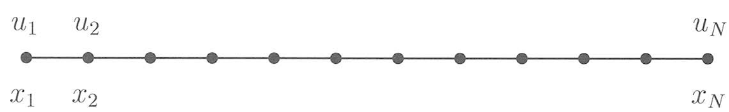
1. Differentiation Matrices

Our starting point is a basic question. Given a set of grid points $\{x_j\}$ and corresponding function values $\{u(x_j)\}$, how can we use this data to approximate the derivative of u? Probably the method that immediately springs to mind is some kind of finite difference formula. It is through finite differences that we shall motivate spectral methods.

To be specific, consider a uniform grid $\{x_1, \ldots, x_N\}$, with $x_{j+1} - x_j = h$ for each j, and a set of corresponding data values $\{u_1, \ldots, u_N\}$:



Let w_j denote the approximation to $u'(x_j)$, the derivative of u at x_j. The standard second-order finite difference approximation is

$$w_j = \frac{u_{j+1} - u_{j-1}}{2h}, \tag{1.1}$$

which can be derived by considering the Taylor expansions of $u(x_{j+1})$ and $u(x_{j-1})$. For simplicity, let us assume that the problem is periodic and take $u_0 = u_N$ and $u_1 = u_{N+1}$. Then we can represent the discrete differentiation process as a matrix-vector multiplication,

$$\begin{pmatrix} w_1 \\ \\ \vdots \\ \\ w_N \end{pmatrix} = h^{-1} \begin{pmatrix} 0 & \frac{1}{2} & & & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \ddots & & \\ & & \ddots & & \\ & & \ddots & 0 & \frac{1}{2} \\ \frac{1}{2} & & & -\frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} u_1 \\ \\ \vdots \\ \\ u_N \end{pmatrix}. \tag{1.2}$$

(Omitted entries here and in other sparse matrices in this book are zero.) Observe that this matrix is *Toeplitz*, having constant entries along diagonals; i.e., a_{ij} depends only on $i-j$. It is also *circulant*, meaning that a_{ij} depends only on $(i-j)\,(\mathrm{mod}\,N)$. The diagonals "wrap around" the matrix.

An alternative way to derive (1.1) and (1.2) is by the following process of local interpolation and differentiation:

For $j = 1, 2, \ldots, N$:

- *Let p_j be the unique polynomial of degree ≤ 2 with $p_j(x_{j-1}) = u_{j-1}$, $p_j(x_j) = u_j$, and $p_j(x_{j+1}) = u_{j+1}$.*
- *Set $w_j = p_j'(x_j)$.*

It is easily seen that, for fixed j, the interpolant p_j is given by

$$p_j(x) = u_{j-1}a_{-1}(x) + u_j a_0(x) + u_{j+1}a_1(x),$$

where $a_{-1}(x) = (x-x_j)(x-x_{j+1})/2h^2$, $a_0(x) = -(x-x_{j-1})(x-x_{j+1})/h^2$, and $a_1(x) = (x-x_{j-1})(x-x_j)/2h^2$. Differentiating and evaluating at $x = x_j$ then gives (1.1).

This derivation by local interpolation makes it clear how we can generalize to higher orders. Here is the fourth-order analogue:

For $j = 1, 2, \ldots, N$:

- *Let p_j be the unique polynomial of degree ≤ 4 with $p_j(x_{j\pm 2}) = u_{j\pm 2}$, $p_j(x_{j\pm 1}) = u_{j\pm 1}$, and $p_j(x_j) = u_j$.*
- *Set $w_j = p_j'(x_j)$.*

Again assuming periodicity of the data, it can be shown that this prescription amounts to the matrix-vector product

$$\begin{pmatrix} w_1 \\ \\ \\ \vdots \\ \\ \\ w_N \end{pmatrix} = h^{-1} \begin{pmatrix} & & \ddots & & & \frac{1}{12} & -\frac{2}{3} \\ & & \ddots & -\frac{1}{12} & & & \frac{1}{12} \\ & & \ddots & \frac{2}{3} & \ddots & & \\ & & \ddots & 0 & \ddots & & \\ & & \ddots & -\frac{2}{3} & \ddots & & \\ -\frac{1}{12} & & & \frac{1}{12} & \ddots & & \\ \frac{2}{3} & -\frac{1}{12} & & & \ddots & & \end{pmatrix} \begin{pmatrix} u_1 \\ \\ \\ \vdots \\ \\ \\ u_N \end{pmatrix}. \qquad (1.3)$$

This time we have a pentadiagonal instead of tridiagonal circulant matrix.

