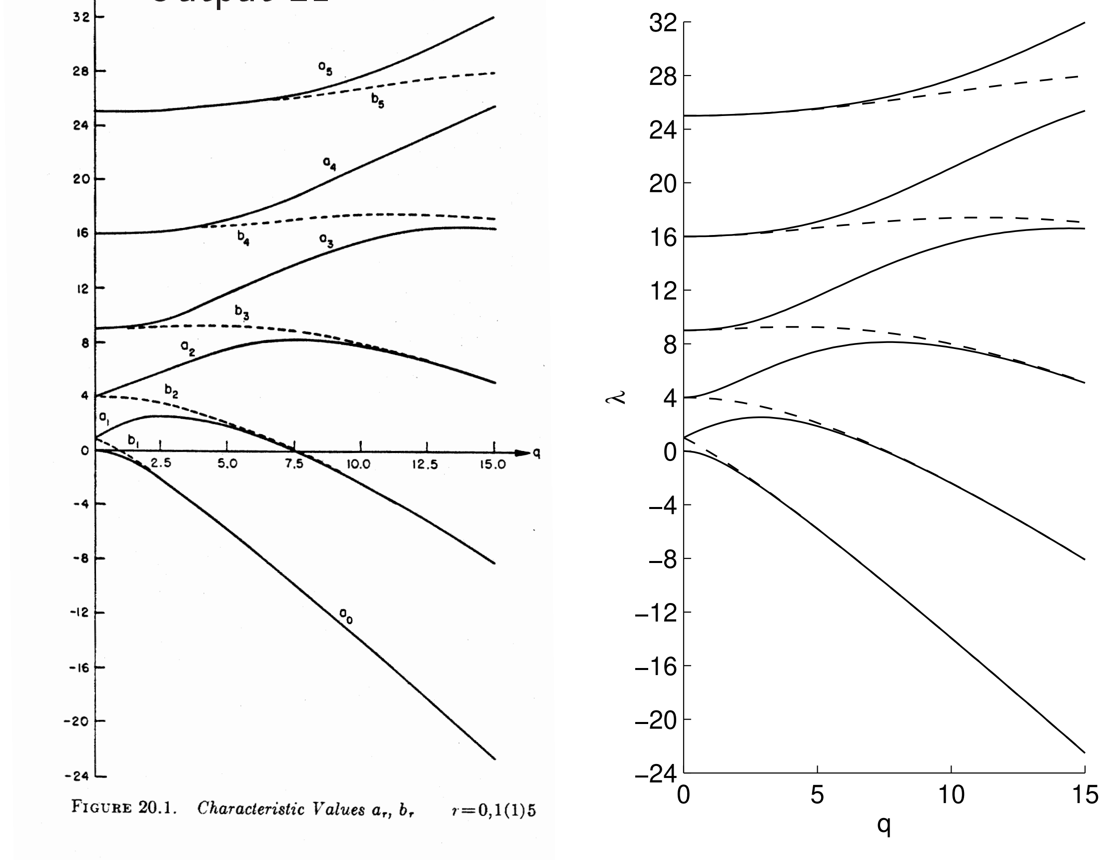
This is the canonical example of an ODE that changes type in different parts of the domain. For $x < 0$, the behavior is oscillatory, while for $x > 0$, we get growing and decaying exponential solutions. Being an ODE of second order, the Airy equation has a two-dimensional linear space of solutions, and the

Program 21

```
% p21.m - eigenvalues of Mathieu operator -u_xx + 2qcos(2x)u
%         (compare p8.m and p. 724 of Abramowitz & Stegun)

  N = 42; h = 2*pi/N; x = h*(1:N);
  D2 = toeplitz([-pi^2/(3*h^2)-1/6 ...
                 -.5*(-1).^(1:N-1)./sin(h*(1:N-1)/2).^2]);
  qq = 0:.2:15; data = [];
  for q = qq;
    e = sort(eig(-D2 + 2*q*diag(cos(2*x))))';
    data = [data; e(1:11)];
  end
  clf, subplot(1,2,1)
  set(gca,'colororder',[0 0 1],'linestyleorder','-|--'), hold on
  plot(qq,data,'linewidth',.8), xlabel q, ylabel \lambda
  axis([0 15 -24 32]), set(gca,'ytick',-24:4:32)
```

Output 21



Output 21: The first 11 eigenvalues of the Mathieu equation (9.1). *Left, Figure* 20.1 *from Abramowitz and Stegun* (1965). *Right, output from Program* 21.

standard basis for this space is the pair of Airy functions $\mathrm{Ai}(x)$, which decays exponentially as $x \to \infty$, and $\mathrm{Bi}(x)$, which grows exponentially.

