6. Chebyshev Differentiation Matrices

In the last chapter we discussed why grid points must cluster at boundaries for spectral methods based on polynomials. In particular, we introduced the Chebyshev points,

$$x_j = \cos(j\pi/N), \qquad j = 0, 1, \ldots, N, \tag{6.1}$$

which cluster as required. In this chapter we shall use these points to construct Chebyshev differentiation matrices and apply these matrices to differentiate a few functions. The same set of points will continue to be the basis of many of our computations throughout the rest of the book.

Our scheme is as follows. Given a grid function v defined on the Chebyshev points, we obtain a discrete derivative w in two steps:

- *Let p be the unique polynomial of degree $\leq N$ with $p(x_j) = v_j$, $0 \leq j \leq N$.*
- *Set $w_j = p'(x_j)$.*

This operation is linear, so it can be represented by multiplication by an $(N+1) \times (N+1)$ matrix, which we shall denote by D_N:

$$w = D_N v.$$

Here N is an arbitrary positive integer, even or odd. The restriction to even N in this book (p. 18) applies to Fourier, not Chebyshev spectral methods.

To get a feel for the interpolation process, we take a look at $N = 1$ and $N = 2$ before proceeding to the general case.

Consider first $N = 1$. The interpolation points are $x_0 = 1$ and $x_1 = -1$, and the interpolating polynomial through data v_0 and v_1, written in Lagrange form, is

$$p(x) = \tfrac{1}{2}(1+x)v_0 + \tfrac{1}{2}(1-x)v_1.$$

Taking the derivative gives

$$p'(x) = \tfrac{1}{2}v_0 - \tfrac{1}{2}v_1.$$

This formula implies that D_1 is the 2×2 matrix whose first column contains constant entries $1/2$ and whose second column contains constant entries $-1/2$:

$$D_1 = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2}. \end{pmatrix}$$

Now consider $N = 2$. The interpolation points are $x_0 = 1$, $x_1 = 0$, and $x_2 = -1$, and the interpolant is the quadratic

$$p(x) \;=\; \tfrac{1}{2}x(1+x)v_0 + (1+x)(1-x)v_1 + \tfrac{1}{2}x(x-1)v_2.$$

The derivative is now a linear polynomial,

$$p'(x) = (x+\tfrac{1}{2})v_0 - 2xv_1 + (x-\tfrac{1}{2})v_2.$$

The differentiation matrix D_2 is the 3×3 matrix whose jth column is obtained by sampling the jth term of this expression at $x = 1$, 0, and -1:

$$D_2 = \begin{pmatrix} \frac{3}{2} & -2 & \frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 2 & -\frac{3}{2} \end{pmatrix}. \tag{6.2}$$

It is no coincidence that the middle row of this matrix contains the coefficients for a centered 3-point finite difference approximation to a derivative, and the other rows contain the coefficients for one-sided approximations such as the one that drives the second-order Adams–Bashforth formula for the numerical solution of ODEs [For88]. The rows of higher-order spectral differentiation matrices can also be viewed as vectors of coefficients of finite difference formulas, but these will be based on uneven grids and thus no longer familiar from standard applications.

We now give formulas for the entries of D_N for arbitrary N. These were first published perhaps in [GHO84], and are derived in Exercises 6.1 and 6.2. Analogous formulas for general sets $\{x_j\}$ rather than just Chebyshev points are stated in Exercise 6.1.

Theorem 7 Chebyshev differentiation matrix.

For each $N \geq 1$, let the rows and columns of the $(N+1) \times (N+1)$ Chebyshev spectral differentiation matrix D_N be indexed from 0 to N. The entries of this matrix are

$$(D_N)_{00} = \frac{2N^2+1}{6}, \qquad (D_N)_{NN} = -\frac{2N^2+1}{6}, \tag{6.3}$$

$$(D_N)_{jj} = \frac{-x_j}{2(1-x_j^2)}, \qquad j = 1, \ldots, N-1, \tag{6.4}$$

$$(D_N)_{ij} = \frac{c_i}{c_j} \frac{(-1)^{i+j}}{(x_i - x_j)}, \qquad i \neq j, \quad i, j = 1, \ldots, N-1, \tag{6.5}$$

where

$$c_i = \begin{cases} 2 & i = 0 \text{ or } N, \\ 1 & \text{otherwise.} \end{cases}$$

