Chapter 8.
Chebyshev spectral methods

8.1. Polynomial interpolation
8.2. Chebyshev differentiation matrices
8.3. Chebyshev differentiation by the FFT
8.4. Boundary conditions
8.5. Stability
8.6. Legendre points and other alternatives
8.7. Implicit methods and matrix iterations
8.8. Notes and references

This chapter discusses spectral methods for domains with boundaries. The effect of boundaries in spectral calculations is great, for they often introduce stability conditions that are both highly restrictive and difficult to analyze. Thus for a first-order partial differential equation solved on an N-point spatial grid by an explicit time-integration formula, a spectral method typically requires $k = O(N^{-2})$ for stability, in contrast to $k = O(N^{-1})$ for finite differences. For a second-order equation the disparity worsens to $O(N^{-4})$ vs. $O(N^{-2})$. To make matters worse, the matrices involved are usually nonnormal, and often very far from normal, so they are difficult to analyze as well as troublesome in practice.

Spectral methods on bounded domains typically employ grids consisting of zeros or extrema of Chebyshev polynomials ("Chebyshev points"), zeros or extrema of Legendre polynomials ("Legendre points"), or some other set of points related to a family or orthogonal polynomials. Chebyshev grids have the advantage that the FFT is available for an $O(N \log N)$ implementation of the differentiation process, and they also have slight advantages connected their ability to approximate functions. Legendre grids have various theoretical and practical advantages because of their connection with Gauss quadrature. At this point one cannot say which choice will win in the long run, but in this book, in keeping with out emphasis on Fourier analysis, most of the discussion is of Chebyshev grids.

Since explicit spectral methods are sometimes troublesome, implicit spectral calculations are increasingly popular. Spectral differentiation matrices are dense and ill-conditioned, however, so solving the associated systems of equations is not a trivial matter, even in one space dimension. Currently popular methods for solving these systems include preconditioned iterative methods and multigrid methods. These techniques are discussed briefly in §8.7.

8.1. Polynomial interpolation

Spectral methods arise from the fundamental problem of approximation of a function by interpolation on an interval. Multidimensional domains of a rectilinear shape are treated as products of simple intervals, and more complicated geometries are sometimes divided into rectilinear pieces.* In this section, therefore, we restrict our attention to the fundamental interval $[-1,1]$. The question to be considered is, what kinds of interpolants, in what sets of points, are likely to be effective?

Let $N \geq 1$ be an integer, even or odd, and let $x_0,\dots,x_N$ or sometimes $x_1,\dots,x_N$ be a set of distinct points in $[-1,1]$. For definiteness let the numbering be in reverse order:

$$1 \quad \geq \quad x_0 > x_1 > \cdots > x_{N-1} > x_N \quad \geq \quad -1. \tag{8.1.1}$$

The following are some grids that are often considered:

$$\textbf{Equispaced points:}\quad x_j = 1 - \frac{2j}{N} \quad (0 \leq j \leq N),$$

$$\textbf{Chebyshev zero points:}\quad x_j = \cos\frac{(j-1/2)\pi}{N} \quad (1 \leq j \leq N),$$

$$\textbf{Chebyshev extreme points:}\quad x_j = \cos\frac{j\pi}{N} \quad (0 \leq j \leq N),$$

$$\textbf{Legendre zero points:}\quad x_j = j\text{th zero of } P_N \quad (1 \leq j \leq N),$$

$$\textbf{Legendre extreme points:}\quad x_j = j\text{th extremum of } P_N \quad (0 \leq j \leq N),$$

where P_N is the Legendre polynomial of degree N. Chebyshev zeros and extreme points can also be described as zeros and extrema of Chebyshev polynomials T_N (more on these in §8.3). Chebyshev and Legendre zero points are also called Gauss-Chebyshev and Gauss-Legendre points, respectively, and Chebyshev and Legendre extreme points are also called Gauss-Lobatto-Chebyshev and Gauss-Lobatto-Legendre points, respectively. (These names originate in the field of numerical quadrature.)