To solve the Airy equation numerically on the real line by spectral methods is not entirely straightforward, because there are an infinite number of oscillations that decay only algebraically in amplitude; there is an essential singularity at $-\infty$. Instead, in Program 22 we consider a slightly different eigenvalue problem posed on a finite interval,

$$u_{xx} = \lambda x u, \qquad u(\pm 1) = 0, \quad -1 < x < 1. \tag{9.3}$$

This differs from our previous eigenvalue problems in a basic way: rather than being of the form (in linear algebra notation) $Au = \lambda u$, it is a *generalized eigenvalue problem* of the form $Au = \lambda B u$. If B is nonsingular, then a generalized eigenvalue problem can be reduced mathematically to a standard one, $B^{-1}Au = \lambda u$. However, this is not necessarily a good idea in practice, and if B is singular, it is impossible even in principle. Instead, alternative numerical methods are generally used that deal with the generalized problem directly. The most standard is known as the QZ algorithm [GoVa96]. In MATLAB one calls upon the QZ algorithm by writing `eig(A,B)` instead of `eig(A)`. We shall not give details.

To discretize (9.3), we use a standard Chebyshev formulation for the second derivative and a diagonal matrix for the pointwise multiplication:

$$Au = \lambda B u, \qquad A = \tilde{D}_N^2, \quad B = \mathrm{diag}(x_0, \ldots, x_N).$$

The computation is straightforward, and it is evident from Output 22 that we have spectral convergence, with ten or more digits of accuracy in the fifth eigenvector.

Figure 9.1 makes the connection back to Airy functions. In solving (9.3) we have imposed the condition $u(-1) = 0$ and looked for the fifth positive eigenvalue. This is equivalent to computing $\mathrm{Ai}(x)$ on the interval $[-L, L]$, where $-L = -7.944133\ldots$ is the location of the fifth zero of $\mathrm{Ai}(x)$. A rescaling back to $[-1, 1]$ then introduces a power $L^3 = 501.348\ldots$, and that is why our eigenvalue came out as L^3. Actually, what we have just said is not exactly true, for it assumed $\mathrm{Ai}(+L) = 0$, whereas in fact, $\mathrm{Ai}(L) \simeq 5.5 \times 10^{-8}$. What we have computed is actually a spectrally accurate approximation of a linear combination of $\mathrm{Ai}(Lx)$ plus a very small multiple, of order 10^{-14}, of $\mathrm{Bi}(Lx)$. The coefficient 10^{-14} arises because $\mathrm{Bi}(L) \simeq 10^6$.

Our third example is a Laplace eigenvalue problem in two space dimensions,

$$-\Delta u + f(x, y) u \;=\; \lambda u, \quad -1 < x, y < 1, \quad u = 0 \text{ on the boundary.} \tag{9.4}$$

For $f = 0$, we have a familiar problem easily solvable by separation of variables: the eigenfunctions have the form