The matrices of (1.2) and (1.3) are examples of *differentiation matrices.* They have order of accuracy 2 and 4, respectively. That is, for data u_j obtained by sampling a sufficiently smooth function u, the corresponding discrete approximations to $u'(x_j)$ will converge at the rates $O(h^2)$ and $O(h^4)$ as $h \to 0$, respectively. One can verify this by considering Taylor series.

Our first MATLAB program, Program 1, illustrates the behavior of (1.3). We take $u(x) = e^{\sin(x)}$ to give periodic data on the domain $[-\pi, \pi]$:

The program compares the finite difference approximation w_j with the exact derivative, $e^{\sin(x_j)}\cos(x_j)$, for various values of N. Because it makes use of MATLAB sparse matrices, this code runs in a fraction of a second on a workstation, even though it manipulates matrices of dimensions as large as 4096 [GMS92]. The results are presented in Output 1, which plots the maximum error on the grid against N. The fourth-order accuracy is apparent. This is our first and last example that does not illustrate a spectral method!

We have looked at second- and fourth-order finite differences, and it is clear that consideration of sixth-, eighth-, and higher order schemes will lead to circulant matrices of increasing bandwidth. The idea behind spectral methods is to take this process to the limit, at least in principle, and work with a differentiation formula of infinite order and infinite bandwidth—i.e., a dense matrix [For75]. In the next chapter we shall show that in this limit, for an

Program 1

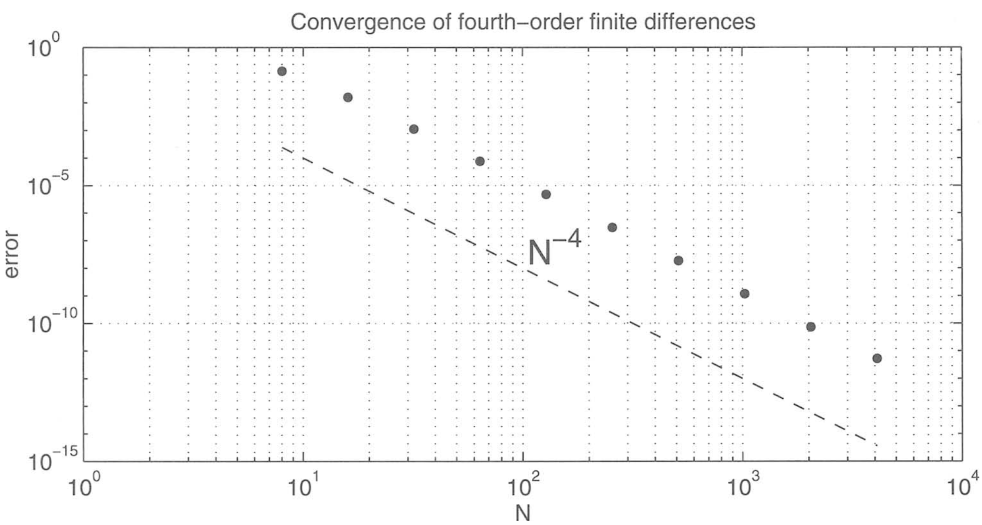
```
% p1.m - convergence of fourth-order finite differences

% For various N, set up grid in [-pi,pi] and function u(x):
  Nvec = 2.^(3:12);
  clf, subplot('position',[.1 .4 .8 .5])
  for N = Nvec
    h = 2*pi/N; x = -pi + (1:N)'*h;
    u = exp(sin(x)); uprime = cos(x).*u;

    % Construct sparse fourth-order differentiation matrix:
    e = ones(N,1);
    D =   sparse(1:N,[2:N 1],2*e/3,N,N)...
        - sparse(1:N,[3:N 1 2],e/12,N,N);
    D = (D-D')/h;

    % Plot max(abs(D*u-uprime)):
    error = norm(D*u-uprime,inf);
    loglog(N,error,'.','markersize',15), hold on
  end
  grid on, xlabel N, ylabel error
  title('Convergence of fourth-order finite differences')
  semilogy(Nvec,Nvec.^(-4),'--')
  text(105,5e-8,'N^{-4}','fontsize',18)
```