---

*Such subdivision methods have been developed independently by I. Babushka and colleagues for structures problems, who call them **"p" finite element methods**, and by A. Patera and colleagues for fluids problems, who call them **spectral element methods.**

It is easy to remember how Chebyshev points are defined: they are the projections onto the interval $[-1,1]$ of equally-spaced points (roots of unity) along the unit circle $|z|=1$ in the complex plane:

Figure 8.1.1. Chebyshev extreme points ($N=8$).

To the eye, Legendre points look much the same, although there is no elementary geometrical definition. Figure 8.1.2 illustrates the similarity:

(a) $N=5$

(b) $N=25$

Figure 8.1.2. Legendre vs. Chebyshev zeros.

As $N\to\infty$, equispaced points are distributed with density

$$\mu(x)=\frac{N}{2} \qquad \text{Equally spaced,} \tag{8.1.2}$$

and Legendre or Chebyshev points—either zeros or extrema—have density

$$\mu(x) = \frac{N}{\pi\sqrt{1-x^2}} \qquad \text{Legendre, Chebyshev.} \tag{8.1.3}$$

Indeed, the density function (8.1.3) applies to point sets associated with any Jacobi polynomials, of which Legendre and Chebyshev polynomials are special cases.

Why is it a good idea to base spectral methods upon Chebyshev, Legendre, and other irregular grids? We shall answer this question by addressing a second, more fundamental question: why is it a good idea to interpolate a function $f(x)$ defined on $[-1,1]$ by a polynomial $p_N(x)$ rather than a trigonometric polynomial, and why is it a good idea to use Chebyshev or Legendre points rather than equally spaced points?

[The remainder of this section is just a sketch... details to be supplied later.]

PHENOMENA

Trigonometric interpolation in equispaced points suffers from the **Gibbs phenomenon**, due to Michelson and Gibbs at the turn of the twentieth century. $\|f - p_N\| = O(1)$ as $N \to \infty$, even if f is analytic. One can try to get around the Gibbs phenomenon by various tricks such as doubling the domain and reflecting, but the price is high.

Polynomial interpolation in equally spaced points suffers from the **Runge phenomenon**, due to Meray and Runge (Figure 8.1.3). $\|f - p_N\| = O(2^N)$—much worse!

Polynomial interpolation in Legendre or Chebyshev points: $\|f - p_N\| = O(\text{constant}^{-N})$ if f is analytic (for some constant greater than 1). Even if f is quite rough the errors will still go to zero provided f is, say, Lipschitz continuous.

Figure 8.1.3. The Runge phenomenon.

FIRST EXPLANATION—EQUIPOTENTIAL CURVES

Think of the limiting point distribution $\mu(x)$, above, as a charge density distribution; a charge at position x is associated with a potential $\log|z-x|$. Look at the equipotential curves of the resulting potential function $\phi(z) = \int_{-1}^{1}\mu(x)\log|z-x|\,dx$.

CONVERGENCE OF POLYNOMIAL INTERPOLANTS

Theorem 8.1.

In general, $\|f-p_N\|\to 0$ as $N\to\infty$ in the largest region bounded by an equipotential curve in which f is analytic. In particular:

For Chebyshev or Legendre points, or any other type of Gauss-Jacobi points, convergence is guaranteed if f is analytic on $[-1,1]$.

For equally spaced points, convergence is guaranteed if f is analytic in a particular lens-shaped region containing $(-1,1)$ (Figure 8.1.4).

Figure 8.1.4. Equipotential curves.

SECOND EXPLANATION—LEBESGUE CONSTANTS

Definition of **Lebesgue constant**:

$$\Lambda_N = \|I_N\|_\infty,$$

where I_N is the interpolation operator $I_N : f \mapsto p_N$. A small Lebesgue constant means that the interpolation process is not much worse than best approximation:

$$\|f - p_N\| \le (\Lambda_N + 1)\|f - p_N^*\|, \tag{8.1.1}$$

where p_N^* is the best (minimax, equiripple) approximation.