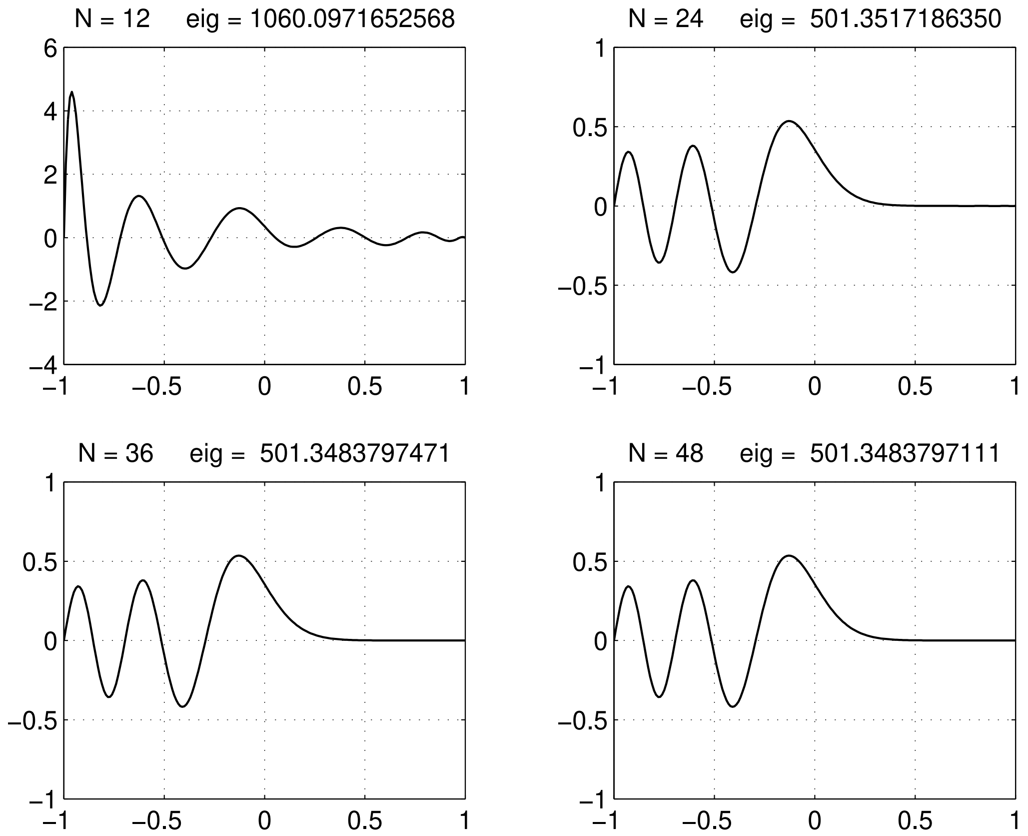
$$\sin(k_x(x+1)) \sin(k_y(y+1)),$$

Program 22

```
% p22.m - 5th eigenvector of Airy equation u_xx = lambda*x*u

  clf
  for N = 12:12:48
    [D,x] = cheb(N); D2 = D^2; D2 = D2(2:N,2:N);
    [V,Lam] = eig(D2,diag(x(2:N)));      % generalized ev problem
    Lam = diag(Lam); ii = find(Lam>0);
    V = V(:,ii); Lam = Lam(ii);
    [foo,ii] = sort(Lam); ii = ii(5); lambda = Lam(ii);
    v = [0;V(:,ii);0]; v = v/v(N/2+1)*airy(0);
    xx = -1:.01:1; vv = polyval(polyfit(x,v,N),xx);
    subplot(2,2,N/12), plot(xx,vv,'linewidth',1), grid on
    title(sprintf('N = %d     eig = %15.10f',N,lambda))
  end
```

Output 22



Output 22: Convergence to the fifth eigenvector of the Airy problem (9.3).

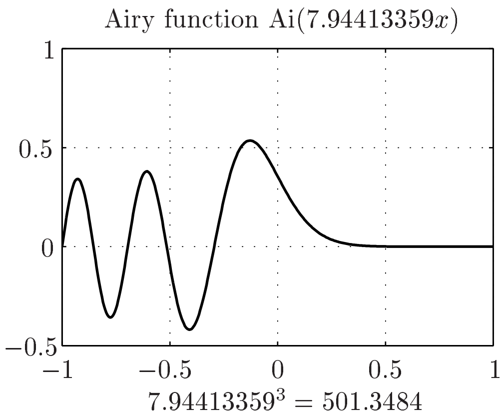


Fig. 9.1. A rescaled solution of the Airy equation, $\mathrm{Ai}(\lambda^{1/3}x)$*. This differs from the solution of Output* 22 *by about* 10^{-8}.

where k_x and k_y are integer multiples of $\pi/2$. This gives eigenvalues

$$\frac{\pi^2}{4}(i^2 + j^2), \qquad i, j = 1, 2, 3, \ldots.$$

Note that most of the eigenvalues are degenerate: whenever $i \neq j$, the eigenvalue has multiplicity two. For $f \neq 0$, on the other hand, (9.4) will have no analytic solution in general and the eigenvalues will not be degenerate. Perturbations will split the double eigenvalues into pairs, a phenomenon familiar to physicists.

To solve (9.4) numerically by a spectral method, we can proceed just as in Program 16 (p. 70). We again set up the discrete Laplacian (7.5) of dimension $(N-1)^2 \times (N-1)^2$ as a sum of two Kronecker products. To this we add a diagonal matrix consisting of the perturbation f evaluated at each of the $(N-1)^2$ points of the grid in the lexicographic ordering described on p. 68. The result is a large matrix whose eigenvalues can be found by standard techniques. In Program 23, this is done by MATLAB's command `eig`. For large enough problems, it would be important to use instead a Krylov subspace iterative method such as the Arnoldi or (if the matrix is symmetric) Lanczos iterations, which are implemented within MATLAB in the alternative code `eigs` (Exercise 9.4).