A picture makes the pattern clearer:

$$D_N = \left[\begin{array}{c|c|c}
\dfrac{2N^2+1}{6} & 2\dfrac{(-1)^j}{1-x_j} & \dfrac{1}{2}(-1)^N \\
\hline
-\dfrac{1}{2}\dfrac{(-1)^i}{1-x_i} & \begin{array}{ccc} & & \dfrac{(-1)^{i+j}}{x_i - x_j} \\ & \dfrac{-x_j}{2(1-x_j^2)} & \\ \dfrac{(-1)^{i+j}}{x_i - x_j} & & \end{array} & \dfrac{1}{2}\dfrac{(-1)^{N+i}}{1+x_i} \\
\hline
-\dfrac{1}{2}(-1)^N & -2\dfrac{(-1)^{N+j}}{1+x_j} & -\dfrac{2N^2+1}{6}
\end{array}\right]$$

The jth column of D_N contains the derivative of the degree N polynomial interpolant $p_j(x)$ to the delta function supported at x_j, sampled at the grid

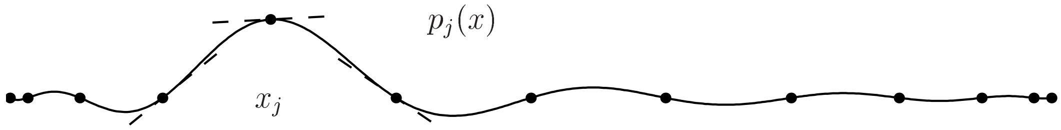


Fig. 6.1. Degree 12 polynomial interpolant $p(x)$ to the delta function supported at x_8 on the 13-point Chebyshev grid with $N = 12$. The slopes indicated by the dashed lines, from right to left, are the entries $(D_{12})_{7,8}$, $(D_{12})_{8,8}$, and $(D_{12})_{9,8}$ of the 13×13 spectral differentiation matrix D_{12}.

points $\{x_i\}$. Three such sampled values are suggested by the dashed lines in Figure 6.1.

Throughout this text, we take advantage of MATLAB's high-level commands for such operations as polynomial interpolation, matrix inversion, and FFT. For clarity of exposition, as explained in the "Note on the MATLAB Programs" at the beginning of the book, our style is to make our programs short and self-contained. However, there will be one major exception to this rule, one MATLAB function that we will define and then call repeatedly whenever we need Chebyshev grids and differentiation matrices. The function is called `cheb`, and it returns a vector x and a matrix D.

cheb.m

```
% CHEB  compute D = differentiation matrix, x = Chebyshev grid

  function [D,x] = cheb(N)
  if N==0, D=0; x=1; return, end
  x = cos(pi*(0:N)/N)';
  c = [2; ones(N-1,1); 2].*(-1).^(0:N)';
  X = repmat(x,1,N+1);
  dX = X-X';
  D  = (c*(1./c)')./(dX+(eye(N+1)));      % off-diagonal entries
  D  = D - diag(sum(D'));                 % diagonal entries
```

Note that this program does not compute D_N exactly by formulas (6.3)–(6.5). It utilizes (6.5) for the off-diagonal entries but then obtains the diagonal entries (6.3)–(6.4) from the identity

$$(D_N)_{ii} = -\sum_{\substack{j=0 \\ j\neq i}}^{N} (D_N)_{ij}. \tag{6.6}$$

This is marginally simpler to program, and it produces a matrix with better stability properties in the presence of rounding errors [BaBe00,BCM94]. Equation (6.6) can be derived by noting that the interpolant to $(1,1,\ldots,1)^T$ is the constant function $p(x)=1$, and since $p'(x)=0$ for all x, D_N must map $(1,1,\ldots,1)^T$ to the zero vector.

Here are the first five Chebyshev differentiation matrices as computed by `cheb`. Note that they are dense, with little apparent structure apart from the antisymmetry condition $(D_N)_{ij} = -(D_N)_{N-i,N-j}$.