LEBESGUE CONSTANTS
Theorem 8.2. *Equispaced points:* $\Lambda_N \sim 2^N/e\,N\log N$. *Legendre points:* $\Lambda_N \sim \mathit{const}\sqrt{N}$. *Chebyshev points:* $\Lambda_N \sim \mathit{const}\log N$.

THIRD EXPLANATION—NUMBER OF POINTS PER WAVELENGTH

Consider approximation of, say, $f_N(x) = \cos\alpha Nx$ as $N \to \infty$. Thus f_N changes but the number of points per wavelength remains constant. Will the error $\|f_N - p_N\|$ go to zero? The answer to this question tells us something about the ability of various kinds of spectral methods to resolve data.

POINTS PER WAVELENGTH
Theorem 8.3. *Equispaced points: convergence if there are at least* 6 *points per wavelength.* *Chebyshev points: convergence if there are at least* π *points per wavelength on average.*

We have to say "on average" because the grid is nonuniform. In fact, it is $\pi/2$ times less dense in the middle than the equally spaced grid with the same number of points N (see (8.1.2) and (8.1.3)). Thus the second part of the theorem says that we need at least 2 points per wavelength in the center of the grid—the familiar Nyquist limit. See Figure 8.1.5. The first part of the theorem is mathematically valid, but of little value in practice because of rounding errors.

(a) Equally spaced points

(b) Chebyshev points

Figure 8.1.5. Error as a function of N in interpolation of $\cos \alpha N x$, with α, hence the number of points per wavelength, held fixed.

8.2. Chebyshev differentiation matrices

[Just a sketch]

From now on "Chebyshev points" means Chebyshev extreme points.

Multiplication by the first-order Chebyshev differentiation matrix D_N transforms a vector of data at the Chebyshev points into approximate derivatives at those points:

$$D_N \begin{bmatrix} v_0 \\ \vdots \\ v_N \end{bmatrix} = \begin{bmatrix} w_0 \\ \vdots \\ w_N \end{bmatrix}.$$

As usual, the implicit definition of D_N is as follows:

CHEBYSHEV SPECTRAL DIFFERENTIATION BY POLYNOMIAL INTERPOLATION.

(1) *Interpolate v by a polynomial $q(x) = q_N(x)$;*

(2) *Differentiate the interpolant at the grid points x_j:*

$$w_j = (D_N v)_j = q'(x_j). \tag{8.2.1}$$

Higher-order differentiation matrices are defined analogously. From this definition it is easy to work out the entries of D_N in special cases. For $N = 1$:

$$\mathbf{x} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \qquad D_1 = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}.$$

For $N = 2$:

$$\mathbf{x} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \qquad D_2 = \begin{bmatrix} \frac{3}{2} & -2 & \frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 2 & -\frac{3}{2} \end{bmatrix}.$$

For $N = 3$:

$$\mathbf{x} = \begin{bmatrix} 1 \\ \frac{1}{2} \\ -\frac{1}{2} \\ -1 \end{bmatrix}, \qquad D_3 = \begin{bmatrix} \frac{19}{6} & -4 & \frac{4}{3} & -\frac{1}{2} \\ 1 & -\frac{1}{3} & -1 & \frac{1}{3} \\ -\frac{1}{3} & 1 & \frac{1}{3} & -1 \\ \frac{1}{2} & -\frac{4}{3} & 4 & -\frac{19}{6} \end{bmatrix}.$$

These three examples illustrate an important fact, mentioned in the introduction to this chapter: Chebyshev spectral differentiation matrices are in general not symmetric or skew-symmetric. A more general statement is that they are not normal.* This is why stability analysis is difficult for spectral methods. The reason they are not normal is that unlike finite difference differentiation, spectral differentiation is not a translation-invariant process, but depends instead on the same global interpolant at all points x_j.