Output 23a shows results from Program 23 for the unperturbed case, computed by executing the code exactly as printed except with the line `L = L + diag(...)` commented out. Contour plots are given of the first four eigenmodes, with eigenvalues equal to $\pi^2/4$ times 2, 5, 5 and 8. As predicted, two of the eigenmodes are degenerate. As always in cases of degenerate eigenmodes, the choice of eigenvectors here is arbitrary. For essentially arbitrary reasons, the computation picks an eigenmode with a nodal line approximately along

Program 23

```
% p23.m - eigenvalues of perturbed Laplacian on [-1,1]x[-1,1]
%         (compare p16.m)

% Set up tensor product Laplacian and compute 4 eigenmodes:
  N = 16; [D,x] = cheb(N); y = x;
  [xx,yy] = meshgrid(x(2:N),y(2:N)); xx = xx(:); yy = yy(:);
  D2 = D^2; D2 = D2(2:N,2:N); I = eye(N-1);
  L = -kron(I,D2) - kron(D2,I);                    %  Laplacian
  L = L + diag(exp(20*(yy-xx-1)));                 %  + perturbation
  [V,D] = eig(L); D = diag(D);
  [D,ii] = sort(D); ii = ii(1:4); V = V(:,ii);

% Reshape them to 2D grid, interpolate to finer grid, and plot:
  [xx,yy] = meshgrid(x,y);
  fine = -1:.02:1; [xxx,yyy] = meshgrid(fine,fine);
  uu = zeros(N+1,N+1);
  [ay,ax] = meshgrid([.56 .04],[.1 .5]); clf
  for i = 1:4
    uu(2:N,2:N) = reshape(V(:,i),N-1,N-1);
    uu = uu/norm(uu(:),inf);
    uuu = interp2(xx,yy,uu,xxx,yyy,'cubic');
    subplot('position',[ax(i) ay(i) .38 .38])
    contour(fine,fine,uuu,-.9:.2:.9)
    colormap([0 0 0]), axis square
    title(['eig = ' num2str(D(i)/(pi^2/4),'%18.12f') '\pi^2/4'])
  end
```

a diagonal; it then computes a second eigenmode linearly independent of the first (though not orthogonal to it), with a nodal line approximately on the opposite diagonal. An equally valid pair of eigenmodes in this degenerate case would have had nodal lines along the x and y axes.

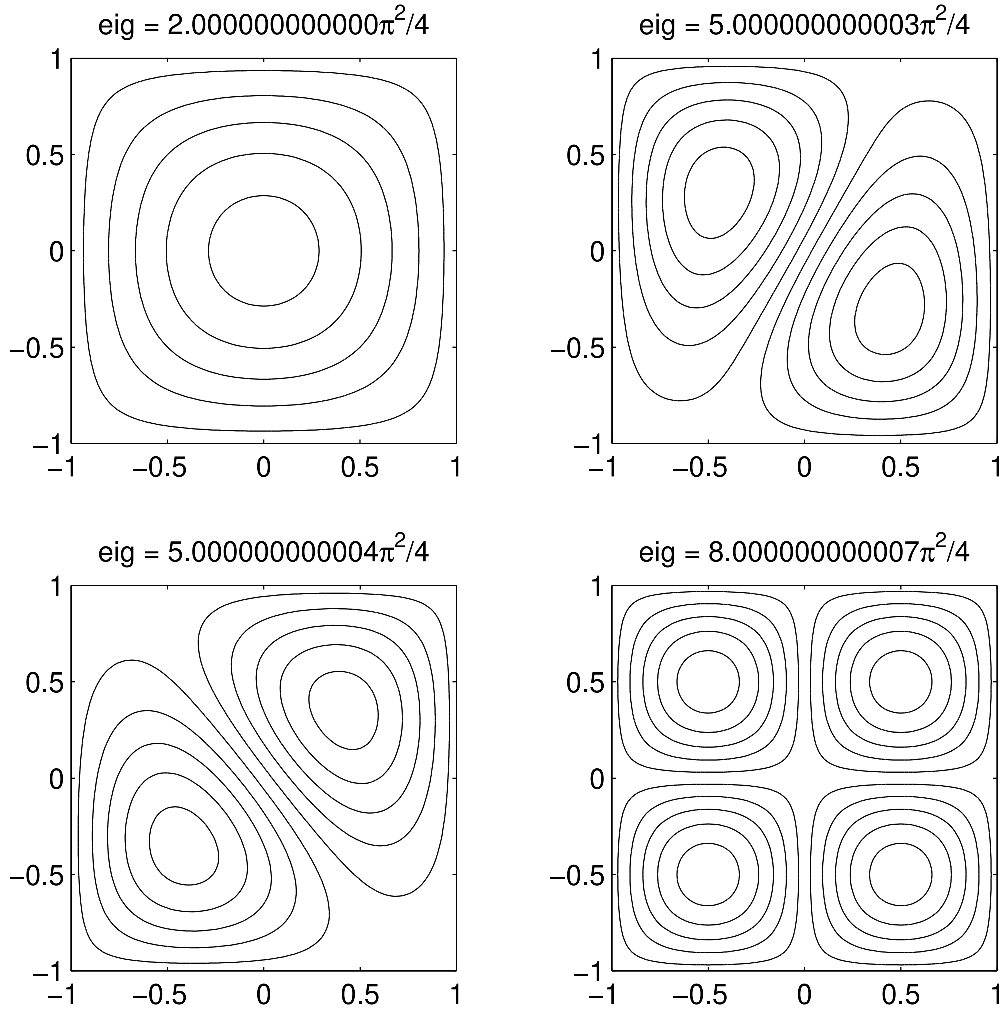
A remarkable feature of Output 23a is that although the grid is only of size 16×16, the eigenvalues are computed to 12-digit accuracy. This reflects the fact that one or two oscillations of a sine wave can be approximated to better than 12-digit precision by a polynomial of degree 16 (Exercise 9.1).