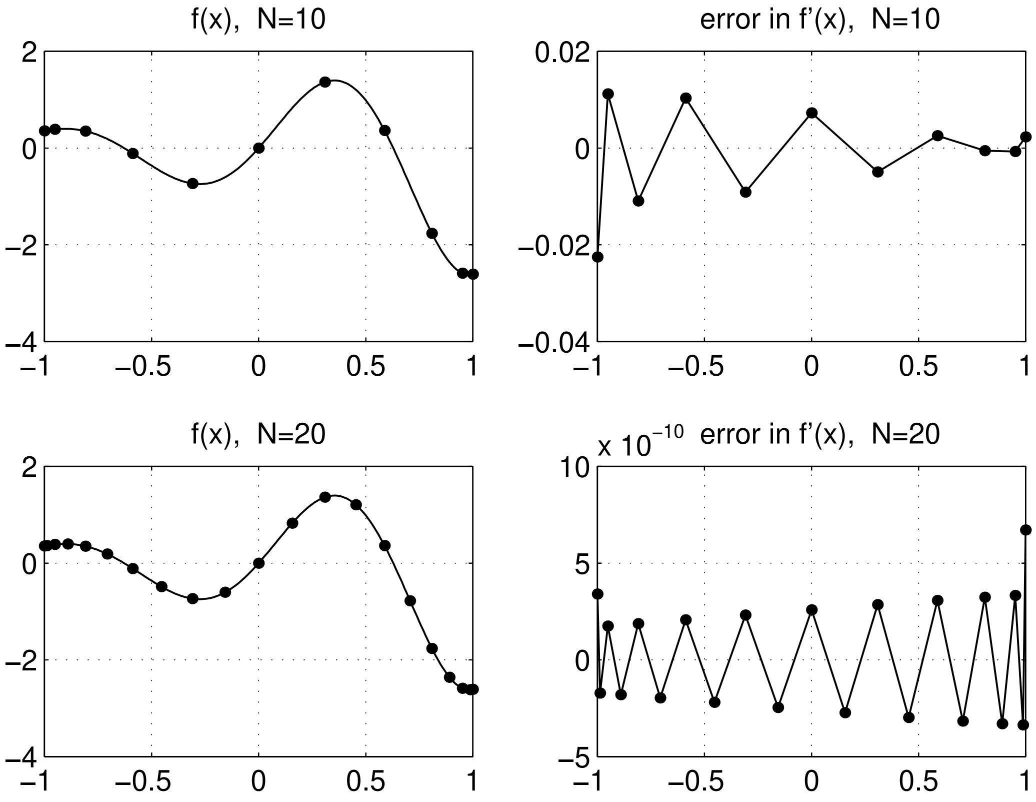
```
>> cheb(1)
    0.5000   -0.5000
    0.5000   -0.5000

>> cheb(2)
    1.5000   -2.0000    0.5000
    0.5000   -0.0000   -0.5000
   -0.5000    2.0000   -1.5000

>> cheb(3)
    3.1667   -4.0000    1.3333   -0.5000
    1.0000   -0.3333   -1.0000    0.3333
   -0.3333    1.0000    0.3333   -1.0000
    0.5000   -1.3333    4.0000   -3.1667

>> cheb(4)
    5.5000   -6.8284    2.0000   -1.1716    0.5000
    1.7071   -0.7071   -1.4142    0.7071   -0.2929
   -0.5000    1.4142   -0.0000   -1.4142    0.5000
    0.2929   -0.7071    1.4142    0.7071   -1.7071
   -0.5000    1.1716   -2.0000    6.8284   -5.5000

>> cheb(5)
    8.5000  -10.4721    2.8944   -1.5279    1.1056   -0.5000
    2.6180   -1.1708   -2.0000    0.8944   -0.6180    0.2764
   -0.7236    2.0000   -0.1708   -1.6180    0.8944   -0.3820
    0.3820   -0.8944    1.6180    0.1708   -2.0000    0.7236
   -0.2764    0.6180   -0.8944    2.0000    1.1708   -2.6180
    0.5000   -1.1056    1.5279   -2.8944   10.4721   -8.5000
```

Program 11 illustrates how D_N can be used to differentiate the smooth, nonperiodic function $u(x) = e^x \sin(5x)$ on grids with $N = 10$ and $N = 20$. The output shows a graph of $u(x)$ alongside a plot of the error in $u'(x)$. With $N = 20$, we get 9-digit accuracy.

Program 11

```
% p11.m - Chebyshev differentation of a smooth function

  xx = -1:.01:1; uu = exp(xx).*sin(5*xx); clf
  for N = [10 20]
    [D,x] = cheb(N); u = exp(x).*sin(5*x);
      subplot('position',[.15 .66-.4*(N==20) .31 .28])
      plot(x,u,'.','markersize',14), grid on
      line(xx,uu,'linewidth',.8)
      title(['u(x),  N=' int2str(N)])
    error = D*u - exp(x).*(sin(5*x)+5*cos(5*x));
      subplot('position',[.55 .66-.4*(N==20) .31 .28])
      plot(x,error,'.','markersize',14), grid on
      line(x,error,'linewidth',.8)
      title(['  error in u''(x),  N=' int2str(N)])
  end
```

Output 11



Output 11: Chebyshev differentiation of $u(x) = e^x \sin(5x)$. Note the vertical scales.