Output 1



Output 1: *Fourth-order convergence of the finite difference differentiation process* (1.3). *The use of sparse matrices permits high values of N.*

infinite equispaced grid, one obtains the following infinite matrix:

$$D = h^{-1}\begin{pmatrix} & & \vdots & & \\ & \ddots & \frac{1}{3} & & \\ & \ddots & -\frac{1}{2} & & \\ & \ddots & 1 & & \\ & & 0 & & \\ & & -1 & \ddots & \\ & & \frac{1}{2} & \ddots & \\ & & -\frac{1}{3} & \ddots & \\ & & \vdots & & \end{pmatrix}. \tag{1.4}$$

This is a skew-symmetric ($D^T = -D$) doubly infinite Toeplitz matrix, also known as a *Laurent operator* [Hal74, Wid65]. All its entries are nonzero except those on the main diagonal.

Of course, in practice one does not work with an infinite matrix. For a finite grid, here is the design principle for spectral collocation methods:

- *Let p be a single function (independent of j) such that $p(x_j) = u_j$ for all j.*
- *Set $w_j = p'(x_j)$.*

We are free to choose p to fit the problem at hand. For a periodic domain, the natural choice is a trigonometric polynomial on an equispaced grid, and the resulting "Fourier" methods will be our concern through Chapter 4 and intermittently in later chapters. For nonperiodic domains, algebraic polynomials on irregular grids are the right choice, and we will describe the "Chebyshev" methods of this type beginning in Chapters 5 and 6.

For finite N, taking N even for simplicity, here is the $N \times N$ dense matrix we will derive in Chapter 3 for a periodic, regular grid:

$$D_N = \begin{pmatrix} & & \vdots & & \\ & \ddots & \frac{1}{2}\cot\frac{3h}{2} & & \\ & \ddots & -\frac{1}{2}\cot\frac{2h}{2} & & \\ & \ddots & \frac{1}{2}\cot\frac{1h}{2} & & \\ & & 0 & & \\ & & -\frac{1}{2}\cot\frac{1h}{2} & \ddots & \\ & & \frac{1}{2}\cot\frac{2h}{2} & \ddots & \\ & & -\frac{1}{2}\cot\frac{3h}{2} & \ddots & \\ & & \vdots & & \end{pmatrix}. \tag{1.5}$$

Program 2

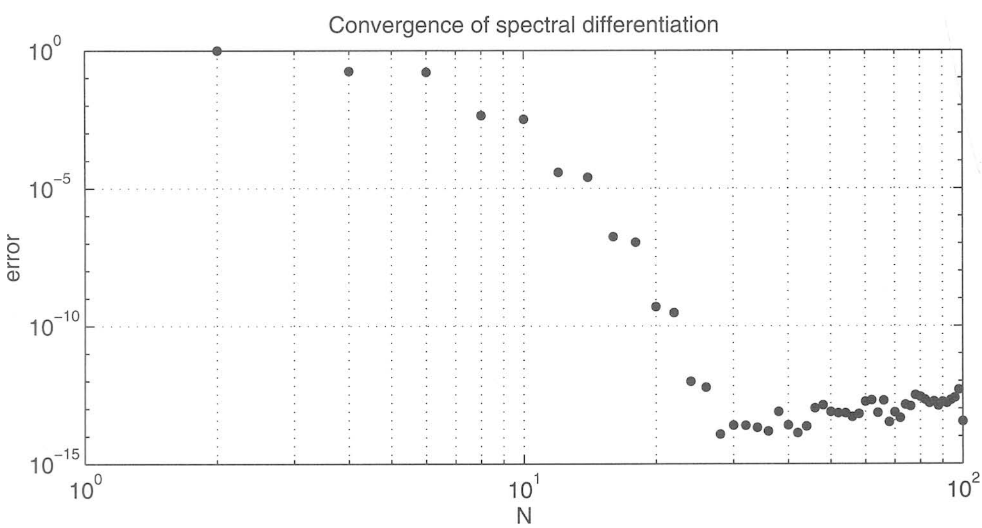
```
% p2.m - convergence of periodic spectral method (compare p1.m)

% For various N (even), set up grid as before:
  clf, subplot('position',[.1 .4 .8 .5])
  for N = 2:2:100;
    h = 2*pi/N;
    x = -pi + (1:N)'*h;
    u = exp(sin(x)); uprime = cos(x).*u;

    % Construct spectral differentiation matrix:
    column = [0 .5*(-1).^(1:N-1).*cot((1:N-1)*h/2)];
    D = toeplitz(column,column([1 N:-1:2]));

    % Plot max(abs(D*u-uprime)):
    error = norm(D*u-uprime,inf);
    loglog(N,error,'.','markersize',15), hold on
  end
  grid on, xlabel N, ylabel error
  title('Convergence of spectral differentiation')
```

Output 2



Output 2: *"Spectral accuracy" of the spectral method* (1.5), *until the rounding errors take over around* 10^{-14}. *Now the matrices are dense, but the values of* N *are much smaller than in Program* 1.