Output 23b presents the same plot with the perturbation in (9.4) included, with

$$f(x,y) = \exp(20(y-x-1)).$$

This perturbation has a very special form. It is nearly zero outside the upper-left triangular region, one-eighth of the total domain, defined by $y - x \geq 1$.

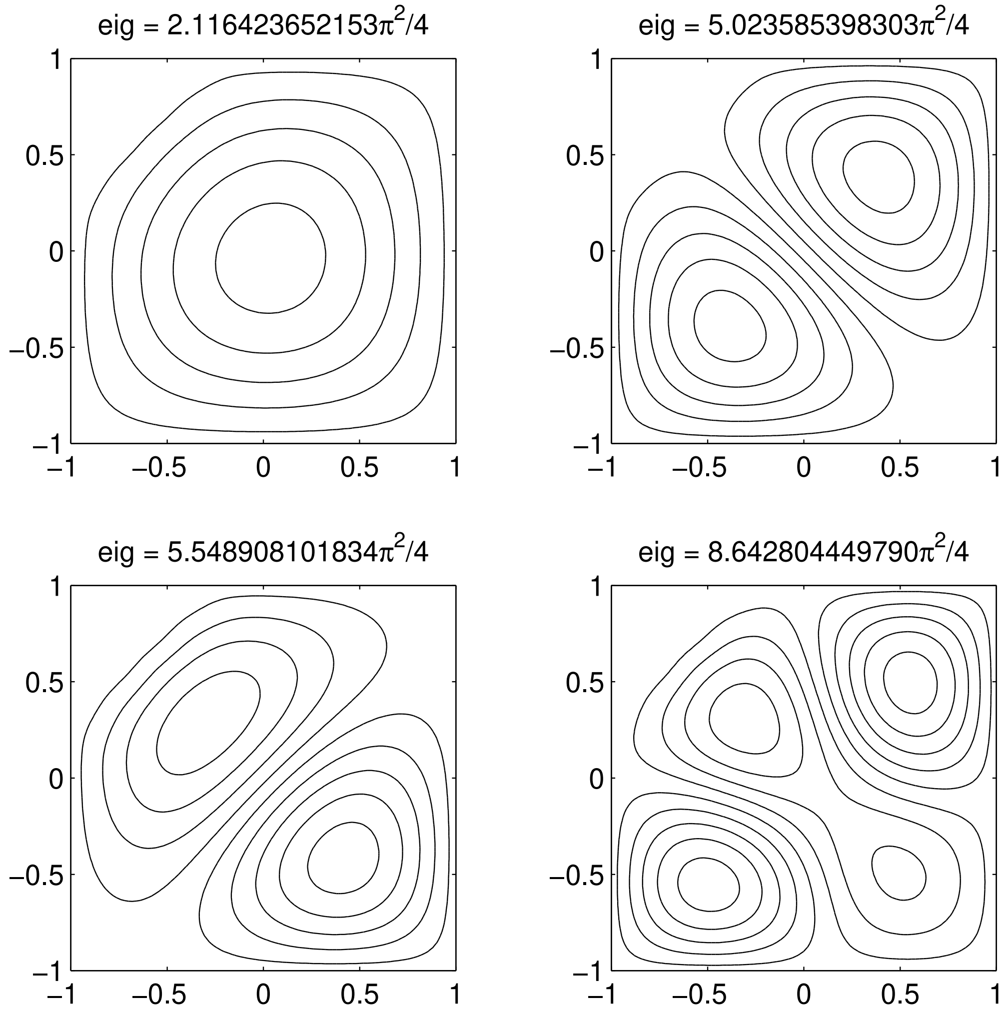
Output 23a



Output 23a: First four eigenmodes of the Laplace problem (9.4) *with* $f(x,y) = 0$. *These plots were produced by running Program* 23 *with the "+ perturbation" line commented out.*