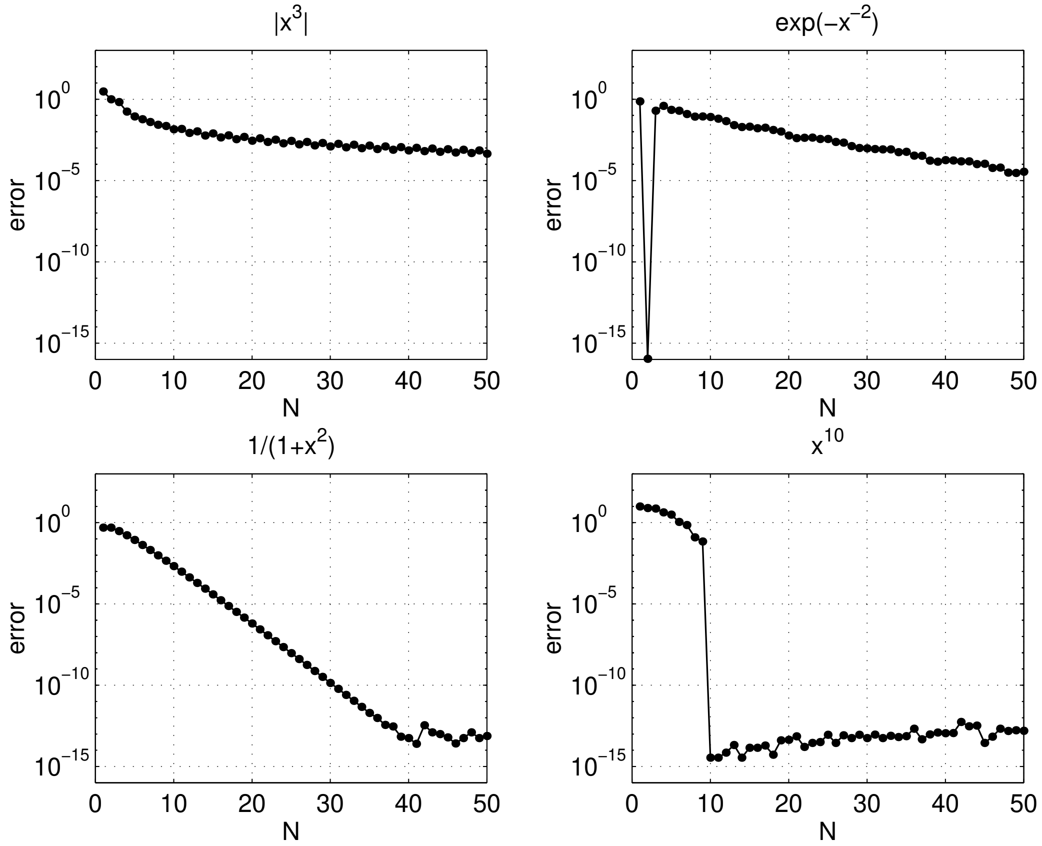
Program 12

```
% p12.m - accuracy of Chebyshev spectral differentiation
%         (compare p7.m)

% Compute derivatives for various values of N:
  Nmax = 50; E = zeros(3,Nmax);
  for N = 1:Nmax;
    [D,x] = cheb(N);
    v = abs(x).^3; vprime = 3*x.*abs(x);     % 3rd deriv in BV
    E(1,N) = norm(D*v-vprime,inf);
    v = exp(-x.^(-2)); vprime = 2.*v./x.^3;  % C-infinity
    E(2,N) = norm(D*v-vprime,inf);
    v = 1./(1+x.^2); vprime = -2*x.*v.^2;    % analytic in [-1,1]
    E(3,N) = norm(D*v-vprime,inf);
    v = x.^10; vprime = 10*x.^9;             % polynomial
    E(4,N) = norm(D*v-vprime,inf);
  end

% Plot results:
  titles = {'|x^3|','exp(-x^{-2})','1/(1+x^2)','x^{10}'}; clf
  for iplot = 1:4
    subplot(2,2,iplot)
    semilogy(1:Nmax,E(iplot,:),'.','markersize',12)
    line(1:Nmax,E(iplot,:),'linewidth',.8)
    axis([0 Nmax 1e-16 1e3]), grid on
    set(gca,'xtick',0:10:Nmax,'ytick',(10).^(-15:5:0))
    xlabel N, ylabel error, title(titles(iplot))
  end
```