The general formula for D_N is as follows. First, define

$$c_i = \begin{cases} 2 & \text{for } i = 0 \text{ or } N, \\ 1 & \text{for } 1 \le i \le N-1, \end{cases} \tag{8.2.2}$$

and of course analogously for c_j. Then:

CHEBYSHEV SPECTRAL DIFFERENTIATION

Theorem 8.4. *Let $N \ge 1$ be any integer. The first-order spectral differentiation matrix D_N has entries*

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

$$(D_N)_{jj} = \frac{-x_j}{2(1-x_j^2)} \qquad \text{for } 1 \le j \le N-1,$$

$$(D_N)_{ij} = \frac{c_i}{c_j} \frac{(-1)^{i+j}}{x_i - x_j} \qquad \text{for } i \ne j.$$

Analogous formulas for D_N^2 can be found in Peyret (1986), Ehrenstein & Peyret [ref?] and in Zang, Streett, and Hussaini, ICASE Report 89-13, 1989. See also Canuto, Hussaini, Quarteroni & Zang.

*Recall that a normal matrix A is one that satisfies $AA^T = A^TA$. Equivalently, A possesses an orthogonal set of eigenvectors, which implies many desirable properties such as $\rho(A^n) = \|A^n\| = \|A\|^n$ for any n.

A note of caution: D_N is rarely used in exactly the form described in Theorem 8.4, for boundary conditions will modify it slightly, and these depend on the problem.

EXERCISES

▷ *8.2.1.* Prove that for any N, D_N is nilpotent: $D_N^n = 0$ for a sufficiently high integer n.

8.3. Chebyshev differentiation by the FFT

Polynomial interpolation in Chebyshev points is equivalent to trigonometric interpolation in equally spaced points, and hence can be carried out by the FFT. The algorithm described below has the optimal order $O(N \log N)$,* but we do not worry about achieving the optimal constant factor. For more practical discussions, see Appendix B of the book by Canuto, et al., and also P. N. Swarztrauber, "Symmetric FFTs," *Math. Comp. 47* (1986), 323–346. Valuable additional references are the book *The Chebyshev Polynomials* by Rivlin and Chapter 13 of P. Henrici, *Applied and Computational Complex Analysis,* 1986.

Consider three independent variables $\theta \in \mathbb{R}$, $x \in [-1,1]$, and $z \in S$, where S is the complex unit circle $\{z : |z| = 1\}$. They are related as follows:

$$z = e^{i\theta}, \qquad x = \operatorname{Re} z = \tfrac{1}{2}(z + z^{-1}) = \cos\theta, \tag{8.3.1}$$

which implies

$$\frac{dx}{d\theta} = -\sin\theta = -\sqrt{1-x^2}. \tag{8.3.2}$$

See Figure 8.3.1. Note that there are two conjugate values $z \in S$ for each $x \in (-1,1)$, and an infinite number of possible choices of θ.

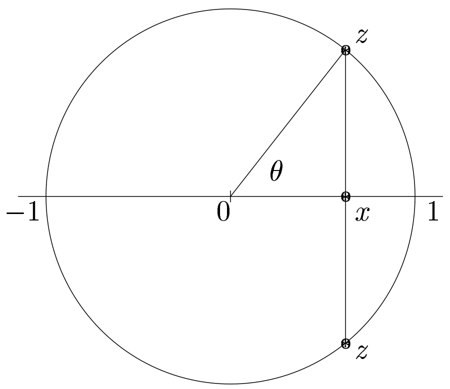

Figure 8.3.1. z, x, and θ.

---

*optimal, that is, so far as anyone knows as of 1994.

In generalization of the fact that the real part of z is x, the real part of z^n $(n \geq 0)$ is $T_n(x)$, the **Chebyshev polynomial** of degree n. This statement can be taken as a definition of Chebyshev polynomials:

$$T_n(x) = \operatorname{Re} z^n = \tfrac{1}{2}(z^n + z^{-n}) = \cos n\theta, \tag{8.3.3}$$

where x and z and θ are, as always, implicitly related by (8.3.1).* It is clear that (8.3.3) defines $T_n(x)$ to be *some* function of x, but it is not obvious that the function is a polynomial. However, a calculation of the first few cases makes it clear what is going on:

$$\begin{aligned}
T_0(x) &= \tfrac{1}{2}(z^0 + z^{-0}) = 1,\\
T_1(x) &= \tfrac{1}{2}(z^1 + z^{-1}) = x,\\
T_2(x) &= \tfrac{1}{2}(z^2 + z^{-2}) = \tfrac{1}{2}(z^1 + z^{-1})^2 - 1 = 2x^2 - 1,\\
T_3(x) &= \tfrac{1}{2}(z^3 + z^{-3}) = \tfrac{1}{2}(z^1 + z^{-1})^3 - \tfrac{3}{2}(z^1 + z^{-1}) = 4x^3 - 3x.
\end{aligned} \tag{8.3.4}$$