Within that region, however, it is very large, achieving values as great as 4.8×10^8. Thus this perturbation is not small at all in amplitude, though it is limited in extent. It is analogous to the "barrier functions" utilized in the field of optimization of functions with constraints. The effect on the eigenmodes is clear. In Output 23b we see that all four eigenmodes avoid the upper-left corner; the values there are very close to zero. It is approximately as if we had solved the eigenvalue problem on the unit square with a corner snipped off. All four eigenvalues have increased, as they must, and the second and third eigenvalues are no longer degenerate. What we find instead is that mode 3,

Output 23b



Output 23b: First four eigenmodes of the perturbed Laplace problem (9.4) *with* $f(x, y) = \exp(20(y-x-1))$. *These plots were produced by running Program* 23 *as written.*

which had low amplitude in the barrier region, has changed a little, whereas mode 2, which had higher amplitude there, has changed quite a lot. These computed eigenvalues, by the way, are not spectrally accurate; the function f varies too fast to be well resolved on this grid. Experiments with various values of N suggest they are accurate to about three or four digits.

All of our examples of eigenvalue problems so far have involved self-adjoint operators, whose eigenvalues are real and whose eigenvectors can be taken to be orthogonal. Our spectral discretizations are not in fact symmetric matrices (they would be, if we used certain Galerkin rather than collocation methods), but they are reasonably close in the sense of having eigenvectors reasonably close to orthogonal so long as the corresponding eigenvalues are distinct. In general, a matrix with a complete set of orthogonal eigenvectors is said to be *normal*. Normal and nearly normal matrices are the ones whose eigenvalue problems are unproblematic, relatively easy both to solve and to interpret physically.

In a certain minority of applications, however, one encounters matrices or operators that are very far from normal in the sense that the eigenvectors, if a complete set exists, are very far from orthogonal—they form an ill-conditioned basis of the vector space under study. In highly non-normal cases, it may be informative to compute pseudospectra* rather than spectra [Tre97,TTRD93]. Suppose that a square matrix A is given and $\|\cdot\|$ is a physically relevant norm. For each $\epsilon > 0$, the *ϵ-pseudospectrum* of A is the subset of the complex plane

$$\Lambda_\epsilon(A) \;=\; \{z \in \mathbb{C}:\; \|(zI - A)^{-1}\| \ge \epsilon^{-1}\}. \qquad (9.5)$$

(We use the convention $\|(zI - A)^{-1}\| = \infty$ if z is an eigenvalue of A.) Alternatively, $\Lambda_\epsilon(A)$ can be characterized by eigenvalues of perturbed matrices:

$$\Lambda_\epsilon(A) \;=\; \{z \in \mathbb{C}:\; z \text{ is an eigenvalue of } A + E \text{ for some } E \text{ with } \|E\| \le \epsilon\}. \qquad (9.6)$$

If $\|\cdot\|$ is the 2-norm, as is convenient and physically appropriate in most applications (sometimes after a diagonal similarity transformation to get the scaling right), then a further equivalence is

$$\Lambda_\epsilon(A) = \{z \in \mathbb{C}:\; \sigma_{\min}(zI - A) \le \epsilon\}, \qquad (9.7)$$

where $\sigma_{\min}$ denotes the minimum singular value.

Pseudospectra can be computed by spectral methods very effectively, and our final example of this chapter illustrates this. The example returns to the harmonic oscillator (4.6), except that a complex coefficient c is now put in front of the quadratic term. We define our linear operator L by

$$Lu \;=\; -u_{xx} + cx^2 u, \qquad x \in \mathbb{R}. \qquad (9.8)$$

The eigenvalues and eigenvectors for this problem are readily determined analytically: they are $\sqrt{c}\,(2k+1)$ and $\exp(-c^{1/2}x^2/2)H_k(c^{1/4}x)$ for $k = 0, 1, 2, \ldots$, where H_k is the kth Hermite polynomial [Exn83]. However, as E. B. Davies first noted [Dav99], the eigenmodes are exponentially far from orthogonal. Output 24 shows pseudospectra for (9.8) with $c = 1 + 3i$ computed in a

*Pseudospectra (plural of pseudospectrum) are sets in the complex plane; pseudospectral methods are spectral methods based on collocation, i.e., pointwise evaluations rather than integrals. There is no connection—except that pseudospectral methods are very good at computing pseudospectra!

straightforward fashion based on (9.7). We discretize L spectrally, evaluate $\sigma_{\min}(zI - L)$ on a grid of points z_{ij}, then send the results to a contour plotter.

For the one and only time in this book, the plot printed as Output 24 is not exactly what would be produced by the corresponding program as listed. Program 24 evaluates $\sigma_{\min}(zI - L)$ on a relatively coarse 26×21 grid; after 546 complex singular value decompositions, a relatively crude approximation to Output 24 is produced. For the published figure, we made the grid four times finer in each direction by replacing `0:2:50` by `0:.5:50` and `0:2:40` by `0:.5:40`. This slowed down the computation by a factor of 16. (As it happens, alternative algorithms can be used to speed up this calculation of pseudospectra and get approximately that factor of 16 back again; see [Tre99].)