A little manipulation of the cotangent function reveals that this matrix is indeed circulant as well as Toeplitz (Exercise 1.2).

Program 2 is the same as Program 1 except with (1.3) replaced by (1.5). What a difference it makes in the results! The errors in Output 2 decrease very rapidly until such high precision is achieved that rounding errors on the computer prevent any further improvement.* This remarkable behavior is called *spectral accuracy*. We will give this phrase some precision in Chapter 4, but for the moment, the point to note is how different it is from convergence rates for finite difference and finite element methods. As N increases, the error in a finite difference or finite element scheme typically decreases like $O(N^{-m})$ for some constant m that depends on the order of approximation and the smoothness of the solution. For a spectral method, convergence at the rate $O(N^{-m})$ for *every* m is achieved, provided the solution is infinitely differentiable, and even faster convergence at a rate $O(c^N)$ $(0 < c < 1)$ is achieved if the solution is suitably analytic.

The matrices we have described have been circulant. The action of a circulant matrix is a convolution, and as we shall see in Chapter 3, convolutions can be computed using a discrete Fourier transform (DFT). Historically, it was the discovery of the fast Fourier transform (FFT) for such problems in 1965 that led to the surge of interest in spectral methods in the 1970s. We shall see in Chapter 8 that the FFT is applicable not only to trigonometric polynomials on equispaced grids, but also to algebraic polynomials on Chebyshev grids. Yet spectral methods implemented without the FFT are powerful, too, and in many applications it is quite satisfactory to work with explicit matrices. Most problems in this book are solved via matrices.

Summary of This Chapter. The fundamental principle of spectral collocation methods is, given discrete data on a grid, to interpolate the data globally, then evaluate the derivative of the interpolant on the grid. For periodic problems, we normally use trigonometric interpolants in equispaced points, and for nonperiodic problems, we normally use polynomial interpolants in unevenly spaced points.

Exercises

1.1. We derived the entries of the tridiagonal circulant matrix (1.2) by local polynomial interpolation. Derive the entries of the pentadiagonal circulant matrix (1.3) in the same manner.

---

*All our calculations are done in standard IEEE double precision arithmetic with $\epsilon_{\text{machine}} = 2^{-53} \approx 1.11 \times 10^{-16}$. This means that each addition, multiplication, division, and subtraction produces the exactly correct result times some factor $1 + \delta$ with $|\delta| \leq \epsilon_{\text{machine}}$. See [Hig96] and [TrBa97].

1.2. Show that (1.5) is circulant.

1.3. The dots of Output 2 lie in pairs. Why? What property of $e^{\sin(x)}$ gives rise to this behavior?

1.4. Run Program 1 to $N = 2^{16}$ instead of 2^{12}. What happens to the plot of the error vs. N? Why? Use the MATLAB commands `tic` and `toc` to generate a plot of approximately how the computation time depends on N. Is the dependence linear, quadratic, or cubic?

1.5. Run Programs 1 and 2 with $e^{\sin(x)}$ replaced by (a) $e^{\sin^2(x)}$ and (b) $e^{\sin(x)|\sin(x)|}$ and with `uprime` adjusted appropriately. What rates of convergence do you observe? Comment.

1.6. By manipulating Taylor series, determine the constant C for an error expansion of (1.3) of the form $w_j - u'(x_j) \sim Ch^4 u^{(5)}(x_j)$, where $u^{(5)}$ denotes the fifth derivative. Based on this value of C and on the formula for $u^{(5)}(x)$ with $u(x) = e^{\sin(x)}$, determine the leading term in the expansion for $w_j - u'(x_j)$ for $u(x) = e^{\sin(x)}$. (You will have to find $\max_{x\in[-\pi,\pi]} |u^{(5)}(x)|$ numerically.) Modify Program 1 so that it plots the dashed line corresponding to this leading term rather than just N^{-4}. This adjusted dashed line should fit the data almost perfectly. Plot the difference between the two on a log-log scale and verify that it shrinks at the rate $O(h^6)$.