In general, the Chebyshev polynomials are related by the three-term recurrence relation

$$\begin{aligned}
T_{n+1}(x) &= \tfrac{1}{2}(z^{n+1} + z^{-n-1})\\
&= \tfrac{1}{2}(z^1 + z^{-1})(z^n + z^{-n}) - \tfrac{1}{2}(z^{n-1} + z^{-n+1})\\
&= 2xT_n(x) - T_{n-1}(x).
\end{aligned} \tag{8.3.5}$$

By (8.3.2) and (8.3.3), the derivative of $T_n(x)$ is

$$T_n'(x) = -n \sin n\theta \frac{d\theta}{dx} = \frac{n \sin n\theta}{\sin\theta}. \tag{8.3.6}$$

Thus just as x, z, and θ are equivalent, so are $T_n(x)$, z^n, and $\cos n\theta$. By taking linear combinations, we obtain three equivalent kinds of polynomials. A **trigonometric polynomial** $q(\theta)$ of degree N is a 2π-periodic sum of complex exponentials in θ (or equivalently, sines and cosines). Assuming that $q(\theta)$ is an even function of θ, it can be written

$$q(\theta) = \tfrac{1}{2}\sum_{n=0}^{N} a_n(e^{in\theta} + e^{-in\theta}) = \sum_{n=0}^{N} a_n \cos n\theta. \tag{8.3.7}$$

A **Laurent polynomial** $q(z)$ of degree N is a sum of negative and positive powers of z up to degree N. Assuming $q(z) = q(\bar{z})$ for $z \in S$, it can be written

$$q(z) = \tfrac{1}{2}\sum_{n=0}^{N} a_n(z^n + z^{-n}). \tag{8.3.8}$$

An **algebraic polynomial** $q(x)$ of degree N is a polynomial in x of the usual kind, and we can express it as a linear combination of Chebyshev polynomials:

$$q(x) = \sum_{n=0}^{N} a_n T_n(x). \tag{8.3.9}$$

*Equivalently, the Chebyshev polynomials can be defined as a system of polynomials orthogonal on $[-1, 1]$ with respect to the weight function $(1-x^2)^{-1/2}$.

The use of the same coefficients a_n in (8.3.7)–(8.3.9) is no accident, for all three of the polynomials above are identical:

$$q(\theta) = q(z) = q(x), \tag{8.3.10}$$

where again, x and z and θ are implicitly related by (8.3.1). For this reason we hope to be forgiven the sloppy use of the same letter q in all three cases.

Finally, for any integer $N \geq 1$, we define regular grids in the three variables as follows:

$$\theta_j = \frac{j\pi}{N}, \qquad z_j = e^{i\theta_j}, \qquad x_j = \operatorname{Re} z_j = \tfrac{1}{2}(z_j + z_j^{-1}) = \cos\theta_j \tag{8.3.11}$$

for $0 \leq j \leq N$. The points $\{x_j\}$ and $\{z_j\}$ were illustrated already in Figure 8.1.1. And now we are ready to state the algorithm for Chebyshev differentiation by the FFT.

ALGORITHM FOR CHEBYSHEV DIFFERENTIATION

1. *Given data* $\{v_j\}$ *defined at the Chebyshev points* $\{x_j\}$, $0 \le j \le N$, *think of the same data as being defined at the equally spaced points* $\{\theta_j\}$ *in* $[0,\pi]$.

2. *(FFT) Find the coefficients* $\{a_n\}$ *of the trigonometric polynomial*

$$q(\theta) = \sum_{n=0}^{N} a_n \cos n\theta \tag{8.3.12}$$

that interpolates $\{v_j\}$ *at* $\{\theta_j\}$.