One can infer from Output 24 that although the eigenvalues of the complex harmonic oscillator are regularly spaced numbers along a ray in the complex plane, all but the first few of them would be of doubtful physical significance in a physical problem described by this operator. Indeed, the resolvent norm appears to grow exponentially as $|z| \to \infty$ along any ray with argument between 0 and $\arg c$, so that every value of z sufficiently far out in this infinite sector is an ϵ-pseudoeigenvalue for an exponentially small value of ϵ.

We shall see three further examples of eigenvalue calculations later in the book. We summarize the eigenvalue examples ahead by continuing the table displayed at the beginning of this chapter:

Program 28	circular membrane	polar coordinates
Program 39	square plate	clamped boundary conditions
Program 40	Orr–Sommerfeld operator	complex arithmetic

Summary of this chapter. Spectral discretization can turn eigenvalue and pseudospectra problems for ODEs and PDEs into the corresponding problems for matrices. If the matrix dimension is large, it may be best to solve these by Krylov subspace methods such as the Lanczos or Arnoldi iterations.

Exercises

9.1. Modify Program 23 so that it produces a plot on a log scale of the error in the computed lowest eigenvalue represented in the first panel of Output 23a as a function of N. Now let $\tau > 0$ be fixed and let $E_N = \inf_p \|p(x) - \sin(\tau x)\|_\infty$, where $\|f\|_\infty = \sup_{x\in[-1,1]} |f(x)|$, denote the error in degree N minimax polynomial approximation to $\sin(\tau x)$ on $[-1, 1]$. It is known (see equation (6.77) of [Mei67]) that for even N, as $N \to \infty$, $E_N \sim 2^{-N}\tau^{N+1}/(N + 1)!$. Explain which value of τ should be taken for this result to be used to provide an order of magnitude estimate of the results in the plot. How close is the estimate to the data? (Compare Exercise 5.3.)

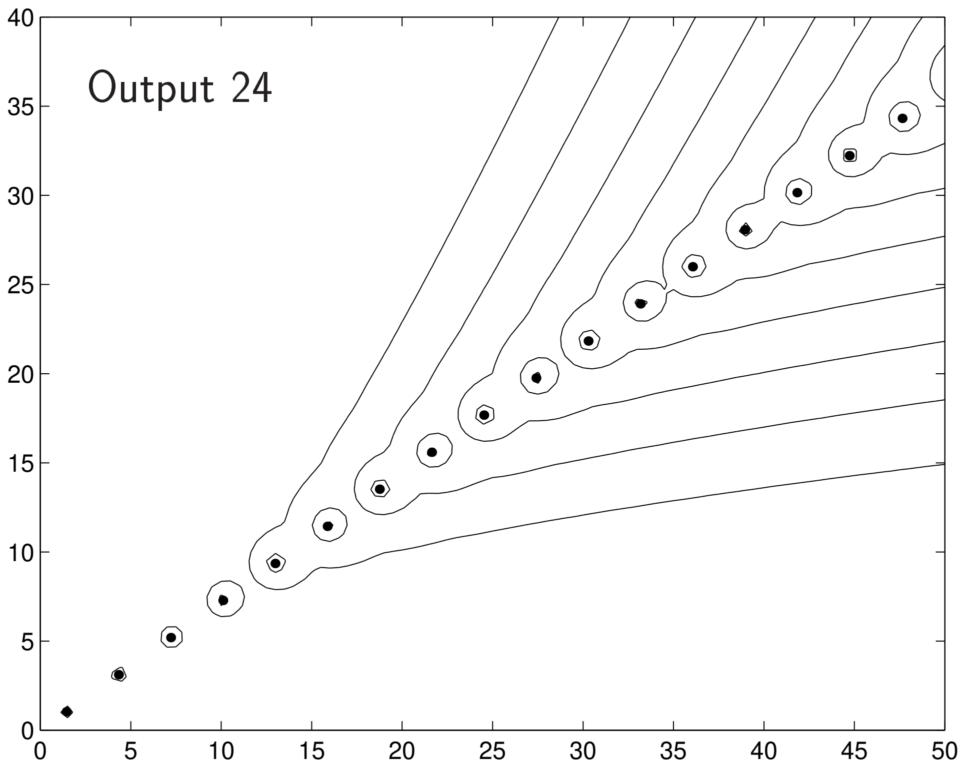
9.2. $A \otimes (B \otimes C) = (A \otimes B) \otimes C$: true or false?