Program 12, the Chebyshev analogue of Program 7, illustrates spectral accuracy more systematically. Four functions are spectrally differentiated: $|x^3|$, $\exp(-x^{-2})$, $1/(1+x^2)$, and x^{10}. The first has a third derivative of bounded variation, the second is smooth but not analytic, the third is analytic in a neighborhood of $[-1,1]$, and the fourth is a polynomial, the analogue for Chebyshev spectral methods of a band-limited function for Fourier spectral methods.

Summary of this chapter. The entries of the Chebyshev differentiation matrix D_N can be computed by explicit formulas, which can be conveniently collected in an eight-line MATLAB function. More general explicit formulas can be used to construct the differentiation matrix for an arbitrarily prescribed set of distinct points $\{x_j\}$.

Output 12



Output 12: Accuracy of the Chebyshev spectral derivative for four functions of increasing smoothness. Compare Output 7 (p. 36).

Exercises

6.1. If $x_0, x_1, \ldots, x_N \in \mathbb{R}$ are distinct, then the *cardinal function* $p_j(x)$ defined by

$$p_j(x) = \frac{1}{a_j}\prod_{\substack{k=0\\k\neq j}}^{N}(x-x_k), \qquad a_j = \prod_{\substack{k=0\\k\neq j}}^{N}(x_j-x_k) \tag{6.7}$$

is the unique polynomial interpolant of degree N to the values 1 at x_j and 0 at x_k, $k \neq j$. Take the logarithm and differentiate to obtain

$$p_j'(x) = p_j(x)\sum_{\substack{k=0\\k\neq j}}^{N}(x-x_k)^{-1},$$

and from this derive the formulas

$$D_{ij} = \frac{1}{a_j} \prod_{\substack{k=0 \\ k\neq i,j}}^{N} (x_i - x_k) = \frac{a_i}{a_j(x_i - x_j)} \qquad (i \neq j) \tag{6.8}$$

and

$$D_{jj} = \sum_{\substack{k=0 \\ k\neq j}}^{N} (x_j - x_k)^{-1} \tag{6.9}$$

for the entries of the $N \times N$ differentiation matrix associated with the points $\{x_j\}$. (See also Exercise 12.2.)

6.2. Derive Theorem 7 from (6.8) and (6.9).

6.3. Suppose $-1 = x_0 < x_1 < x_2 < \cdots < x_N = 1$ lie in the minimal-energy configuration in $[-1, 1]$ in the sense discussed on p. 49. Show that except in the corners, the diagonal entries of the corresponding differentiation matrix D are zero.

6.4. It is mentioned on p. 55 that Chebyshev differentiation matrices have the symmetry property $(D_N)_{ij} = -(D_N)_{N-i,N-j}$. (a) Explain where this condition comes from. (b) Derive the analogous symmetry condition for $(D_N)^2$. (c) Taking N to be odd, so that the dimension of D_N is even, explain how $(D_N)^2$ could be constructed from just half the entries of D_N. For large N, how does the floating point operation count for this process compare with that for straightforward squaring of D_N?

6.5. Modify `cheb` so that it computes the diagonal entries of D_N by the explicit formulas (6.3)–(6.4) rather than by (6.6). Confirm that your code produces the same results except for rounding errors. Then see if you can find numerical evidence that it is less stable numerically than `cheb`.

6.6. The second panel of Output 12 shows a sudden dip for $N = 2$. Show that in fact, $E(2, 2) = 0$ (apart from rounding errors).

6.7. Theorem 6 makes a prediction about the geometric rate of convergence in the third panel of Output 12. Exactly what is this prediction? How well does it match the observed rate of convergence?

6.8. Let D_N be the usual Chebyshev differentiation matrix. Show that the power $(D_N)^{N+1}$ is identically equal to zero. Now try it on the computer for $N = 5$ and 20 and report the computed 2-norms $\|(D_5)^6\|_2$ and $\|(D_{20})^{21}\|_2$. Discuss.