3. *(FFT) Compute the derivative*

$$\frac{dq}{d\theta} = -\sum_{n=0}^{N} n\, a_n \sin n\theta. \tag{8.3.13}$$

4. *Change variables to obtain the derivative with respect to* x:

$$\frac{dq}{dx} = \frac{dq}{d\theta}\frac{d\theta}{dx} = \sum_{n=0}^{N} \frac{n\, a_n \sin n\theta}{\sin\theta} = \sum_{n=0}^{N} \frac{n\, a_n \sin n\theta}{\sqrt{1-x^2}}. \tag{8.3.14}$$

At $x = \pm 1$, *i.e.* $\theta = 0, \pi$, *L'Hopital's rule gives the special values*

$$\frac{dq}{dx}(\pm 1) = \sum_{n=0}^{N} (\pm 1)^n n^2 a_n \tag{8.3.15}$$

5. *Evaluate the result at the Chebyshev points:*

$$w_j = \frac{dq}{dx}(x_j). \tag{8.3.16}$$

Note that by (8.3.3), equation (8.3.12) can be interpreted as a linear combination of Chebyshev polynomials, and by (8.3.6), equation (8.3.14) is the corresponding linear combination of derivatives.* But of course the algorithmic content of the description above relates to the θ variable, for in Steps 2 and 3, we have performed Fourier spectral differentiation exactly as in §7.3: discrete Fourier transform, multiply by $i\xi$, inverse discrete Fourier transform. Only the use of sines and cosines rather than complex exponentials, and of n instead of ξ, has disguised the process somewhat.

* or of Chebyshev polynomials $U_n(x)$ of the *second kind.*

EXERCISES

▶ *8.3.1. Fourier and Chebyshev spectral differentiation.*

Write four brief, elegant Matlab programs for first-order spectral differentiation:

FDERIVM, CDERIVM: construct differentiation matrices;

FDERIV, CDERIV: differentiate via FFT.

In the Fourier case, there are N equally spaced points $x_{-N/2},\dots,x_{N/2-1}$ (N even) in $[-\pi,\pi]$, and no boundary conditions. In the Chebyshev case, there are N Chebyshev points $x_1,\dots,x_N$ in $[-1,1)$ (N arbitrary), with a zero boundary condition at $x=1$. The effect of this boundary condition is that one removes the first row and first column from D_N, leading to a square matrix of dimension N instead of $N+1$.

You do not have to worry about computational efficiency (such as using an FFT of length N rather than $2N$ in the Chebyshev case), but you are welcome to worry about it if you like.

Experiment with your programs to make sure they differentiate successfully. Of course, the matrices can be used to check the FFT programs.

(a) Turn in a plot showing the function $u(x)=\cos(x/2)$ and its derivative computed by FDERIV, for $N=32$. Discuss the results.

(b) Turn in a plot showing the function $u(x)=\cos(\pi x/2)$ and its derivative computed by CDERIV, again for $N=32$. Discuss the results.

(c) Plot the eigenvalues of D_N for Fourier and Chebyshev spectral differentiation with $N=8$, 16, 32, 64.

8.5. Stability

This section is not yet written. What follows is a copy of a paper of mine from K. W. Morton and M. J. Baines, eds., *Numerical Methods for Fluid Dynamics III*, Clarendon Press, Oxford, 1988.

Because of stability problems like those described in this paper, more and more attention is currently being devoted to implicit time-stepping methods for spectral computations. The associated linear algebra problems are generally solved by preconditioned matrix iterations, sometimes including a multigrid iteration.

This paper was written before I was using the terminology of pseudospectra. I would now summarize Section 5 of this paper by saying that although the spectrum of the Legendre spectral differentiation matrix is of size $\Theta(N)$ as $N \to \infty$, the pseudospectra are of size $\Theta(N^2)$ for any $\epsilon > 0$. The connection of pseudospectra with stability of the method of lines was discussed in Sections 4.5–4.7.