Program 24

```
% p24.m - pseudospectra of Davies' complex harmonic oscillator
%         (For finer, slower plot, change 0:2 to 0:.5.)

% Eigenvalues:
  N = 70; [D,x] = cheb(N); x = x(2:N);
  L = 6; x = L*x; D = D/L;                    % rescale to [-L,L]
  A = -D^2; A = A(2:N,2:N) + (1+3i)*diag(x.^2);
  clf, plot(eig(A),'.','markersize',14)
  axis([0 50 0 40]), drawnow, hold on

% Pseudospectra:
  x = 0:2:50; y = 0:2:40; [xx,yy] = meshgrid(x,y); zz = xx+1i*yy;
  I = eye(N-1); sigmin = zeros(length(y),length(x));
  h = waitbar(0,'please wait...');
  for j = 1:length(x), waitbar(j/length(x))
    for i = 1:length(y), sigmin(i,j) = min(svd(zz(i,j)*I-A)); end
  end, close(h)
  contour(x,y,sigmin,10.^(-4:.5:-.5)), colormap([0 0 0])
```



Ouput 24: Eigenvalues and ϵ-pseudospectra in $\mathbb{C}$ of the complex harmonic oscillator (9.8), $c = 1 + 3i$, $\epsilon = 10^{-0.5}, 10^{-1}, 10^{-1.5}, \ldots, 10^{-4}$.

9.3. Modify Program 23 so that it finds the lowest eigenvalue of the Laplacian on the cube $[-1,1]^3$ rather than the square $[-1,1]^2$. For $N = 6$ and 8, how big is the matrix you are working with, how accurate are the results, and how long does the computation take? Estimate what the answers would be for $N = 12$.

9.4. In continuation of Exercise 9.3, you can solve the problem with $N = 12$ if you use MATLAB'S iterative eigenvalue solver `eigs` rather than its "direct" solver `eig`. Modify your code further to use `eigs`, and be sure that `eigs` is given a sparse matrix to work with (putting `speye` instead of `eye` in your code will ensure this). With $N = 12$, how long does the computation take, and how accurate are the results?

9.5. Consider a circular membrane of radius 1 that vibrates according to the second-order wave equation $y_{tt} = r^{-1}(ry_r)_r + r^{-2}y_{\theta\theta}$, $y(1,t) = 0$, written in polar coordinates. Separating variables leads to consideration of solutions $y(r,\theta,t) = u(r)e^{im\theta}e^{i\omega t}$, with $u(r)$ satisfying $r^{-1}(ru_r)_r + (\omega^2 - r^{-2}m^2)u = 0$, $u_r(0) = 0$, $u(1) = 0$. This is a second-order, linear ODE boundary value problem with homogeneous boundary conditions, so one solution is $u(r) = 0$. Nonzero solutions will only occur for eigenvalues ω of the equation

$$r^{-1}(ru_r)_r - r^{-2}m^2u = -\omega^2u, \qquad u_r(0) = u(1) = 0. \tag{9.9}$$

This is a form of *Bessel's equation*, and the solutions are Bessel functions $J_m(\omega r)$, where ω has the property $J_m(\omega) = 0$. Write a MATLAB program based on a spectral method that, for given m, constructs a matrix whose smaller eigenvalues approximate the smaller eigenvalues of (9.9). List the approximations to the first six eigenvalues ω produced by your program for $m = 0, 1$ and $N = 5, 10, 15, 20$.

9.6. In continuation of Exercise 9.5, the first two eigenvalues for $m = 1$ differ nearly, but not quite, by a factor of 2. Suppose, with musical harmony in mind, we wish to design a membrane with radius-dependent physical properties such that these two eigenvalues have ratio exactly 2. Consider the modified BVP eigenvalue problem

$$r^{-1}(p(r)ru_r)_r - r^{-2}m^2u = -\omega^2u, \qquad u_r(0) = u(1) = 0$$

where $p(r) = 1 + \alpha\sin^2(\pi r)$ for some real number α. Produce a plot that shows the first eigenvalue and $\frac{1}{2}$ times the second eigenvalue as functions of α. For what value of α do the two curves intersect? By solving an appropriate nonlinear equation, determine this critical value of α to at least six digits. Can you explain why a correction of the form $\alpha p(r)$ modifies the ratio of the eigenvalues in the direction required?

9.7. Exercise 6.8 (p. 59) considered powers of the Chebyshev differentiation matrix D_N. For $N = 20$, produce a plot of the eigenvalues and ϵ-pseudospectra of D_N for $\epsilon = 10^{-2}, 10^{-3}, \ldots, 10^{-16}$. Comment on how this plot relates to the results of that exercise.