8.6. Some review problems

EXERCISES

▷ *8.6.1. TRUE or FALSE?* Give each answer together with at most two or three sentences of explanation. The best possible explanation is a proof, a counterexample, or the citation of a theorem in the text from which the answer follows. If you can't do quite that well, try at least to give a convincing reason why the answer you have chosen is the right one. In some cases a well-thought-out sketch will suffice.

(a) The Fourier transform of $f(x) = \exp(-x^4)$ has compact support.

(b) When you multiply a matrix by a vector on the right, i.e. Ax, the result is a linear combination of the columns of that matrix.

(c) If an ODE initial-value problem with a smooth solution is solved by the fourth-order Adams-Bashforth formula with step size k, and the missing starting values v^1, v^2, v^3 are obtained by taking Euler steps with some step size k', then in general we will need $k' = O(k^4)$ to maintain overall fourth-order accuracy.

(d) If a consistent finite difference model of a well-posed linear initial-value problem violates the CFL condition, it must be unstable.

(e) If you Fourier transform a function $u \in L^2$ four times in a row, you end up with u again, times a constant factor.

(f) If the function $f(x) = (x^2 - 2x + 26/25)^{-1}$ is interpolated by a polynomial $q_N(x)$ in N equally spaced points of $[-1,1]$, then $\|f - q_N\|_\infty \to 0$ as $N \to \infty$.

(g) $e^x = O(xe^{x/2})$ as $x \to \infty$.

(h) If a stable finite-difference approximation to $u_t = u_x$ with real coefficients has order of accuracy 3, then the formula must be dissipative.

(i) If

$$A = \begin{pmatrix} \frac{1}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix},$$

then $\|A^n\| < C \ \ \forall n$ for some constant $C < \infty$.

(j) If the equation $u_t = -100A^2 u$ is solved by the fourth-order Adams-Moulton formula, where $u(x,t)$ is a 2-vector and A is the matrix above, then $k = 0.01$ is a sufficiently small time step to ensure time-stability.

(k) Let $u_t = u_{xx}$ on $[-\pi, \pi]$, with periodic boundary conditions, be solved by Fourier pseudospectral differentiation in x coupled with a fourth-order Runge-Kutta formula in t. For $N=32$, $k=0.01$ is a sufficiently small time step to ensure time-stability.

(l) The ODE initial-value problem $u_t = f(u,t) = \cos^2 u$, $u(0)=1$, $0 \le t \le 100$, is well-posed.

(m) In exact arithmetic and with exact starting values, the numerical approximations computed by the linear multistep formula

$$v^{n+3} = \tfrac{1}{3}(v^{n+2}+v^{n+1}+v^n)+\tfrac{2}{3}k(f^{n+2}+f^{n+1}+f^n)$$

are guaranteed to converge to the unique solution of a well-posed initial-value problem in the limit $k \to 0$.

(n) If computers did not make rounding errors, we would not need to study stability.

(o) The solution at time $t=1$ to $u_t = u_x + u_{xx}$ ($x \in \mathbb{R}$, initial data $u(x,0)=f(x)$) is the same as what you would get by first diffusing the data $f(x)$ according to the equation $u_t = u_{xx}$, then translating the result leftward by one unit according to the equation $u_t = u_x$.

(p) The discrete Fourier transform of a three-dimensional periodic set of data on an $N \times N \times N$ grid can be computed on a serial computer in $O(N^3 \log N)$ operations.

(q) The addition of numerical dissipation may sometimes increase the stability limit of a finite difference formula without affecting the order of accuracy.

(r) For a nondissipative semidiscrete finite-difference model (i.e., discrete space but continuous time), phase velocity as well as group velocity is a well-defined quantity.

(s) $v_0^{n+1} = v_4^n$ is a stable left-hand boundary condition for use with the leap frog model of $u_t = u_x$ with $k/h = 0.5$.

(t) If a finite difference model of a partial differential equation is stable with $k/h = \lambda_0$ for some $\lambda_0 > 0$, then it is stable with $k/h = \lambda$ for any $\lambda \le \lambda_0$.

(u) To solve the system of equations that results from a standard second-order discretization of Laplace's equation on an $N \times N \times N$ grid in three dimensions by the obvious method of banded Gaussian elimination, without any clever tricks, requires $\Theta(N^7)$ operations on a serial computer.

(v) If $u(x,t)$ is a solution to $u_t = iu_{xx}$ for $x \in \mathbb{R}$, then the 2-norm $\|u(\cdot,t)\|$ is independent of t.

(w) In a method of lines discretization of a well-posed linear IVP, having the appropriate eigenvalues fit in the appropriate stability region is sufficient but not necessary for Lax-stability.

(x) Suppose a signal that's band-limited to frequencies in the range $[-40\text{kHz}, 40\text{kHz}]$ is sampled 60,000 times a second, i.e., fast enough to resolve frequencies in the range $[-30\text{kHz}, 30\text{kHz}]$. Then although some aliasing will occur, the information in the range $[-20\text{kHz}, 20\text{kHz}]$ remains uncorrupted.

8.7. Two final problems

EXERCISES

▶ *8.7.1. Equipotential curves.* Write a short and elegant Matlab program to plot equipotential curves in the plane corresponding to a vector of point charges (interpolation points) $x_1, \dots, x_N$. Your program should simply sample $N^{-1}\sum \log|z - x_j|$ on a grid, then produce a contour plot of the result. (See `meshdom` and `contour`.) Turn in beautiful plots corresponding to (a) 6 equispaced points, (b) 6 Chebyshev points, (c) 30 equispaced points, (d) 30 Chebyshev points. By all means play around with 3D graphics, convergence and divergence of associated interpolation processes, or other amusements if you're in the mood.

▶ *8.7.2. Fun with Chebyshev spectral methods.* The starting point of this problem is the Chebyshev differentiation matrix of Exercise 8.3.1. It will be easiest to use a program like CDERIVM from that exercise, which works with an explicit matrix rather than the FFT. Be careful with boundary conditions; you will want to square the $(N+1)\times(N+1)$ matrix first before stripping off any rows or columns.

(a) *Poisson equation in 1D.* The function $u(x) = (1-x^2)e^x$ satisfies $u(\pm 1) = 0$ and has second derivative $u''(x) = -(1+4x+x^2)e^x$. Thus it is the solution to the boundary value problem

$$u_{xx} = -(1+4x+x^2)e^x, \quad x \in [-1,1], \quad u(\pm 1) = 0. \tag{1}$$

Write a little Matlab program to solve (1) by a Chebyshev spectral method and produce a plot of the computed discrete solution values ($N+1$ discrete points in $[-1,1]$) superimposed upon exact solution (a curve). Turn in the plot for $N=6$ and a table of the errors $u_{\text{computed}}(0) - u_{\text{exact}}(0)$ for $N = 2,4,6,8$. What can you say about the rate of convergence?

(b) *Poisson equation in 2D.* Similarly, the function $u(x,y) = (1-x^2)(1-y^2)\cos(x+y)$ is the solution to the boundary value problem

$$u_{xx} + u_{yy} = \texttt{<sorry, illegible!>}, \quad x,y \in [-1,1], \quad u(x,\pm 1) = u(\pm 1, y) = 0. \tag{2}$$

Write a Matlab program to solve (2) by a Chebyshev spectral method involving a grid of $(N-1)^2$ interior points. You may find that the Matlab command KRON comes in handy for this purpose. You don't have to produce a plot of the computed solution, but do turn in a table of $u_{\text{computed}}(0,0) - u_{\text{exact}}(0,0)$ for $N = 2,4,6,8$. How does the rate of convergence look?

(c) *Heat equation in 1D.* Back to 1D now. Suppose you have the problem

$$u_t = u_{xx}, \quad u(\pm 1, t) = 0, \quad u(x,0) = (1-x^2)e^x. \tag{3}$$

At what time t_c does $\max_{x\in[-1,1]} u(x,t)$ first fall below 1? Figure out the answer to at least 2 digits of relative precision. Then describe what you would do if I asked for 12 digits.